Dr Jekyll and Mr Hyde:
The Strange Case of Off-Policy Policy Updates

Romain Laroche[*]
Microsoft Research Montréal, Canada

Rémi Tachet des Combes[*]
Microsoft Research Montréal, Canada

Abstract

The policy gradient theorem states that the policy should only be updated in states that are visited by the current policy, which leads to insufficient planning in the *off-policy* states, and thus to convergence to suboptimal policies. We tackle this planning issue by extending the policy gradient theory to *policy updates* with respect to any state density. Under these generalized policy updates, we show convergence to optimality under a necessary and sufficient condition on the updates' state densities, and thereby solve the aforementioned planning issue. We also prove asymptotic convergence rates that significantly improve those in the policy gradient literature. To implement the principles prescribed by our theory, we propose an agent, Dr Jekyll & Mr Hyde (J&H), with a double personality: Dr Jekyll purely exploits while Mr Hyde purely explores. J&H's independent policies allow to record two separate replay buffers: one on-policy (Dr Jekyll's) and one off-policy (Mr Hyde's), and therefore to update J&H's models with a mixture of on-policy and off-policy updates. More than an algorithm, J&H defines principles for actor-critic algorithms to satisfy the requirements we identify in our analysis. We extensively test on finite MDPs where J&H demonstrates a superior ability to recover from converging to a suboptimal policy without impairing its speed of convergence. We also implement a deep version of the algorithm and test it on a simple problem where it shows promising results.

1 Introduction

Policy Gradient algorithms in Reinforcement Learning (RL) have enjoyed great success both theoretically [50, 43, 16, 36] and empirically [23, 38, 37, 27]. Their principle consists in optimizing an objective function $\mathcal{J}$ (the expected discounted return) through gradient steps, both being formally specified below [43]:

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

$$\nabla_\theta \mathcal{J}(\pi) = \sum_{s\in\mathcal{S}} d_{\pi,\gamma}(s) \sum_{a\in\mathcal{A}} q_\pi(s,a)\nabla_\theta \pi(a|s) \quad \text{and} \quad \theta_{t+1} \xleftarrow[proj]{} \theta_t + \eta_t \nabla_{\theta_t}\mathcal{J}(\pi_t), \tag{2}$$

using standard Markov Decision Process notations (recalled in App. A), where η_t is the learning rate and $d_{\pi,\gamma}(s) \doteq \sum_{t=0}^{\infty}\gamma^t \mathbb{P}(S_t = s|\pi)$ the discounted state density, the policies $\pi \doteq \pi_\theta$ and $\pi_t \doteq \pi_{\theta_t}$ are implicitly parametrized by θ for conciseness, and $\xleftarrow[proj]{}$ denotes the projection onto the parameter space. We observe that the update established by the policy gradient theorem is proportional to the state density of the current policy. This is problematic, as it implies that no value improvement can be induced in *off-policy* states (*i.e.* states that are rarely visited). As a consequence,

[*] Equal contribution. Correspondence to: rolaroch@microsoft.com. All code available at http://aka.ms/jnh.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

planning is inefficient in those states to the point of potentially compromising convergence to the optimal policy.

To address the planning issue in off-policy states, we generalize the policy gradient theory by considering the more general policy update:

$$U(\theta, d) \doteq \sum_{s\in\mathcal{S}} d(s) \sum_{a\in\mathcal{A}} q_\pi(s,a)\nabla_\theta \pi(a|s) \quad \text{and} \quad \theta_{t+1} \xleftarrow[proj]{} \theta_t + \eta_t U(\theta_t, d_t), \tag{3}$$

where d_t designates the state distribution on which the policy is updated. It does not have to match the distribution induced by π and updates may therefore be off-policy. We note that $U(\theta, d)$ is not a gradient in the general case, hence the use of the term *update*. We prove in the direct: $\pi_\theta \doteq \theta$ and softmax: $\pi_\theta \dot\sim \exp(\theta)$ parametrizations that following the exact policy updates induces a sequence of value functions $q_t \doteq q_{\pi_t}$ that is monotonously increasing, upper bounded, and thereby converges to a q_∞. We then show that the condition "$\forall s \in \mathcal{S},\ \sum_t \eta_t d_t(s) = \infty$" on the sequence (d_t) is necessary and sufficient for q_∞ to be optimal. Our result generalizes previous theorems to broader updates and milder assumptions [2, 22]. Finally, we significantly improve the existing asymptotic convergence rates. We show that (q_t) converges to the optimal value i) exactly in finite time for the direct parametrization (previously in $\mathcal{O}\left(t^{-1/2}|\mathcal{S}||\mathcal{A}|(1-\gamma)^{-6}\right)$[2]), and ii) in $\mathcal{O}\left(t^{-1}|\mathcal{S}||\mathcal{A}|(1-\gamma)^{-2}\right)$ for the softmax parametrization, to be compared with $\mathcal{O}\left(t^{-1}|\mathcal{S}|^2|\mathcal{A}|^2(1-\gamma)^{-6}\right)$ [22].

Building on our theoretical results, we design a novel algorithm: Dr Jekyll and Mr Hyde (J&H). Its principle consists in training two independent policies: a pure exploitation one (Dr Jekyll) and a pure exploration one (Mr Hyde), and give control to either one for full trajectories. J&H's independent policies allow to record two separate replay buffers: one on-policy (Dr Jekyll's) and one off-policy (Mr Hyde's), and therefore to update J&H's models with any desired mixture of on-policy and off-policy updates. Beyond an algorithm, J&H introduces conditions based on our analysis that actor-critic algorithms should follow to properly plan off-policy. Furthermore, the separation of exploration and exploitation allows to stabilise the training of the exploitation policy while ensuring a full coverage of the state-action space through deep exploration [30].

We empirically validate our theoretical analysis and algorithmic innovation in both planning and RL settings. In the planning setting where we assume that q_t is exactly known, we analyze the impact of the off-policiness of d_t and conclude that, while it improves over classic policy gradient, the theoretical sufficient condition does not allow to converge in a reasonable amount of time in very hard planning tasks: we thereby recommend to enforce the stronger condition "$\forall s \in \mathcal{S},\ \sum_t \eta_t d_t(s) \in \Omega(t)$". In the reinforcement learning domain, where q_t must be estimated from the collected transitions, the off-policiness of J&H's policy updates allows by design to satisfy the recommendation as long as Mr Hyde is able to cover the whole state space. This leads J&H to significantly outperform all competing actor-critic algorithms in the hard planning tasks and to be competitive with gradient updates in simple planning problems. As a proof of concept, we also test J&H in a deep reinforcement learning setting and show that it outperforms various baselines on a simple environment.

The paper is organized as follows. Section 2 develops the policy update theory. Section 3 describes J&H and positions it with respect to the literature. Section 4 presents the experiments and their results. Finally, Section 5 concludes the paper. The interested reader may refer to the appendix for the proofs (App. B), the domains (App. C), and the full experiment reports (App. D, E, and F). Finally, the supplementary material contains the code for the experiments: algorithms and environments.

2 Theoretical analysis

First, we recall some background: policy gradient methods depend on a parametrization $\theta \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ of the policy. Like [2, 22], we will focus on the classic direct and softmax parametrizations:

- direct: $\pi(a|s) \doteq \theta_{s,a}$, for which $u_{s,a} = d_t(s)q_\pi(s,a)$ and the projection is on the simplex.
- softmax: $\pi(a|s) \doteq \dfrac{\exp(\theta_{s,a})}{\sum_{a'}\exp(\theta_{s,a'})}$, for which $u_{s,a} = d_t(s)\pi(a|s)(q_\pi(s,a) - v_\pi(s))$.

For both parametrizations, [2] proves that following the policy gradient $\nabla_\theta \mathcal{J}(\pi)$ from Eq. (2) eventually leads to the optimal policy in finite MDPs under the following assumptions and condition[2]:

A1. The model (p_0, p, r) is known.

A2. The initial state-distribution covers the full state space: $\forall s \in \mathcal{S}, p_0(s) > 0$.

C3. The learning rate is constant: $\eta_t = \frac{(1-\gamma)^3}{2\gamma|\mathcal{A}|}$ (direct) and $\eta_t = \frac{(1-\gamma)^3}{8}$ (softmax).

Those are stringent requirements. A1 implies that the values q_π and state density $d_{\pi,\gamma}$ of the policy are known exactly. A2 is generally not satisfied in standard RL domains. Finally, verifying C3 makes learning slow to the point that it becomes impractical.

2.1 Convergence properties

In this section, we study, under A1, the convergence properties of the sequence of value functions (q_t) induced by the update rule defined in Eq. (3). We note that $U(\theta, d)$ is not a gradient in general. Our first theoretical result is the monotonicity of the value function sequence (q_t).

Theorem 1 (Monotonicity under the direct and softmax parametrization). *Under A1, the sequence of value functions $q_t \doteq q_{\pi_t}$ and $v_t \doteq v_{\pi_t}$ are monotonously increasing.*

By the monotonous convergence theorem, Thm. 1 directly implies the convergence of (q_t):

Corollary 1 (Convergence under the direct and softmax parametrization). *Under A1, the sequence of value functions q_t uniformly converges: $q_\infty \doteq \lim_{t\to\infty} q_t$.*

In order to go further and prove convergence to a global optimal value, we need to enforce an additional condition. Indeed, the update U relies multiplicatively on d_t which could be equal (or tend very fast) to 0 in some states, compromising the policy's ability to reach the optimal value. This argument is not only theoretical, it has been observed by many that policy gradient can get stuck in suboptimal policies, even with entropy regularization. We designed our chain domain to exhibit such behaviour (see Section 4.1).

Next, we show that (q_t) converges to the optimal value function under some necessary and sufficient condition.

Theorem 2 (Optimality under the direct and softmax parametrization). *Under A1, the following condition:*

C4. 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 optimality: $q_\infty = q_\star \doteq \max_{\pi\in\Pi} q_\pi$.

Proof sketch. In both direct and softmax parametrizations, we assume there exists a state-action pair (s, a) advantageous with respect to the state value limits q_∞ and $v_\infty \doteq \lim_{t\to\infty} v_t$, that is: $\text{adv}_\infty(s, a) \doteq q_\infty(s, a) - v_\infty(s) > 0$. We then prove that in this state s, the policy improvement yielded by the update is lower bounded by a linear function of the update weight $\eta_t d_t(s)$. Both parametrizations require different proof techniques and are dealt with in different theorems. Summing over t, we notice that this lower bounding sum diverges to infinity, which contradicts Corollary 1. We therefore infer that there cannot exist such a state-action pair, which allows us to conclude that no policy improvement is possible, and thus that the values are optimal: $q_\infty = \max_{\pi\in\Pi} q_\pi$.

For the necessity of C4, we show that the parameter update is upper bounded in both parametrizations by a term linear in the action gap. By choosing the reward function adversarially, we may set it sufficiently small so that the sum of all the gradient steps is insufficient to reach optimality. □

Thm. 2 is impactful along five dimensions.

Practice of on-policy undiscounted updates: As Thm. 2 is applicable to any distribution sequence, it allows us in particular to consider the practice of using on-policy undiscounted updates: $d_t \doteq d_{\pi_t,1}$.

[2] An assumption A# is a requirement on the environment or the application setting, while a condition C# is a requirement that may be enforced by a dedicated algorithm in any environment or setting.

Thus, it resolves a longstanding gap between the policy gradient theory and the actor-critic algorithm implementations [12, 7, 51, 17, 1, 19, 41]. This mismatch and its lack of theoretical grounding were identified in [45]. Later, [26] proved that the practitioners' undiscounted updates are not the gradient of any function and may be strongly biased under state aliasing. Recently, [53] studied the practical advantage of the undiscounted updates from a representation learning perspective. C4 encompasses both standard policy gradient and the undiscounted update rule: convergence to optimality of both is guaranteed as long as C4 is verified. Our analysis thus shows that the convergence properties of policy gradient and undiscounted updates require the same set of assumptions and conditions. Conversely, it is possible to prove convergence to sub-optimal policies of either when C4 is violated. In other words, Thm. 2 allows to reduce the study of specific algorithms to whether they verify C4.

Experience replay and off-policy updates: It also justifies the use of an experience replay for the actor, a trick also widely used in the literature to distributed the training over several agents [24, 34, 49, 14, 35]. Furthermore, while widely overlooked in the literature, we prove that off-policy updates have an even higher impact, since they are necessary to guarantee the convergence to optimality[3]. We leverage this discovery to introduce new design principles and a novel algorithm in Section 3.

Policy gradient theory: Next, Thm. 2 generalizes previous optimality convergence results along two axes. The initial state distribution p_0 is not required to cover the state space anymore (aka A2). A2 is unrealistic in most applications and cannot be enforced by an algorithm. Relaxing it is an important open problem [2], we disprove it in the full class of MDP. However, it might be possible to find a class of MDPs larger than the one verifying A2 where policy gradient also converges. Additionally, the condition on our learning rates is also much more flexible than C3.

Density vs. policy regularization: In contrast, C4 can be controlled by a dedicated algorithm. This theoretical result promotes the principle of density-based regularization [21, 32], at the expense of policy-based regularization that cannot guarantee that C4 is satisfied and sometimes fails to plan over the whole state space (our chain domain experiments in Section 4 illustrate it well).

Towards a generalized policy iteration theorem: Finally, by combining Thm. 2, which analyzes policy improvement, with a thorough theoretical analysis of the policy evaluation step (discussed in Section 3.1), we see a path towards results describing conditions on both components of generalized policy iteration that guarantee convergence to optimality [4]. Furthermore, Thm. 2 gives sufficient conditions on the policy improvement step. The generalized policy iteration theorem was conjectured in [42] but never proved.

2.2 Convergence rates

Next, we give convergence rates for both softmax and direct parametrizations:

Theorem 3 (**Asymptotic convergence rates under the direct parametrization**). *With direct parametrization, under A1 and C4, the sequence of value functions q_t converges to optimality in finite time:*

$$\exists t_0, \textit{ such that } \forall t \geq t_0, \quad q_t = q_\star. \tag{4}$$

C4 is required for optimality and convergence rate contrarily to the softmax parametrization. We significantly improve over previous bounds in $\mathcal{O}\left(t^{-1/2}|\mathcal{S}||\mathcal{A}|(1-\gamma)^{-6}\right)$[2]. We wish to emphasize that Theorem 3 does not say anything about how large t_0 is, the rate is purely asymptotic.

Theorem 4 (**Asymptotic convergence rates under the softmax parametrization**). *With softmax parametrization, under A1, C4, and A8:*

A8. The optimal policy is unique: $\forall s,\ q_\star(s,a_1) = q_\star(s,a_2) = v_\star(s)$ *implies* $a_1 = a_2$,

the sequence of value functions q_t converges asymptotically as follows:

$$\exists t_0, \textit{ such that } \forall t \geq t_0, \quad v_\star(s) - v_t(s) \leq \frac{8|\mathcal{A}|(v_\top - v_\bot)}{(1-\gamma)\min_{s\in \textit{supp}(d_{\pi_\star,\gamma})} \delta(s) \sum_{t'=t_0}^{t-1} \eta_{t'} d_{t'}(s)}, \tag{5}$$

[3] Other papers proving convergence to optimality of policy gradient rely on A2, which implies C4.

where $\delta(s) = v_\star(s) - \max_{a\in\mathcal{A}/\{\pi_\star(s)\}} q_\star(s,a)$ is the gap with the best suboptimal action in state s, $v_\top$ (resp. $v_\perp$) is the maximal (resp. minimal) value, and $\text{supp}(d_{\pi_\star,\gamma})$ denotes the support of the distribution of the optimal policy.

If (d_t) satisfies the additional mild condition that the support of $d_\star$ is covered *on average*:

C5. $\forall s \in \text{supp}(d_\star), \lim_{t\to\infty} \frac{1}{t}\sum_{t'=0}^{t} \eta_{t'} d_{t'}(s) \doteq e_\perp(s) > 0,$

then we obtain the following convergence rate in $\mathcal{O}(t^{-1})$:

$$\exists t_0, \text{ such that } \forall t \geq t_0, \quad v_\star(s) - v_t(s) \leq \frac{8|\mathcal{A}|(v_\top - v_\perp)}{(t-t_0)(1-\gamma)\min_{s\in\text{supp}(d_{\pi_\star,\gamma})}\delta(s)e_\perp(s)}. \tag{6}$$

C5 is implied by A2 and C3. Alternately, assuming $\eta_t \geq \eta_\perp > 0\ \forall t$, it is verified by a uniform distribution in which case $e_\perp(s) = \frac{\eta_\perp}{|\mathcal{S}|}$, or by an on-policy distribution d_{π_t} that converges to the distribution of some optimal policy $d_{\pi_\star}$ in which case $e_\perp(s) = \eta_\perp d_{\pi_\star}(s)$. Also, note that C4 is required for optimality, but not for convergence rates with respect to q_∞. Thm. 4 establishes asymptotic rates in $\mathcal{O}\left(t^{-1}|\mathcal{S}||\mathcal{A}|(1-\gamma)^{-2}\right)$, to be compared with $\mathcal{O}\left(t^{-1}|\mathcal{S}|^2|\mathcal{A}|^2(1-\gamma)^{-6}\right)$ [22]. The gap is partially explained by dropping C3 and allowing any learning rate (see App. A.6 for a thorough comparison). It is possible, see Thm. 7 of App. B, to show that under A2, a condition on the learning rate, and a bounding condition on d_t ($\exists d_\top \geq d_\perp > 0$ such that $\forall s,t,\ d_\top \geq d_t(s) \geq d_\perp$), we have: $t_0 \leq \mathcal{O}\left(\frac{|\mathcal{S}|}{(1-\gamma)^7}\frac{d_\top}{d_\perp}\frac{1}{\min_s \delta(s)}\right)$.

2.3 Related work

[11, 15, 52] study the off-policy, continuing setting paired with the average value objective. That objective is either expressed with a state visitation coming from the behavioral policy (the excursion objective) or from the trained policy (the alternative life objective). Its gradient can be computed exactly or approximated and used to optimize the objective. The weights assigned to each state during an update stem from the gradient computation (exactly as in the discounted return gradient we study). It is an interesting question, left for future work, to understand whether those weights have desirable properties from a convergence standpoint. In particular, it is possible that a condition akin to C4 exists in the continuing objective case.

From a proof technique perspective, [2] studies gradient ascent and, as a byproduct, can rely on standard optimization results: strong convexity and convergence of gradients to 0. We cannot do so as our updates are not gradients anymore. More precisely, Lemma C2 in [2] uses the strong convexity of the objective function to prove that following the gradient results in a value improvement (assuming a learning rate sufficiently small). We provide a more general (we do not need the condition on the learning rate) and, in our admittedly biased opinion, more elegant proof of that lemma in our Theorem 1. Second, their Lemma C5 uses (i) the convergence of the gradient to 0 (a standard result with gradient as-/des-cent), and (ii) the assumption that all states have a non-vanishing density to infer that, in the limit, the learnt policy does not assign any mass to states that have a non-zero advantage. Neither (i) nor (ii) hold in our setting, we had to leverage C4 instead.

3 Dr Jekyll and Mr Hyde: an actor-critic with convergence guarantees

3.1 Conditions for an actor-critic algorithm to converge to optimality

C4 states the necessary and sufficient condition for optimal planning with the exact model. In a reinforcement learning setting however, the model is not known; the policy must be learnt from samples collected in the environment. As a consequence, A1 cannot be made anymore, we need to rely on an exploration condition instead:

C6. Each state-action pair is explored infinitely many times: $\forall s, a,\ \lim_{t\to\infty} n_t(s,a) = \infty$,

where $n_t(s,a)$ is the count of samples collected for the state-action pair (s,a) at time t.

In finite MDPs, C6 provides a necessary and sufficient condition for having an unbiased estimator of the value. As is customary, we call this estimator the critic and denote it $\mathring{q}_t$. C6 is necessary because the required statistical precision can only be achieved when the number of samples tends to infinity; it is sufficient as guaranteed by many off-policy policy evaluation algorithms from the literature [31, 46]. Classic concentration bounds, such as Hoeffding's inequality, tell us that with $\mathcal{O}(1/\xi^2)$ samples in every state, a ξ-accurate critic can be achieved. Under C4 and C6, there thus exists a timestep after which updates based on $\mathring{q}_t$ are sufficiently close to the true ones for a continuity argument to show that the policy improves eventually[4] (as guaranteed by theory under A1).

Our results till now have assumed a finite MDP and models with sufficient capacity. In larger domains, those assumptions do not hold anymore, and therefore our theory does not apply. For instance, [26] proves that using the on-policy undiscounted update $d_{\pi_t,1}$ instead of the policy gradient can lead to highly suboptimal policies under state aliasing. Their counter-example applies to our policy updates as well. Nevertheless, we believe that state aliasing is a worst-case scenario and conjecture that it does not happen to neural networks thanks to their high expressive capacity. Also note that this concern with respect to distribution shift is general and could be formulated for any neural model, including supervised models, or in RL the purely value-based ones such as DQN that are frequently trained off-policy. The consensus in the literature is that neural models do not suffer too much from distribution shift as long as the testing set distribution is well covered by the training set. Since appropriate coverage of the state-action space is actually the final objective of our off-policy policy updates, we expect minimal impact on this dimension, though it does remain to be formally demonstrated. Our empirical results from Section 4.4 support this conjecture. The theoretical study of the function approximation setting is left for future work.

3.2 Our solution to enforce C4 and C6

Enforcing C4 C4 defines the necessary and sufficient condition for asymptotic convergence to optimality. However, we will see in Section 4.2 that in difficult planning tasks, C4 can be insufficient for convergence to happen in a reasonable amount of time. Below, we therefore introduce C4-s, a condition stronger[5] than C4, as well as two techniques for C4-s to be verified:

C4-s. $\forall s \in \mathcal{S}, \quad \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T} \eta_t d_t(s) \geq d_\perp(s) > 0.$

The first technique to enforce C4-s is to apply the approximate expected policy update [39, 20, 10]:

$$\widehat{U}(\theta, s) \doteq \sum_{a\in\mathcal{A}} \mathring{q}(s,a)\nabla_\theta \pi(a|s). \tag{7}$$

The expected policy update is deterministic in the sense that it does not require sampling actions. As there exists a deterministic optimal policy, this in turn implies that the learning rate η_t does not need to satisfy the second Robbins-Monro condition: $\sum_t \eta_t^2 < \infty$, and thus may be constant $\eta_t \doteq \eta$.

The second technique to enforce C4-s is to ensure that the density of updates d_t does not decay to 0 for any state s. This can be obtained by maintaining a constant proportion o_t of off-policy actor updates in order to cover the whole state space. We propose to do so by recording two replay buffers: one with on-policy samples, *i.e.* samples collected with an exploitation policy, and another with off-policy samples, *i.e.* samples collected with an exploration policy.

To the best of our knowledge, this type of prioritized experience replay has never been used in this fashion, nor to this effect. We note that off-policy policy gradient is a concept that exists in the literature, but it refers to the application of policy gradients from batch samples [21]. Following the success of DQN [24], experience replays have also been used in actor-critic methods [49] which bears some resemblance to our suggestion. Finally, off-policy actor-critic with shared prioritized experience replay [34] has been applied to large-scale experiments with distributed agents [14, 35].

Enforcing C6 Many RL algorithms (actor-critic or purely value-based) ensure some form of exploration. They broadly form two groups: dithering exploration (*e.g.*, epsilon greedy, softmax, entropy regularization [47, 42, 13]), and deep exploration (*e.g.*, UCB, thompson sampling, density constraints [28, 3, 29, 32]). Strong cases have been made against dithering exploration, arguing in particular its inability to ensure visits to all states and therefore convergence to optimality [30].

[4] A formal proof would require dealing with several technical challenges due to the stochasticity of the value updates that break the monotonicity of the estimator accuracy. We leave it for future work.

[5] C4-s is identical to C5, but applied to the full state space $\mathcal{S}$ (it thus implies C5).

```
Input: Scheduling of exploration $(\epsilon_t)$, of off-policiness $(o_t)$ and of actor learning rate $(\mathring{\eta}_t)$.
 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 $\infty$ 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 $\tau$ was terminal, then $(\pi_b, D_b) \leftarrow (\tilde{\pi}, \tilde{D})$ w.p. $\epsilon_t$, $(\pi_b, D_b) \leftarrow (\mathring{\pi}, \mathring{D})$ 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 $\tau$.        ▷ e.g. with $Q$-learning trained on UCB rewards
11:         Update Dr Jekyll's critic $\mathring{q}$ on $\tau$.        ▷ e.g. with SARSA update
12:         Expected update of Dr Jekyll's actor $\mathring{\pi}$ in state $s$.        ▷ Eq. (7)
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 updates for Mr Hyde's policy $\tilde{\pi}$ and Dr Jekyll's critic $\mathring{q}$ are underspecified, and may be any algorithm in the literature satisfying the exploration (for Mr Hyde) or unbiased (for Dr Jekyll) conditions. When $(\epsilon_t) = 0$, J&H amounts to on-policy expected updates from a single replay buffer.

In spite of these arguments (that we recall in App. A.5), there are still only very few actor-critic algorithms that realize deep exploration [9, 33]. We conjecture that this relates to the following observations: i) exploration involves a moving objective, hence a non-stationarity of the desired policy, ii) actor-critic algorithms have a structural inertia in their policy (in contrast to value-based methods that can completely switch policy when an action's value overtakes another's). As a consequence, a dual exploration-exploitation actor-critic algorithm takes a lot of time to switch from deep exploration to exploitation, and back, making it inefficient. To avoid this issue, we propose to train two policies, a pure exploration one and a pure exploitation one. As an added benefit, they will be used to constitute the on-policy and off-policy replay buffers introduced in the second technique above.

3.3 Dr Jekyll and Mr Hyde algorithm (J&H)

The objective of this section is to introduce a novel algorithm that satisfies conditions C4 (or C4-s) and C6 by design. To do so, we maintain a mixture of two policies:

- Dr Jekyll $\mathring{\pi}_t$ is a pure exploitation policy,
- Mr Hyde $\tilde{\pi}_t$ is a pure exploration policy.

At the beginning of a new trajectory, Mr Hyde $\tilde{\pi}_t$ (resp. Dr Jekyll $\mathring{\pi}_t$) is chosen with probability ϵ_t (resp. $1 - \epsilon_t$) and used to generate the full trajectory. Dr Jekyll $\mathring{\pi}_t$ is trained with the update rule of Eq. (3), where d_t is defined by prioritized sampling over the experience replays, q_t is replaced with an unbiased estimator $\mathring{q}$ of the value of $\mathring{\pi}_t$, and $\mathring{\theta}_t$ are the parameters of $\mathring{\pi}_t$. $\tilde{\pi}_t$ is a pure exploratory policy, designed to verify almost surely: $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$, $\lim_{t\to\infty} n_{|\tilde{D}|}(s,a) \geq \tilde{d}_\perp > 0$,, where $n_{|\tilde{D}_t|}(s,a)$ is the count of samples (s,a) in its experience replay. We note that a uniform policy satisfies this condition, but with a very small constant $\tilde{d}_\perp$ which compromises convergence in a reasonable time. Any deep exploration algorithm should be guaranteeing it[6]. In our finite MDP experiments, we implement q-learning with UCB rewards [3]: $\tilde{\pi}_t \doteq \text{argmax}_{a\in\mathcal{A}} \tilde{q}_t$ where $\tilde{q}_t$ is trained to predict the expectation of the discounted sum of rewards: $\tilde{r}(s,a) \doteq \frac{1}{\sqrt{n_t(s,a)}}$ (see full Alg.

[6] Efficient exploratory algorithms are a challenge in large environments. The design of such algorithms is beyond the scope of this paper.

3 in App. E). In our deep learning experiments, Hyde is a Double-DQN [48] trained to maximize an exploration bonus based on Random Network Distillation [6] (Section 4.4).

The advantages of separating the exploration from the exploitation policy are the following. First, it is easier to specify the exploration/exploitation trade-off and get full control on the exploration requirements for condition C6: $\forall s, a, \sum_t d_{\pi_t}(s,a) = \infty$. Second, Jekyll's actor $\mathring{\pi}$ is not optimized under a moving objective which would otherwise induce a high level of instability on the policy. Third, one can define the on-policiness/off-policiness o_t by recording two separate replay buffers $\mathring{D}$ and $\tilde{D}$, for trajectories respectively controlled by Dr Jekyll and Mr Hyde. This offers full control on condition C4/C4-s and on the asymptotic behaviour of $\sum_t \eta_t d_t(s)$.

These various observations lead to the design of the Dr Jekyll and Mr Hyde algorithm (J&H), formally detailed in Alg. 1.

4 Experiments

4.1 Domains

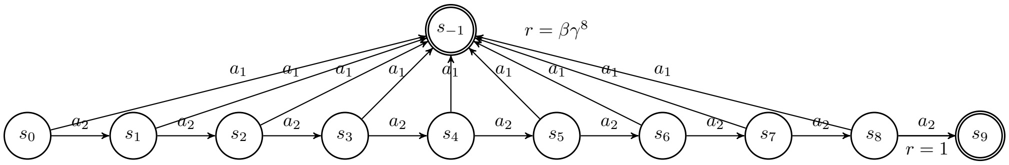


Figure 1: Deterministic chain MDP. Initial state is s_0. Reward is 0 everywhere except when accessing final states s_{-1} and s_9. Reward in s_{-1} is set such that $q(s_0, a_1) = \beta q_\star(s_0, a_2)$, with $\beta \in [0, 1)$.

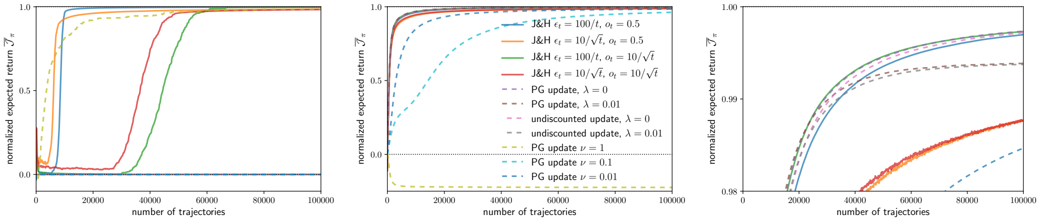
Chain Domain The chain domain is designed to measure the ability of algorithms to overcome immediate rewards pushing the policy gradient towards suboptimal policies. In every state s_k, the agent has the opportunity to play action a_1 and receive an immediate reward of $\beta\gamma^{|\mathcal{S}|-2}$, or to play a_2 and progress to next state s_{k+1} without any immediate reward. A reward of 1 is eventually obtained when reaching state $s_{|\mathcal{S}|-1}$. Fig. 1 represents a chain of size 10. We report the normalized performance: $\overline{\mathcal{J}}_\pi = \frac{\mathcal{J}_\pi - \mathcal{J}_\perp}{\mathcal{J}_\star - \mathcal{J}_\perp}$, where $\mathcal{J}_\star$ is the optimal performance and $\mathcal{J}_\perp \doteq q(s_0, a_1) = \beta\gamma^{|\mathcal{S}|-2}$.

Random MDPs The random MDPs domain is designed to test the algorithms in situations where exploration is not an issue. Indeed, by its design of stochastic transition functions, random MDPs will have a non-null chance to visit every state whatever the behavioural policy. It is therefore a domain where we expect policy gradient updates to perform well, perhaps optimally, and hope that our modified updates still perform comparably. We reproduce the random MDPs environment published in App. B.1.3 of [18]. We report the normalized performance: $\overline{\mathcal{J}}_\pi = \frac{\mathcal{J}_\pi - \mathcal{J}_u}{\mathcal{J}_\star - \mathcal{J}_u}$, where $\mathcal{J}_\star$ is the optimal performance and $\mathcal{J}_u$ is the performance of the uniform policy. The full description of both domains is available in App. C.

4.2 Finite MDP policy planning (A1)

We test performance against time, learning rate η of the actor, MDP parameters $|\mathcal{S}|$ and β, off-policiness o_t, and policy entropy regularization weight λ, with both direct and softmax parametrizations, on the chain and random MDPs. The full report is available in App. D.

On the chain domain, we confirm that enforcing updates with $d_t \doteq o_t d_u + (1 - o_t)\frac{d_{\pi_t,\gamma}}{\|d_{\pi_t,\gamma}\|_1}$ including an off-policy component $o_t > 0$ on a uniform state distribution $d_u(s) \doteq \frac{1}{|\mathcal{S}|}$ helps path discovery and policy planning, while on-policy updates fail at converging to the optimal policy, even with policy entropy regularization. We also observe that while $o_t \in \Omega(t)$ enforces C4, and therefore guarantees convergence to optimality, it may not happen in a reasonable amount of time, and a constant off-policiness is preferable. By sweeping over the chain parameters β and $|\mathcal{S}|$, we observe that on-policy updates are sensitive to both: even with a small $\beta = 0.1$, a reasonably sized chain $|\mathcal{S}| = 15$ cannot be solved. In contrast, $o_t = 0.5$ converges fast to optimality even with $\beta = 0.95$ and $|\mathcal{S}| = 25$.



(a) Chain experiment (b) Random MDPs experiment (c) Random MDPs zoomed in

Figure 2: RL experiments: normalized expected return vs. number of trajectories (200+ runs).

On the random MDPs domain, we observe empirically that discounted and undiscounted updates perform well but oftentimes stagnate at 99% of the optimal performance; d_t with an off-policy component performs better and gets even closer to optimality. Finally, we note that purely uniform updates: $o_t = 1$ slow down training in the random MDPs experiment. We conclude that it is best to include both on-policy and off-policy components in d_t. These experiments also allow us to observe the biased convergence implied by policy entropy regularization.

4.3 Finite MDP reinforcement learning experiments

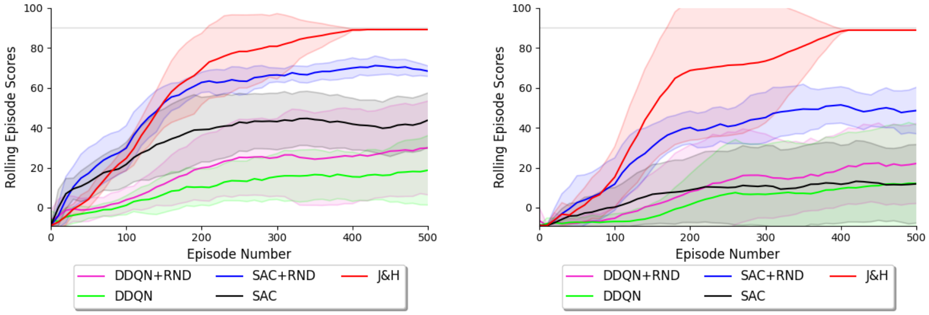
With the softmax parametrization, we tested J&H with various scheduling for ϵ_t and o_t against on-policy (PG/undiscounted) updates, with/without policy entropy regularization (hyperparameter λ), and with/without UCB critic (UCB hyperparameter ν). The full description of the algorithms implementations is available in App. E.1. Our policy planning experiments suggested that o_t was best constant at 0.5, the same holds in our RL experiments. Fig. 2a reveals that on-policy updates with/without policy entropy regularization cannot solve the chain experiment of size 10 with $\beta = 0.8$. In contrast, the task is solved by on-policy updates with a strong UCB bonus $\nu = 1$ for the critic, and by all J&H implementations. Concerning J&H scheduling hyperparameters, we observe that $o_t = 0.5$ allows to identify the optimal policy significantly faster, and $\epsilon_t = \frac{100}{t}$ is sufficient and allows to converge faster asymptotically than $\epsilon_t = \frac{10}{\sqrt{t}}$. Fig. 2b shows that all algorithms are able to solve the random MDP task. However, UCB critics with high $\nu \in \{0.1, 1\}$ (including the only on-policy setting that succeeded on the chain task) fail to do so properly: they eventually will, but explore too much within the experimental time. If we look more closely at the tight convergence of every algorithm on Fig. 2c, we observe that the policy entropy regularization incurs a bias and that off-policiness of updates of J&H does not slow down convergence compared to vanilla on-policy updates. Note that J&H with $\epsilon_t = \frac{10}{\sqrt{t}}$ performs worse only because we report the expected performance of the mixture of Dr Jekyll and Mr Hyde. The performance of Dr Jekyll alone is comparable, and even slightly better than vanilla on-policy updates (see Fig. 9b in App. E).

For completeness, we also test performance against time, learning rates of the actor η and critic η_c, MDP parameters $|\mathcal{S}|$ and β, off-policiness o_t, exploration ϵ_t, and critic initialization q_0 with the softmax parametrization on the chain and random MDPs domains. See App. E for the full report.

4.4 Deep reinforcement learning experiments

To conclude our empirical evaluation, we implemented a deep version of J&H, as described in Algorithm 1, by parametrizing the agents using deep neural networks (see App. F for full details).

Dr Jekyll is a policy network, with a standard architecture, trained using the updates described in Eq. (7). As previously mentioned, any exploration algorithm can be used for Mr Hyde. In our experiments, we chose Random Network Distillation [6, RND] to generate exploration bonuses. Two networks, one random, the target, and a second one, the predictor, are used to assess the novelty of an observed state via the distance between the output of the target and the output of the predictor on that state. Each time a given state is evaluated, the predictor is trained to predict the output of the target. The more a state is seen, the smaller the prediction error will be. The error is used as a reward signal for Mr Hyde, a standard Double-DQN [48, DDQN] trained to maximize it. By doing so, Mr Hyde is incentivized to explore parts of the state space that have not been visited much yet.



(a) Level 1

(b) Level 2

Figure 3: Deep RL experiments: score vs. number of training episodes, averaged over 10 seeds.

We train J&H on a version of the Four Rooms environment [44], a 15x15 grid split into four rooms (see App. F for the exact layout). The agent, placed at random initially, needs to navigate to a fixed goal location, where it is granted a positive reward. For each step taken in the environment, the agent incurs a small negative reward. We consider two levels for the task. In level 1 (the original version of the game), the initial state distribution covers the entire state space, which corresponds to A2 being verified. To make exploration harder, we also consider an initial state distribution that does not contain any state from the room where the goal is located, and call that task level 2. We compare J&H to DDQN and Soft-Actor Critic [13, SAC]. For fair comparison, we also train these agents on the environment rewards augmented with the RND rewards used to train Mr Hyde, represented by the curves labelled DDQN+RND and SAC+RND.

Results can be found in Fig. 3. On level 1, we see that J&H reaches quite fast the maximal score of 90, The baselines do not perform as well and fail to converge to the optimal policy. Interestingly, even with A2 verified, adding an RND bonus led to increased performance. We also notice that overall SAC outperforms DDQN. The same observations can be made on level 2. We note that on some seeds, J&H takes more time to learn the optimal policy, leading to visible plateaus and temporary high variance. However, a powerful property underlined by our experiments and offered by the decoupling of exploration from exploitation is the ability of J&H, unhindered by dithering and/or conflicting reward signals, to converge to optimality. Comparatively, the baselines get stuck in a mixed exploration/exploitation behavior and fail to reach the maximum score.

5 Contributions and limitations

Contributions: We study a planning issue in actor-critic algorithms and tackle it by extending the policy gradient theory to *policy updates* with respect to any state density. Under these generalized policy updates, we show convergence to optimality under a necessary and sufficient condition on the updates' state densities. We also significantly improve previous asymptotic convergence rates. We implement the principles prescribed by our theory in a novel algorithm, Dr Jekyll & Mr Hyde (J&H), with a double personality: Dr Jekyll purely exploits while Mr Hyde purely explores. J&H's independent policies allow to record two separate replay buffers: one on-policy (Dr Jekyll's) and one off-policy (Mr Hyde's), and therefore to update J&H's models with a mixture of on-policy and off-policy updates. Beyond J&H, we define conditions for actor-critic algorithms to satisfy the requirements from our analysis. We extensively test J&H on finite MDPs and deep RL where it demonstrates superior planning abilities, and at least comparable asymptotic rates, to its competitors.

Limitations and future work: On the theory side, our non-asymptotic convergence rates are limited to the softmax and require A2 and C3. Although we are convinced our arguments hold, we did not develop the formal convergence proof in the RL setting (see second paragraph of Section 3.1). Finally, our theory does not tackle function approximation. In that context, we believe techniques from [2] could be useful. On the empirical side, we wish to emphasize that our deep RL experiments are to be considered as a proof of concept of the relevance of J&H's principles in that setting. A thorough study, encompassing more environments and baselines, potentially both in discrete and continuous actions spaces, is left for future work.

Acknowledgments and Disclosure of Funding

We would like to thank Shangtong Zhang for our fruitful discussions and Alessandro Sordoni for helping us presenting and organizing the paper.

References

[1] Joshua Achiam. Spinning up in deep reinforcement learning, 2018.

[2] Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In Jacob Abernethy and Shivani Agarwal, editors, Proceedings of 3rd Conference on Learning Theory (COLT), volume 125 of Proceedings of Machine Learning Research, pages 64–66. PMLR, 09–12 Jul 2020.

[3] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In Proceedings of the 29th Advances in Neural Information Processing Systems (NIPS), pages 1471–1479, 2016.

[4] Jalaj Bhandari and Daniel Russo. On the linear convergence of policy gradient methods for finite mdps. In Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS), pages 2386–2394. PMLR, 2021.

[5] Mathieu Blondel, Akinori Fujino, and Naonori Ueda. Large-scale multiclass support vector machine training via euclidean projection onto the simplex. In Proceedings of the 22nd International Conference on Pattern Recognition (ICPR), pages 1289–1294. IEEE, 2014.

[6] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In Proceedings of the 7th International Conference on Learning Representations (ICLR), 2019.

[7] Itai Caspi, Gal Leibovich, Gal Novik, and Shadi Endrawis. Reinforcement learning coach, December 2017.

[8] Petros Christodoulou. Deep reinforcement learning algorithms with pytorch. https://github.com/p-christ/Deep-Reinforcement-Learning-Algorithms-with-PyTorch, 2019.

[9] Kamil Ciosek, Quan Vuong, Robert Loftin, and Katja Hofmann. Better exploration with optimistic actor critic. In Proceedings of the 32nd Advances in Neural Information Processing Systems (NeurIPS), volume 32. Curran Associates, Inc., 2019.

[10] Kamil Ciosek and Shimon Whiteson. Expected policy gradients. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, volume 32, 2018.

[11] Thomas Degris, Martha White, and Richard S. Sutton. Off-policy actor-critic. In Proceedings of the 29th nternational Conference on Machine Learning (ICML), page 179–186, 2012.

[12] Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. https://github.com/openai/baselines, 2017.

[13] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the 35th International Conference on Machine Learning (ICML), pages 1861–1870. PMLR, 2018.

[14] Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado van Hasselt, and David Silver. Distributed prioritized experience replay. In Proceedings of the 6th International Conference on Learning Representations (ICLR), 2018.

[15] Ehsan Imani, Eric Graves, and Martha White. An off-policy policy gradient theorem using emphatic weightings. In Proceedings of the 31st Advances in Neural Information Processing Systems (NeurIPS), volume 31, 2018.

[16] Vijay R Konda. Actor-critic algorithms. PhD thesis, Massachusetts Institute of Technology, 2002.

[17] Ilya Kostrikov. Pytorch implementations of reinforcement learning algorithms. https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail, 2018.

[18] Romain Laroche, Paul Trichelair, and Rémi Tachet des Combes. Safe policy improvement with baseline bootstrapping. In Proceedings of the 36th International Conference on Machine Learning (ICML), 2019.

[19] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. Rllib: Abstractions for distributed reinforcement learning. In Proceedings of the 35th International Conference on Machine Learning (ICML), 2018.

[20] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In Proceedings of the 4th International Conference on Learning Representations (ICLR, poster), 2016.

[21] Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Off-policy policy gradient with stationary distribution correction. In Ryan P. Adams and Vibhav Gogate, editors, Proceedings of The 35th Uncertainty in Artificial Intelligence Conference (UAI), pages 1180–1190. PMLR, 2020.

[22] Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. 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, 2020.

[23] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Proceedings of the 33rd International Conference on Machine Learning (ICML), 2016.

[24] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 2015.

[25] Kimia Nadjahi, Romain Laroche, and Rémi Tachet des Combes. 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), 2019.

[26] Chris Nota and Philip S. Thomas. Is the policy gradient a gradient? In Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2020.

[27] OpenAI. Openai five. https://openai.com/five/, 2018.

[28] Ronald Ortner and Peter Auer. Logarithmic online regret bounds for undiscounted reinforcement learning. In Proceedings of the 20th Advances in Neural Information Processing Systems (NIPS), volume 19, page 49, 2007.

[29] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In Proceedings of the 29th Advances in Neural Information Processing Systems (NIPS), pages 4026–4034, 2016.

[30] Ian Osband, Benjamin Van Roy, Daniel J Russo, and Zheng Wen. Deep exploration via randomized value functions. Journal of Machine Learning Research, 20(124):1–62, 2019.

[31] Doina Precup. Eligibility traces for off-policy policy evaluation. Computer Science Department Faculty Publication Series, page 80, 2000.

[32] Zengyi Qin, Yuxiao Chen, and Chuchu Fan. Density constrained reinforcement learning, 2021.

[33] Srinjoy Roy, Saptam Bakshi, and Tamal Maharaj. Opac: Opportunistic actor-critic. arXiv preprint arXiv:2012.06555, 2020.

[34] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In Proceedings of the 4th International Conference on Learning Representations (ICLR, poster), 2016.

[35] Simon Schmitt, Matteo Hessel, and Karen Simonyan. Off-policy actor-critic with shared experience replay. In Hal Daumé III and Aarti Singh, editors, Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pages 8545–8554. PMLR, 13–18 Jul 2020.

[36] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In Proceedings of the 32nd International Conference on Machine Learning (ICML), 2015.

[37] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

[38] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. Nature, 2016.

[39] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In Proceedings of the 31st International Conference on Machine Learning (ICML), pages 387–395. PMLR, 2014.

[40] Thiago D. Simão, Romain Laroche, and Rémi Tachet des Combes. Safe policy improvement with estimated baseline bootstrapping. In Proceedings of the 19th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), 2020.

[41] Adam Stooke and Pieter Abbeel. rlpyt: A research code base for deep reinforcement learning in pytorch. arXiv preprint arXiv:1909.01500, 2019.

[42] Richard S Sutton and Andrew G Barto. Reinforcement Learning: An Introduction (2nd Edition). MIT press, 2018.

[43] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In Proceedings of the 13th Advances in Neural Information Processing Systems (NIPS), 2000.

[44] Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. Artificial Intelligence, 112(1):181 – 211, 1999.

[45] Philip Thomas. Bias in natural actor-critic algorithms. In Proceedings of the 31st International Conference on Machine Learning (ICML), 2014.

[46] Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In International Conference on Machine Learning, pages 2139–2148. PMLR, 2016.

[47] Michel Tokic. Adaptive ε-greedy exploration in reinforcement learning based on value differences. In Proceeding of the 33rd Annual German Conference on AI, volume 6359, page 203. Springer, 2010.

[48] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning, 2015. cite arxiv:1509.06461Comment: AAAI 2016.

[49] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. arXiv preprint arXiv:1611.01224, 2016.

[50] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning, 1992.

[51] Shangtong Zhang. Modularized implementation of deep rl algorithms in pytorch. https://github.com/ShangtongZhang/DeepRL, 2018.

[52] Shangtong Zhang, Wendelin Boehmer, and Shimon Whiteson. Generalized off-policy actor-critic. In Proceedings of the 32nd Advances in Neural Information Processing Systems (NeurIPS), volume 32, 2019.

[53] Shangtong Zhang, Romain Laroche, Harm van Seijen, Shimon Whiteson, and Remi Tachet des Combes. A deeper look at discounting mismatch in actor-critic algorithms. arXiv preprint arXiv:2010.01069, 2021.

A MDP notations and basic knowledge on policy gradients

A.1 General notations

- $\Delta_{\mathcal{X}}$ is the simplex over set $\mathcal{X}$.
- $|\mathcal{X}|$ is the cardinality of set $\mathcal{X}$.
- $X \sim d$ means that random variable X is sampled from distribution d.
- $\mathbb{E}[f(X)|X \sim d]$ denotes the expectation of $f(X)$ when random variable X is sampled from distribution d.
- $\mathbb{P}[X > x|Y = y]$ denotes the probability that random variable X is greater than x conditioned to the fact that random variable Y equals y.
- Subscript $\top$ (resp. $\bot$) denotes the maximal (resp. minimal) value in a range. For instance, $r_\top$ denotes the maximal reward.
- $f(\mathcal{X})$ denotes the sum of $f(x)$ over set $\mathcal{X}$: $f(\mathcal{X}) \doteq \sum_{x\in\mathcal{X}} f(x)$.

A.2 Markov Decision Processes

A Markov Decision Process (MDP) is a tuple $m = \langle \mathcal{S}, \mathcal{A}, p, p_0, r, \gamma\rangle$, where:

- $\mathcal{S}$ is the set of states,
- $\mathcal{A}$ is the set of actions,
- $p(s'|s,a) : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ is the probability of accessing state s' after performing action a in state s,
- $p_0 \in \Delta_{\mathcal{S}}$ is the initial state probability,
- $r(s,a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the (possibly stochastic) reward function,
- and γ is the discount factor.

A stochastic policy $\pi(a|s) : \mathcal{S} \to \Delta_{\mathcal{A}}$ determines the probability of performing action a in state s and therefore determines the behaviour of an agent. The objective of a planning algorithm is to maximize the expected return:

$$\mathcal{J}(\pi) \doteq \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t R_t \middle| S_0 \sim p_0, A_t \sim \pi(\cdot|S_t), S_{t+1} \sim p(\cdot|S_t, A_t), R_t \sim r(\cdot|S_t, A_t)\right]. \tag{8}$$

A.3 Values and advantages

More generally, the state (resp. state-action) value function $v_\pi : \mathcal{S} \to \mathbb{R}$ (resp. $q_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$) is the expected return starting from state s (resp. after performing action a in state s), when following policy π afterwards:

$$v_\pi(s) \doteq \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t R_t \middle| S_0 = s, A_t \sim \pi(\cdot|S_t), S_{t+1} \sim p(\cdot|S_t, A_t), R_t \sim r(\cdot|S_t, A_t)\right], \tag{9}$$

$$q_\pi(s,a) \doteq \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t R_t \middle| S_0 = s, A_0 = a, A_t \sim \pi(\cdot|S_t), S_{t+1} \sim p(\cdot|S_t, A_t), R_t \sim r(\cdot|S_t, A_t)\right]. \tag{10}$$

We have several notorious identities, for instance:

$$\mathcal{J}(\pi) = \mathbb{E}\left[v_\pi(S)|S_0 \sim p_0\right], \tag{11}$$

$$v_\pi(s) = \mathbb{E}\left[q_\pi(s,A)|A \sim \pi(\cdot|s)\right], \tag{12}$$

$$q_\pi(s,a) = \mathbb{E}\left[R + \gamma v_\pi(S')|S' \sim p(\cdot|s,a), R \sim r(\cdot|s,a)\right]. \tag{13}$$

We also define the advantage of an action a (resp. a policy π) over a value v in a given state s:

$$\text{adv}_v(s,a) \doteq \mathbb{E}\left[R + \gamma v(S') | S' \sim p(\cdot|s,a), R \sim r(\cdot|s,a)\right] - v(s) \tag{14}$$

$$\text{resp.} \quad \text{adv}_v(s,\pi) \doteq \mathbb{E}\left[\text{adv}_v(s,A) | A \sim \pi(\cdot|s)\right]. \tag{15}$$

By extension, we talk about advantage over a stochastic policy and write adv_π to denote the advantage over the value induced by this stochastic policy $\text{adv}_\pi \doteq \text{adv}_{v_\pi}$. For notational simplicity, we write v_t, q_t, adv_t instead of v_{π_t}, q_{π_t} and adv_{π_t}. A policy π' is said to be advantageous over a value or another policy if its advantage is positive in all states $s \in \mathcal{S}$. During our analysis, we will constantly use the policy improvement theorem:

Theorem 5. *(Policy Improvement Theorem, see, e.g., Eq 4.7 & 4.8 in [42])*
If $\forall s \in \mathcal{S}$, π' is advantageous over π:

$$\text{adv}_\pi(s,\pi') \geq 0, \textit{then } \forall s \in \mathcal{S}, v_{\pi'}(s) \geq v_\pi(s). \tag{16}$$

If the first inequality is strict in a given state s, so is the second inequality on s.

A.4 Densities and policy gradients

For a given stochastic policy π, we define its state visit density as:

$$d_{\pi,\gamma}(s) \doteq \sum_{t=0}^{\infty} \gamma^t \mathbb{P}\left[S_t = s | S_0 \sim p_0, A_t \sim \pi(\cdot|S_t), S_{t+1} \sim p(\cdot|S_t, A_t)\right] \tag{17}$$

The canonical policy gradient is:

$$\nabla_\theta \mathcal{J}(\pi) \doteq \textstyle\sum_s d_{\pi,\gamma}(s) \sum_a q_\pi(s,a) \nabla_\theta \pi(a|s). \tag{18}$$

The update vector often used in practice is:

$$\textstyle\sum_s d_{\pi,1}(s) \sum_a q_\pi(s,a) \nabla_\theta \pi(a|s). \tag{19}$$

We consider a more general update vector

$$U(\theta, d) \doteq \textstyle\sum_s d(s) \sum_a q_\pi(s,a) \nabla_\theta \pi(a|s),$$

where $d : \mathcal{S} \to \mathbb{R}^+$ may be any positive function over $\mathcal{S}$.

The policy gradient methods depend on a parametrization $\theta \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ of the policy. In our finite MDPs theory, we focus on the two most common ones:

- direct: $\pi(a|s) \doteq \theta_{s,a}$ with update: $u_{s,a} = d(s) q_\pi(s,a)$ and projection on the simplex.
- softmax: $\pi(a|s) \doteq \dfrac{\exp(\theta_{s,a})}{\sum_{a'} \exp(\theta_{s,a'})}$ with update: $u_{s,a} = d(s)\pi(a|s)\text{adv}_\pi(s,a)$.

A.5 Limits of on-policy updates with and without policy entropy regularization

The problem with on-policy updates (updates that are issued from a density over the state visitation of the current policy) is the conflicting objective of exploring and converging. If we want to converge, then mechanically the updates on off-density states will dry out. Let us assume that, in order to discover a better policy, the agent needs to go off-policy from state s, and that the state s' where there is a positive advantage that would lead to a policy flip is actually encountered h timesteps after going off-policy. If we assume that exploration is performed through dithering (typically policy entropy regularization in actor-critic algorithms), ie with a small exploration probability ϵ_t to take the action leading to s', then state s' will be visited only with probability ϵ_t^h. As a consequence, to ensure that $\sum_t d_{\pi_t}(s) = \infty$, as we show to be necessary in Thm. 2, ϵ_t would need to decrease in $\Omega(t^{-1/h})$ which is very slow (and h would be a hyperparameter strongly dependent on the environment). The overall issue comes from the myopia of exploration, a problem already observed in other works that promote deep exploration such as [30].

Only a handful of works have investigated deep exploration in actor-critic methods [6, 9]. We believe that they face difficulties due to the inertia of the policy network. Indeed, the policy gradient requires on-policy updates which implies that the policy network must constantly adapt when switching from an exploration strategy to an exploitation strategy and back, which can be time-consuming and inefficient. Our fix consists in splitting the exploration and exploitation behaviour into two policies: Mr Hyde and Dr Jekyll. Mr Hyde performs pure deep exploration and will eventually collect, for each state-action, transitions in the replay buffer in a sufficient amount to obtain statistical significance. Dr Jekyll purely exploits what it learnt from a mixture of on-policy and off-policy policy updates, the latter granting a faster adaptation to the off-policy discoveries made during exploration.

A.6 Comparison of bounds established in Thm. 4 with previous bounds

[22] established (to the best of their and our knowledge) the first convergence-rate result for softmax policy gradient for MDPs. As far as we know, there has not been others since. We start by recalling the differences between our setting and theirs:

- They have an all-time upper bound of the sub-optimality $\mathcal{J}(\pi_\star) - \mathcal{J}(\pi_t) \leq \epsilon$, while we only provide an asymptotic result of the form: $\exists t_0$, such that $\forall t > t_0$, $\mathcal{J}(\pi_\star) - \mathcal{J}(\pi_t) \leq \epsilon$.
- Our result is more general in the sense that it is applicable to any policy update scheme satisfying C4, while theirs only apply to policy gradient in settings (A2 and C3) that guarantee C4 and C5.
- Our theorem assumes A8 (see Rem. 1), requiring that there exists a unique optimal policy (*i.e.* $\nexists\ (s, a_1 \neq a_2) \in \mathcal{S} \times \mathcal{A}^2$, such that $v_\star(s) = q_\star(s, a_1) = q_\star(s, a_2)$). Although [22] states that their approach works with multiple optimal policies, they did not prove it formally and the extension of their proofs from a unique optimal policy to multiple ones is not clear to us. It seems that they would face the same kind of difficulties we have (basically a not well-defined action gap with moving q_t). We discuss the limitations with respect to A8 in more details in Rem. 1 of App. B.2.3.

Now we recall their theorem under our notations:

Theorem 6 (Thm. 4 of [22])**.** *Assuming A1, A2, C3, and A8 hold, then for all* $t \geq 1$*:*

$$v_\star(s) - v_t(s) \leq \frac{16|\mathcal{S}|(r_\top - r_\bot)}{t(1-\gamma)^6 \inf_{s\in\mathcal{S}, t\geq 1} \pi_t(\pi_\star(s)|s)^2} \cdot \left\|\frac{(1-\gamma)d_{\pi_\star,\gamma}}{p_0}\right\|_\infty^2 \cdot \left\|\frac{1}{p_0}\right\|_\infty, \qquad (20)$$

where $\pi_\star : \mathcal{S} \to \mathcal{A}$ *is an arbitrary deterministic optimal policy.*

And we recall our result under A1, C5, and A8:

$$\exists t_0, \text{ such that } \forall t \geq t_0, \quad v_\star(s) - v_t(s) \leq \frac{8|\mathcal{A}|(v_\top - v_\bot)}{(t-t_0)(1-\gamma)\min_{s\in\text{supp}(d_\star)} \delta(s) e_\bot(s)}. \qquad (21)$$

C5. $\forall s \in \text{supp}(d_\star), \lim_{t\to\infty} \frac{1}{t}\sum_{t'=0}^{t} \eta_{t'} d_{t'}(s) \doteq e_\bot(s) > 0,$

Now we look at each classic dependency:

- Time t: both are in $\mathcal{O}(t^{-1})$ but ours uses an offset t_0, hence its qualification of asymptotic convergence rate.
- State set size $|\mathcal{S}|$: we have a hidden dependency in $\min_{s\in\text{supp}(d_\star)} \delta(s)e_\bot(s)$ which is in $\Omega(1/|\mathcal{S}|)$, so our bound is in $\mathcal{O}(|\mathcal{S}|)$. Their theorem has an additional state set size dependency in $\|\frac{1}{p_0}\|_\infty$, so their bound is in $\mathcal{O}(|\mathcal{S}|^2)$.
- Action set size $|\mathcal{A}|$: we are in $\mathcal{O}(|\mathcal{A}|)$. Their theorem has an implicit action set size dependency in $\inf_{s\in\mathcal{S},t\geq 1} \pi_t(\pi_\star(s)|s)$, so their bound is in $\mathcal{O}(|\mathcal{A}|^2)$.
- Horizon/discount factor $\frac{1}{1-\gamma}$: we have a possible hidden dependency in $v_\top - v_\bot$ (depending on e.g. the sparsity of rewards), which yields a bound in $\mathcal{O}((1-\gamma)^{-2})$, to be compared with their dependency in $\mathcal{O}((1-\gamma)^{-6})$. They pay the cost for setting through C3 the learning rate η in $\mathcal{O}((1-\gamma)^{-3})$. We interpret the last $1-\gamma$ difference by the fact that we only deal with asymptotic rates and do not have to account for propagation through the MDP. Note that while the horizon appears in $\|\frac{(1-\gamma)d_{\pi_\star,\gamma}}{p_0}\|_\infty^2$, this term is larger than 1 (it is the sup of a ratio of distributions), hence the $1-\gamma$ cannot be taken out.

- State initialization distribution p_0: there lies the beauty of policy updates, we do not have any dependency on it. However, we do show that policy gradient will not converge to the optimal solution in general if A2 is not satisfied, and in that case off-policy updates must be used. The dependency in p_0 is particularly strong: possibly cubic with $\|\frac{1}{p_0}\|_\infty$. The worst case happens in $\|\frac{(1-\gamma)d_{\pi_\star,\gamma}}{p_0}\|_\infty^2$ when a state is rarely seen at initialization but visited a lot by optimal policies .

Finally, we discuss the constants that are difficult to compare:

- Divergence from the optimal policy during training $\inf_{s\in\mathcal{S},t\geq 1}\pi_t(\pi_\star(s)|s)$: this is a uncontrolled term that shows that policy gradient is very sensitive to preliminary convergence to sub-optimal solution. This term can easily take infinitesimal values (see *e.g.* Fig. 1d of [22]), and the dependency is squared.
- Optimal action gap $\delta(s)$: we depend on this constant because we first prove the convergence speed in policy and only then in value. If the action gap is very small in some states, resulting in a loose bound, it may be useful to use different proof techniques to avoid the convergence in policy step.
- State density of policy updates $e_\perp(s)$: with a constant factor κ on uniform policy updates, $e_\perp(s) \doteq \frac{\kappa}{|\mathcal{S}|}$, and we accounted for this hidden dependency in the state set size comparison above.

B Theoretical results

B.1 Assumptions and conditions

A1 The model p_0, p, r is known,

A2 The initial state-distribution covers the full state space: $\forall s \in \mathcal{S}, p_0(s) > 0$,

C3 The learning rate is constant: $\eta_t = \frac{(1-\gamma)^3}{2\gamma|\mathcal{A}|}$ (direct) and $\eta_t = \frac{(1-\gamma)^3}{8}$ (softmax),

C4. Each state s is updated with weights that sum to infinity over time: $\sum_{t=0}^{\infty} \eta_t d_t(s) = \infty$,

C4-s. $\forall s \in \mathcal{S}, \quad \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \eta_t d_t(s) \geq d_\perp(s) > 0$.

C5. $\forall s \in \text{supp}(d_\star), \lim_{t\to\infty} \frac{1}{t} \sum_{t'=0}^{t} \eta_{t'} d_{t'}(s) \doteq e_\perp(s) > 0$,

C6. Each state-action pair is explored infinitely many times: $\forall s, a, \lim_{t\to\infty} n_t(s,a) = \infty$.

C7. There exists $d_\top \geq d_\perp > 0$ such that $\forall s, t, d_\top \geq d_t(s) \geq d_\perp$.

A8. The optimal policy is unique: $\forall s, \; q_\star(s,a_1) = q_\star(s,a_2) = v_\star(s)$ implies $a_1 = a_2$.

B.2 Proofs

B.2.1 Monotonicity and convergence

Theorem 1 (Monotonicity under the direct and softmax parametrization). *Under A1, the sequence of value functions $q_t \doteq q_{\pi_t}$ and $v_t \doteq v_{\pi_t}$ are monotonously increasing.*

Proof sketch. First, we prove that, with both direct and softmax parametrizations, the update defined in Eq. (3) leads to a policy that is advantageous over the original one. By the policy improvement theorem, it means that the value function v_t must be increasing, which also implies that q_t is increasing. □

Proof. The direct and softmax parametrizations are treated independently in Theorems 1a and 1b. □

Theorem 1a (Monotonicity under the direct parametrization). *Under A1, the sequence of value functions $q_t \doteq q_{\pi_t}$ and $v_t \doteq v_{\pi_t}$ are monotonously increasing.*

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

$$\pi_{t+1}(\cdot|s) = \mathcal{P}_{\Delta(\mathcal{A})}(\pi_t(\cdot|s) + \eta d(s) q_t(s,\cdot)). \tag{22}$$

By the definition of projections, this gives us:

$$\begin{array}{rrcl}
 & \|\pi_t(\cdot|s) + \eta d(s) q_t(s,\cdot) - \pi_{t+1}(\cdot|s)\|^2 & \leq & \|\pi_t(\cdot|s) + \eta d(s) q_t(s,\cdot) - \pi_t(\cdot|s)\|^2 \quad (23) \\
\iff & 2\langle \pi_t(\cdot|s) - \pi_{t+1}(\cdot|s), \eta d(s) q_t(s,\cdot)\rangle \; + \; \|\pi_t(\cdot|s) - \pi_{t+1}(\cdot|s)\|^2 & \leq & 0 \quad (24) \\
\iff & \frac{1}{2}\|\pi_t(\cdot|s) - \pi_{t+1}(\cdot|s)\|^2 & \leq & \langle \pi_{t+1}(\cdot|s) - \pi_t(\cdot|s), \eta d(s) q_t(s,\cdot)\rangle \quad (25) \\
\implies & 0 & \leq & \langle \pi_{t+1}(\cdot|s) - \pi_t(\cdot|s), \eta d(s) q_t(s,\cdot)\rangle. \quad (26)
\end{array}$$

Using the fact that $d(s) > 0$ and $\eta > 0$ and replacing the vectors by their values allows to infer:

$$\text{adv}_t(s, \pi_{t+1}) \doteq \sum_{a\in\mathcal{A}} (\pi_{t+1}(a|s) - \pi_t(a|s)) q_t(s,a) \geq 0, \tag{27}$$

which is true for all s and therefore allows to apply the policy improvement theorem, which concludes the proof of the monotonicity of the value through the direct parametrization policy update. □

Theorem 1b (Monotonicity under the softmax parametrization). *Under A1, the sequence of value functions $q_t \doteq q_{\pi_t}$ and $v_t \doteq v_{\pi_t}$ are monotonously increasing.*

Proof. This lemma is a generalization of Lemma C.2 in [2], relaxing the assumption on the learning rate, as well as extending the result to our update rule. By the performance difference lemma (see e.g. Lemma 3.2 in [2]), it is sufficient to show that:

$$\forall s \in \mathcal{S}, \quad \text{adv}_t(s, \pi_{t+1}) \doteq \sum_a \pi_{t+1}(a|s)\text{adv}_t(s,a) \geq 0. \tag{28}$$

Let us define:

$$\mathcal{A}_s^\top = \{a \in \mathcal{A} | \text{adv}_t(s,a) \geq 0\}, \tag{29}$$

$$\mathcal{A}_s^\perp = \{a \in \mathcal{A} | \text{adv}_t(s,a) < 0\}. \tag{30}$$

For any s, we have (the weights in the exponential have all increased, and the advantages are non-negative):

$$\sum_{a \in \mathcal{A}_s^\top} e^{(\theta_t)_{s,a} + \eta_t d_t(s) \pi_t(a|s)\text{adv}_t(s,a)} \text{adv}_t(s,a) \geq \sum_{a \in \mathcal{A}_s^\top} e^{(\theta_t)_{s,a}} \text{adv}_t(s,a). \tag{31}$$

Similarly (the weights in the exponential have all decreased, and the advantages are negative):

$$\sum_{a \in \mathcal{A}_s^\perp} e^{(\theta_t)_{s,a} + \eta_t d_t(s) \pi_t(a|s)\text{adv}_t(s,a)} \text{adv}_t(s,a) \geq \sum_{a \in \mathcal{A}_s^\perp} e^{(\theta_t)_{s,a}} \text{adv}_t(s,a). \tag{32}$$

Summing gives:

$$\sum_a e^{(\theta_{t+1})_{s,a}} \text{adv}_t(s,a) \geq \sum_a e^{(\theta_t)_{s,a}} \text{adv}_t(s,a) = 0. \tag{33}$$

Normalizing the left-hand side by $\sum_a e^{(\theta_{t+1})_{s,a}}$ gives the policy and concludes the proof. □

Corollary 1 (Convergence under the direct and softmax parametrization). *Under A1, the sequence of value functions q_t uniformly converges: $q_\infty \doteq \lim_{t\to\infty} q_t$.*

Proof. Since $\forall s, a, t, q_t(s,a) \leq \frac{1}{1-\gamma} r_\top$, the monotonous convergence theorem guarantees the existence of $q_\infty \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ that is the limit of the sequence of q_t. □

B.2.2 Optimality

Theorem 2 (Optimality under the direct and softmax parametrization). *Under A1, the following condition:*

> *C4. 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 optimality: $q_\infty = q_\star \doteq \max_{\pi\in\Pi} q_\pi$.

Proof sketch. In both direct and softmax parametrizations, we assume that there exists a state-action pair (s,a) that is advantageous over the state value limits q_∞ and $v_\infty \doteq \lim_{t\to\infty} v_t$: $\text{adv}_\infty(s,a) \doteq q_\infty(s,a) - v_\infty(s) > 0$. Then, we prove that in this state, the policy improvement yielded by the update is lower bounded by a linear function over the update weight $\eta_t d_t(s)$ applied in state s. Next, we sum over t and notice that this lower bounding sum diverges to infinity, which contradicts Corollary 1.

We may therefore infer that there cannot exist a state-action pair that is advantageous over q_∞ and v_∞, which allows us to conclude that no policy improvement is possible, and thus that the values are optimal: $q_\infty = \max_{\pi\in\Pi} q_\pi$.

For the necessity of C4, we show that the parameter update is upper bounded in both parametrizations by a term linear in the action gap. By choosing the reward function adversarially, we may set it sufficiently small so that the sum of all the gradient steps are insufficient to reach optimality. □

Proof. The sufficient conditions for direct and softmax parametrizations are treated independently in Thm. 2a and 2b. The necessary condition is proven in Thm. 2c. □

Theorem 2a (Optimality under the direct parametrization). *Under A1 and C4, the sequence of value functions q_t converges to optimality:*

$$q_\infty = q_\star \doteq \max_{\pi\in\Pi} q_\pi. \tag{34}$$

Proof. Let us assume that $v_\infty < v_\star$, 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 $q_t \to_{t\to\infty} q_\infty$, 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 $t \geq t_0$:

$$\forall a \in \mathcal{A}_\top(s), q_t(s, a) \geq q_\infty(s, a_\top) - \frac{\delta(s)}{2}, \tag{35}$$

$$\forall a \in \mathcal{A}_\perp(s), q_t(s, a) \leq q_\infty(s, a_\top) - \delta(s). \tag{36}$$

Remembering that

$$\pi_{t+1}(\cdot|s) \doteq \mathcal{P}_{\Delta(\mathcal{A})}\left(\pi_t(\cdot|s) + \eta_t d_t(s) q_t(s, \cdot)\right), \tag{37}$$

the above inequalities allow us to apply Lemma 1 to $\pi_t(\cdot|s)$ with $\mathcal{A}_\top(s)$ the set of coordinates $\{1, \dots, k\}$, $\alpha = \eta_t d_t(s)(q_\infty(s, a_\top) - \frac{\delta(s)}{2})$ and $\beta = \eta_t d_t(s)(q_\infty(s, a_\top) - \delta(s))$, giving:

$$\sum_{a\in\mathcal{A}_\top(s)} \pi_{t+1}(a|s) \geq \min\left(1, \sum_{a\in\mathcal{A}_\top(s)} \pi_t(a|s) + \frac{\alpha - \beta}{2}\right) \tag{38}$$

$$= \min\left(1, \sum_{a\in\mathcal{A}_\top(s)} \pi_t(a|s) + \eta_t d_t(s)\frac{\delta(s)}{4}\right). \tag{39}$$

By assumption, we know that $\sum_{a\in\mathcal{A}_\top(s)} \pi_t(a|s) \leq \lim_{t'\to\infty} \sum_{a\in\mathcal{A}_\top(s)} \pi_{t'}(a|s) < 1$, we may conclude that $\forall t \geq t_0$:

$$\sum_{a\in\mathcal{A}_\top(s)} \pi_{t+1}(a|s) - \sum_{a\in\mathcal{A}_\top(s)} \pi_t(a|s) \geq \eta_t d_t(s)\frac{\delta(s)}{4}. \tag{40}$$

Hence,

$$\lim_{t'\to\infty} \sum_{a\in\mathcal{A}_\top(s)} \pi_{t'}(a|s) - \sum_{a\in\mathcal{A}_\top(s)} \pi_{t_0}(a|s) \geq \sum_{t'=t_0}^{\infty} \eta_{t'} d_{t'}(s)\frac{\delta(s)}{4}. \tag{41}$$

Since $\sum_{t'=t}^{\infty} \eta_{t'} d_{t'}(s) = \infty$ and the left-hand side is bounded, this contradicts the assumption that there exists an advantageous action in state s. This property must be true in every state s, we may therefore conclude that v_∞ is optimal. □

Theorem 2b (Optimality under the softmax parametrization). *Under A1 and C4, the sequence of value functions q_t converges to optimality:*

$$q_\infty = q_\star \doteq \max_{\pi\in\Pi} q_\pi. \tag{42}$$

Proof sketch. This theorem generalizes Thm. 5.1 of [2] and our proof borrows many of their ideas. The generalization could have been made by adapting some of the lemmas their proof relies on, but we believe that the full rewriting of the proof has its merits: it is self contained, and ends up being significantly shorter (3 pages versus 10 pages). In this proof sketch, we stack without proving them the arguments used in the full proof that follows.

We start by fixing state s and by partitioning the action set according to their value $q_\infty(s,a)$ in s as compared to $v_\infty(s)$: smaller $\mathcal{A}_s^<$, equal $\mathcal{A}_s^=$, or larger $\mathcal{A}_s^>$. We assume towards a contradiction that there must be some s for which $\mathcal{A}_s^>$ is non-empty, as otherwise, the policy improvement theorem tells us that v_∞ is optimal. In such a state, by the uniform convergence theorem, there is a timestep t_0 from which for all $t \geq t_0$, the values q_t are ξ-close to q_∞. We choose ξ to be smaller than any action gap in state s. We then notice that if an action in $\mathcal{A}_s^=$ gets a policy smaller than that of an action in $\mathcal{A}_s^>$, then it remains this way in the future (it undergoes smaller updates). We further partition $\mathcal{A}_s^=$: $\mathcal{A}_s^{\overset{\leq}{=}}$ contains the actions for which the policy gets smaller than the policy in all actions of $\mathcal{A}_s^>$, $\mathcal{A}_s^{\overset{\geq}{=}}$ contains the actions for which this event never happens. We call $t_1 \geq t_0$, the timestep when $\forall a^{\overset{\leq}{=}} \in \mathcal{A}_s^{\overset{\leq}{=}}, a^> \in \mathcal{A}_s^>, \pi_{t_1}(a^{\overset{\leq}{=}}|s) \leq \pi_{t_1}(a^>|s)$.

Then, we observe that $\mathcal{A}_s^{\overset{\geq}{=}}$ cannot be empty because this set is the only one that contains actions that may prevent full convergence on $\mathcal{A}_s^>$. We then look at the evolution of the ratio of policy mass on $\mathcal{A}_s^>$ versus $\mathcal{A}_s^< \cup \mathcal{A}_s^{\overset{\leq}{=}}$ and show that this ratio is monotonously increasing. We then establish an upper bound on it by arguing that otherwise it would imply that the sum of the updates on $\mathcal{A}_s^{\overset{\geq}{=}}$ would become negative. This cannot happen because actions in $\mathcal{A}_s^{\overset{\geq}{=}}$ need to diverge to infinity to get some policy mass, and none of them can compensate by diverging to $-\infty$, as it would otherwise have been partitioned in $\mathcal{A}_s^{\overset{\leq}{=}}$. We conclude the proof by showing that if the parameter of an action in $\mathcal{A}_s^{\overset{\geq}{=}}$ diverges to $+\infty$, then the parameters of actions in $\mathcal{A}_s^>$ will also diverge to $+\infty$. This implies the aforementioned ratio diverges to ∞ contradicting it being upper bounded. Hence $\mathcal{A}_s^>$ must be empty in every state and consequently v_∞ optimal. □

Full proof. We recall the update rule with softmax parametrization:

$$(\theta_{t+1})_{s,a} = (\theta_t)_{s,a} + \eta_t d_t(s)\pi_t(a|s)\text{adv}_t(s,a). \tag{43}$$

We start from Corollary 1 that proves the existence of $v_\infty(s)$ and $q_\infty(s,a)$ to which $v_t(s)$ and $q_t(s,a)$ converge from below for any state and action. We then consider the following partition of the actions in state s:

$$\mathcal{A}_s^> = \{a \in \mathcal{A} | q_\infty(s,a) > v_\infty(s)\} \tag{44}$$

$$\mathcal{A}_s^= = \{a \in \mathcal{A} | q_\infty(s,a) = v_\infty(s)\} \tag{45}$$

$$\mathcal{A}_s^< = \{a \in \mathcal{A} | q_\infty(s,a) < v_\infty(s)\}. \tag{46}$$

If $\forall s,\ \mathcal{A}_s^> = \emptyset$, the policy improvement theorem tells us that q_∞ is optimal and our work is done.

We thus assume towards a contradiction that $\mathcal{A}_s^>$ is non-empty for a certain state s. We define $\delta_\infty(s)$ as the smallest positive improvement of q_∞ over v_∞ in state s:

$$\delta_\infty(s) \coloneqq \min_{a \in \mathcal{A}_s^>} q_\infty(s,a) - v_\infty(s). \tag{47}$$

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

$$v_\infty(s) - \frac{1}{2}\xi \leq v_t(s) \leq v_\infty(s), \tag{48}$$

$$q_\infty(s,a) - \xi \leq q_t(s,a) \leq q_\infty(s,a). \tag{49}$$

Moving forward, we choose $\xi \leq \frac{1}{2}\min\{\delta_\infty(s), v_\infty(s) - \max_{a \in \mathcal{A}_s^<} q_\infty(s,a)\}$ and let t_0 be the corresponding timestep.

This allows us to further partition the set $\mathcal{A}_s^=$ as:

$$\mathcal{A}_s^{\overset{\leq}{=}} \doteq \{a \in \mathcal{A}_s^= | \exists t_a \geq t_0, \forall t' \geq t_a, \forall a^> \in \mathcal{A}_s^>, \theta_{t'}(s,a^>) \geq \theta_{t'}(s,a)\} \tag{50}$$

$$\mathcal{A}_s^{\overset{\geq}{=}} \doteq \mathcal{A}_s^= \setminus \mathcal{A}_s^{\overset{\leq}{=}}. \tag{51}$$

We note that trivially: $\mathcal{A} = \mathcal{A}_s^{\overset{\leq}{=}} \cup \mathcal{A}_s^{\overset{\geq}{=}} \cup \mathcal{A}_s^> \cup \mathcal{A}_s^<$. We let $t_1 = \max_{a \in \mathcal{A}_s^{\overset{\leq}{=}}} t_a$ and in the following consider $t \geq t_1$. We recall the notation $f(\mathcal{X}) \doteq \sum_{x \in \mathcal{X}} f(x)$ for any function f and set $\mathcal{X}$.

The updates $u_t(s,a) \doteq \pi_t(a|s)\text{adv}_t(s,a)$ on those various sets can be bounded as follows.

On $\mathcal{A}_s^{>}$:

$$
\begin{aligned}
u_t(s, \mathcal{A}_s^{>}) &\doteq \sum_{a^{>} \in \mathcal{A}_s^{>}} u_t(s, a^{>}) && (52)\\
&= \sum_{a^{>} \in \mathcal{A}_s^{>}} \pi_t(a^{>}|s)(q_t(s, a^{>}) - v_t(s)) && (53)\\
&\geq ((v_\infty(s) + \delta_\infty(s) - \xi) - v_\infty(s)) \sum_{a^{>} \in \mathcal{A}_s^{>}} \pi_t(a^{>}|s) && (54)\\
&= (\delta_\infty(s) - \xi)\pi_t(\mathcal{A}_s^{>}|s). && (55)
\end{aligned}
$$

On $\mathcal{A}_s^{\geqq}$:

$$
\begin{aligned}
u_t(s, \mathcal{A}_s^{\geqq}) &\doteq \sum_{a^{\geqq} \in \mathcal{A}_s^{\geqq}} u_t(s, a^{\geqq}) && (56)\\
&= \sum_{a^{\geqq} \in \mathcal{A}_s^{\geqq}} \pi_t(a^{\geqq}|s) \left(\sum_{b^{<} \in \mathcal{A}_s^{<} \cup \mathcal{A}_s^{\leqq}} \pi_t(b^{<}|s)(q_t(s, a^{\geqq}) - q_t(s, b^{<})) \right. \\
&\qquad + \sum_{b^{>} \in \mathcal{A}_s^{>}} \pi_t(b^{>}|s)(q_t(s, a^{\geqq}) - q_t(s, b^{>})) && (57)\\
&\qquad \left. + \sum_{b^{\geqq} \in \mathcal{A}_s^{\geqq}} \pi_t(b^{\geqq}|s)(q_t(s, a^{\geqq}) - q_t(s, b^{\geqq})) \right)\\
&\leq \sum_{a^{\geqq} \in \mathcal{A}_s^{\geqq}} \pi_t(a^{\geqq}|s) \left(\pi_t(\mathcal{A}_s^{<}|s) + \pi_t(\mathcal{A}_s^{\leqq}|s) \right) (v_\infty(s) - v_\perp)\\
&\quad - \sum_{a^{\geqq} \in \mathcal{A}_s^{\geqq}} \pi_t(a^{\geqq}|s)\pi_t(\mathcal{A}_s^{>}|s)(\delta_\infty(s) - \xi) && (58)\\
&\quad + \sum_{(a^{\geqq}, b^{\geqq}) \in \mathcal{A}_s^{\geqq} \times \mathcal{A}_s^{\geqq}} \pi_t(a^{\geqq}|s)\pi_t(b^{\geqq}|s)(q_t(s, a^{\geqq}) - q_t(s, b^{\geqq})) && (59)\\
&= \pi_t(\mathcal{A}_s^{\geqq}|s) \left(\left(\pi_t(\mathcal{A}_s^{<}|s) + \pi_t(\mathcal{A}_s^{\leqq}|s) \right) (v_\infty(s) - v_\perp) - \pi_t(\mathcal{A}_s^{>}|s)(\delta_\infty(s) - \xi) \right) && (60)\\
&\leq \left(1 - \pi_t(\mathcal{A}_s^{\geqq}|s) \right) (v_\infty(s) - v_\perp). && (61)
\end{aligned}
$$

Finally, on $\mathcal{A}_s^{<}$:

$$
u_t(s, \mathcal{A}_s^{<}) \doteq \sum_{a^{<} \in \mathcal{A}_s^{<}} u_t(s, a^{<}) \leq 0. \tag{62}
$$

This gives:

$$
\begin{aligned}
(\theta_{t'})_{s, \mathcal{A}_s^{>}} &\geq (\theta_t)_{s, \mathcal{A}_s^{>}} + (\delta_\infty(s) - \xi) \sum_{t''=t}^{t'-1} \eta_{t''} d_{t''}(s) \pi_{t''}(\mathcal{A}_s^{>}|s), && (63)\\
(\theta_{t'})_{s, \mathcal{A}_s^{\geqq}} &\leq (\theta_t)_{s, \mathcal{A}_s^{\geqq}} + (v_\infty(s) - v_\perp) \sum_{t''=t}^{t'-1} \eta_{t''} d_{t''}(s) \left(1 - \pi_{t''}(\mathcal{A}_s^{\geqq}|s) \right), && (64)\\
(\theta_{t'})_{s, \mathcal{A}_s^{<}} &\leq (\theta_t)_{s, \mathcal{A}_s^{<}}. && (65)
\end{aligned}
$$

We treat the case of $\mathcal{A}_s^{\leqq}$ slightly differently, and will instead use the following inequalities.

$\forall a^{>} \in \mathcal{A}_s^{>}$, $a^{\leqq} \in \mathcal{A}_s^{\leqq}$, $\theta_t(s, a^{>}) \geq \theta_t(s, a^{\leqq})$, $\pi_t(s, a^{>}) \geq \pi_t(s, a^{\leqq})$, and $u_t(s, a^{>}) \geq 2u_t(s, a^{\leqq})$.

The first two inequalities stem directly by construction of $\mathcal{A}_s^{\overset{\leq}{=}}$. For the last one, we have:

$$\begin{aligned}
u_t(s,a^>) = \pi_t(a^>|s)\text{adv}_t(s,a^>) &\geq \pi_t(s,a^{\overset{\leq}{=}})\text{adv}_t(s,a^>) \\
&= \pi_t(s,a^{\overset{\leq}{=}})(q_t(s,a^>) - v_t(s)) \\
&\geq \pi_t(s,a^{\overset{\leq}{=}})(v_\infty(s) + \delta_\infty - \xi - v_t(s)) \\
&\geq \pi_t(s,a^{\overset{\leq}{=}})(\delta_\infty - \xi) \\
&\geq \xi\pi_t(s,a^{\overset{\leq}{=}}) \\
&\geq 2\pi_t(s,a^{\overset{\leq}{=}})\text{adv}_t(s,a^{\overset{\leq}{=}}) = 2u_t(s,a^{\overset{\leq}{=}}). \qquad (66)
\end{aligned}$$

In the last line, we used: $\text{adv}_t(s,a^{\overset{\leq}{=}}) = q_t(s,a^{\overset{\leq}{=}}) - v_t(s) \leq q_\infty(s,a^{\overset{\leq}{=}}) - v_t(s) = v_\infty(s) - v_t(s) \leq \frac{\xi}{2}$.

$\mathcal{A}_s^{\overset{\geq}{=}} \neq \emptyset$. If $\mathcal{A}_s^{\overset{\geq}{=}}$ is empty, Eq. (63), Eq. (65) and Eq. (66) imply that $\pi_t(\mathcal{A}_s^>|s)$ increases monotonously with t. C4 then guarantees that $(\theta_{t'})_{s,\mathcal{A}_s^>}$ diverges to ∞. Consequently, $\pi_t(\mathcal{A}_s^>|s)$ will converge to 1 and $v_t(s)$ will tend to a value larger than $v_\infty(s) + \delta_\infty(s)$ which is impossible. Moving forward, we consider $\mathcal{A}_s^{\overset{\geq}{=}} \neq \emptyset$ non-empty.

Mass on $\mathcal{A}_s^>$ and $\mathcal{A}_s^< \cup \mathcal{A}_s^{\overset{\leq}{=}}$. We now look at the evolution of $\text{ratio}_t(\mathcal{A}_s^>, \mathcal{A}_s^< \cup \mathcal{A}_s^{\overset{\leq}{=}}) \doteq \frac{\pi_t(\mathcal{A}_s^>|s)}{\pi_t(\mathcal{A}_s^< \cup \mathcal{A}_s^{\overset{\leq}{=}}|s)}$:

$$\begin{aligned}
\text{ratio}_{t+1}(\mathcal{A}_s^>, \mathcal{A}_s^< \cup \mathcal{A}_s^{\overset{\leq}{=}}) &= \frac{\sum_{a\in\mathcal{A}_s^>} \exp\left((\theta_{t+1})_{s,a}\right)}{\sum_{a\in\mathcal{A}_s^< \cup \mathcal{A}_s^{\overset{\leq}{=}}} \exp\left((\theta_{t+1})_{s,a}\right)} && (67) \\
&= \frac{\sum_{a\in\mathcal{A}_s^>} \exp\left((\theta_t)_{s,a} + u_t(s,a)\right)}{\sum_{a\in\mathcal{A}_s^< \cup \mathcal{A}_s^{\overset{\leq}{=}}} \exp\left((\theta_t)_{s,a} + u_t(s,a)\right)} && (68) \\
&\geq \frac{\sum_{a\in\mathcal{A}_s^>} \exp\left((\theta_t)_{s,a}\right)}{\sum_{a\in\mathcal{A}_s^< \cup \mathcal{A}_s^{\overset{\leq}{=}}} \exp\left((\theta_t)_{s,a}\right)} && (69) \\
&\geq \text{ratio}_t(\mathcal{A}_s^>, \mathcal{A}_s^< \cup \mathcal{A}_s^{\overset{\leq}{=}}) && (70)
\end{aligned}$$

Eq. (70) establishes the monotonicity of the ratio and shows that $\forall t' \geq t$:

$$\pi_{t'}(\mathcal{A}_s^>|s) \geq \pi_{t'}(\mathcal{A}_s^< \cup \mathcal{A}_s^{\overset{\leq}{=}}|s)\text{ratio}_t(\mathcal{A}_s^>, \mathcal{A}_s^< \cup \mathcal{A}_s^{\overset{\leq}{=}}). \qquad (71)$$

Eq. (60) remains positive. Eq. (70) allows us to show that Eq. (60) cannot become negative. Let us assume towards a contradiction that it does become negative at a given time t, ie

$$\left(\pi_t(\mathcal{A}_s^<|s) + \pi_t(\mathcal{A}_s^{\overset{\leq}{=}}|s)\right)(v_\infty(s) - v_\perp) - \pi_t(\mathcal{A}_s^>|s)(\delta_\infty(s) - \xi) \leq 0.$$

The monotonicity of the above ratio then guarantees that it will remain negative for all $t' > t$. In other words, for all $t' > t$, $u_{t'}(s, \mathcal{A}_s^{\overset{\geq}{=}}) \leq 0$.

If the sum of updates on actions in $\mathcal{A}_s^{\overset{\geq}{=}}$ is negative, two things can happen:

- Either the actions in $\mathcal{A}_s^{\overset{\geq}{=}}$ all have parameters θ that are upper bounded. This means that $\pi_t(\mathcal{A}_s^>|s)$ will eventually converge to 1. In this case, we can reuse the arguments used to disprove $\mathcal{A}_s^{\overset{\geq}{=}} = \emptyset$, and reach the same contradiction.
- Or at least one action in $\mathcal{A}_s^{\overset{\geq}{=}}$ has a parameter θ whose $\limsup$ is $+\infty$. In that case, since the sum of updates on the set $\mathcal{A}_s^{\overset{\geq}{=}}$ is negative, at least one action in $\mathcal{A}_s^{\overset{\geq}{=}}$ must have a θ whose $\liminf$ is $-\infty$. Letting a denote that action, there thus exists a time t at which $\forall a^> \in \mathcal{A}_s^>$, $\theta_t(s,a^>) > \theta_t(s,a)$. Following the reasoning from Eq. (66), the updates to those various actions will guarantee that the property holds for all $t' \geq t$, which contradicts the construction of $\mathcal{A}_s^{\overset{\geq}{=}}$.

Thus, we assume from now on that Eq. (60) is positive.

Boundedness of the ratio and lower bound on $\pi_{t'}(\mathcal{A}_s^{>}|s)$ The positivity of Eq. (60) gives us an upper bound on the ratio: $\text{ratio}_t(\mathcal{A}_s^{>}, \mathcal{A}_s^{<} \cup \mathcal{A}_s^{\overset{\leq}{=}}) \leq \frac{v_\infty(s)-v_\perp}{\delta_\infty(s)-\xi}$, and equivalently the interesting inequality:

$$\pi_t(\mathcal{A}_s^{<} \cup \mathcal{A}_s^{\overset{\leq}{=}}|s)(v_\infty(s) - v_\perp) \geq \pi_t(\mathcal{A}_s^{>}|s)(\delta_\infty(s) - \xi) \tag{72}$$

$$\iff \quad \pi_t(\mathcal{A}_s^{<} \cup \mathcal{A}_s^{\overset{\leq}{=}}|s)(v_\infty(s) - v_\perp + \delta_\infty(s) - \xi) \geq (\pi_t(\mathcal{A}_s^{>}|s) + \pi_t(\mathcal{A}_s^{<} \cup \mathcal{A}_s^{\overset{\leq}{=}}|s))(\delta_\infty(s) - \xi) \tag{73}$$

$$\iff \quad \pi_t(\mathcal{A}_s^{<} \cup \mathcal{A}_s^{\overset{\leq}{=}}|s) \geq \frac{(1 - \pi_t(\mathcal{A}_s^{\overset{\geq}{=}}|s))(\delta_\infty(s) - \xi)}{v_\infty(s) - v_\perp + \delta_\infty(s) - \xi}. \tag{74}$$

Combining it with Eq. (71), we get for all $t' > t$:

$$\begin{aligned} \pi_{t'}(\mathcal{A}_s^{>}|s) &\geq \pi_{t'}(\mathcal{A}_s^{<} \cup \mathcal{A}_s^{\overset{\leq}{=}}|s)\text{ratio}_t(\mathcal{A}_s^{>}, \mathcal{A}_s^{<} \cup \mathcal{A}_s^{\overset{\leq}{=}}) && (75) \\ &\geq \underbrace{\frac{\text{ratio}_t(\mathcal{A}_s^{>}, \mathcal{A}_s^{<} \cup \mathcal{A}_s^{\overset{\leq}{=}})(\delta_\infty(s) - \xi)}{v_\infty(s) - v_\perp + \delta_\infty(s) - \xi}}_{\kappa>0} \left(1 - \pi_{t'}(\mathcal{A}_s^{\overset{\geq}{=}}|s)\right). && (76) \end{aligned}$$

The policy cannot converge to $\mathcal{A}_s^{\overset{\geq}{=}}$. Let us assume that the policy converges on $\mathcal{A}_s^{\overset{\geq}{=}}$:

$$\lim_{t\to\infty} \pi_t(\mathcal{A}_s^{\overset{\geq}{=}}|s) = 1. \tag{77}$$

From Eq. (55), we know that $(\theta_t)_{s,\mathcal{A}^{>}}$ increases with time (and we assumed that $\mathcal{A}_s^{>}$ is non-empty). This implies that $(\theta_t)_{s,\mathcal{A}^{\overset{\geq}{=}}} \to \infty$. We may therefore infer from Eq. (64) that:

$$\sum_{t=0}^{\infty} \eta_t d_t(s) \left(1 - \pi_{t'}(\mathcal{A}_s^{\overset{\geq}{=}}|s)\right) = \infty, \tag{78}$$

then:

$$\begin{aligned} (\theta_{t'})_{s,\mathcal{A}_s^{>}} &\geq (\theta_t)_{s,\mathcal{A}_s^{>}} + \sum_{t''=t}^{t'-1} \eta_{t''} d_{t''}(s) \pi_{t''}(\mathcal{A}_s^{>}|s) \left(\delta_\infty(s) - \xi\right) && (79) \\ &\geq (\theta_t)_{s,\mathcal{A}_s^{>}} + \kappa \left(\delta_\infty(s) - \xi\right) \sum_{t''=t}^{t'-1} \eta_{t''} d_{t''}(s) \left(1 - \pi_{t''}(\mathcal{A}_s^{\overset{\geq}{=}}|s)\right). && (80) \end{aligned}$$

Thus: $(\theta_{t'})_{s,\mathcal{A}_s^{>}} \to \infty$, which implies the divergence of $\text{ratio}_t(\mathcal{A}_s^{>}, \mathcal{A}_s^{<} \cup \mathcal{A}_s^{\overset{\leq}{=}})$, and a contradiction with the upper bound established earlier: we cannot have $\pi_t(\mathcal{A}_s^{\overset{\geq}{=}}|s) \to 1$.

Conclusion. We thus know that some policy mass must remain outside of $\mathcal{A}_s^{\overset{\geq}{=}}$. Because the ratio is bounded and increasing, some mass must be assigned to both $\mathcal{A}_s^{>}$ and $\mathcal{A}_s^{<} \cup \mathcal{A}_s^{-\infty}$. Equation 63 thus implies that $(\theta_t)_{s,\mathcal{A}_s^{>}}$ must diverge to ∞, and the ratio with it, leading to a final contradiction. We conclude that there does not exist a state where policy improvement is possible, in other words, the policy is optimal. □

Theorem 2c (**Necessity of C4 for optimality**). *Assumption C4 is necessary to guarantee that the sequence of value functions v_t converges to optimality.*

Proof. We consider a minimal MDP with one state: $\mathcal{S} \doteq \{s\}$ and two terminal actions: $\mathcal{A} \doteq \{a_1, a_2\}$. We assume that $\sum_t \eta_t d_t(s) = m < \infty$ and $r_1 = 1$. Now, we prove that we may choose $r_2 < 1$ such that the sequence (v_t) does not converge to optimality: $\lim_{t\to\infty} v_t < v_\star = 1$.

Direct parametrization: the update is:

$$\begin{aligned}
u_{s,a_1} &= d_t(s)q_\pi(s,a_1) && (81)\\
&= d_t(s) && (82)\\
u_{s,a_2} &= d_t(s)q_\pi(s,a_2) && (83)\\
&= d_t(s)r_2 && (84)\\
(\theta_{t+1})_{s,a_1} &= \min\left(1, (\theta_t)_{s,a_1} + \eta_t d_t(s)\frac{1-r_2}{2}\right) && (85)\\
(\theta_{t+1})_{s,a_2} &= \max\left(0, (\theta_t)_{s,a_2} + \eta_t d_t(s)\frac{r_2-1}{2}\right). && (86)
\end{aligned}$$

Starting from $(\theta_0)_{s,a_1}$ (resp. $\theta_0(a_2|s)$), $\theta_t(a_1|s)$ (resp. $\theta_t(a_2|s)$) hits 1 (resp. 0), if and only if:

$$\begin{aligned}
& \sum_{t=0}^{\infty}\eta_t d_t(s)\frac{1-r_2}{2} \geq 1-(\theta_0)_{s,a_1} && (87)\\
\iff \quad & m(1-r_2) \geq 2-2(\theta_0)_{s,a_1} && (88)\\
\iff \quad & r_2 \leq 1-\frac{2}{m}\left(1-(\theta_0)_{s,a_1}\right). && (89)
\end{aligned}$$

If we adversarially choose $r_2 = 1-\frac{1}{m}\left(1-(\theta_0)_{s,a_1}\right)$, then the optimality is never reached.

Softmax parametrization: the update is:

$$\begin{aligned}
u_{s,a_1,t} &= d_t(s)\pi_t(a_1|s)\text{adv}_t(s,a_1) && (90)\\
&= d_t(s)\pi_t(a_1|s)(1-\pi_t(a_1|s)-\pi_t(a_2|s)r_2) && (91)\\
&= d_t(s)\pi_t(a_1|s)(1-\pi_t(a_1|s))(1-r_2) && (92)\\
&\leq d_t(s)\frac{1-r_2}{4} && (93)\\
u_{s,a_2,t} &= d_t(s)\pi_t(a_2|s)\text{adv}_t(s,a_2) && (94)\\
&= d_t(s)\pi_t(a_2|s)(r_2-\pi_t(a_1|s)-\pi_t(a_2|s)r_2) && (95)\\
&= d_t(s)\pi_t(a_1|s)(1-\pi_t(a_1|s))(r_2-1) && (96)\\
&\geq d_t(s)\frac{r_2-1}{4}. && (97)
\end{aligned}$$

Therefore, we know that, starting from $(\theta_0)_{s,a_1}$, θ_t is upper bounded:

$$\begin{aligned}
(\theta_t)_{s,a_1} &\doteq (\theta_0)_{s,a_1} + \sum_{t'=0}^{t}\eta_{t'}u_{s,a_1,t} && (98)\\
&\leq (\theta_0)_{s,a_1} + \sum_{t'=0}^{t}\eta_{t'}d_t(s)\frac{1-r_2}{4} && (99)\\
&\leq (\theta_0)_{s,a_1} + m\frac{1-r_2}{4}. && (100)\\
& && (101)
\end{aligned}$$

Similarly, we know that, starting from $(\theta_0)_{s,a_2}$, θ_t is upper bounded:

$$\begin{aligned}
(\theta_t)_{s,a_2} &\doteq (\theta_0)_{s,a_2} + \sum_{t'=0}^{t}\eta_{t'}u_{s,a_2,t} && (102)\\
&\geq (\theta_0)_{s,a_2} + \sum_{t'=0}^{t}\eta_{t'}d_t(s)\frac{r_2-1}{4} && (103)\\
&\geq (\theta_0)_{s,a_2} - m\frac{1-r_2}{4}. && (104)\\
& && (105)
\end{aligned}$$

As a consequence, for any $0 \leq r_2 < 1$ and any t, we have:

$$
\begin{aligned}
v_\star(s) - v_t(s) &= 1 - (1 - \pi_t(a_2|s)(1 - r_2)) && (106)\\
&\geq \frac{\exp\left((\theta_0)_{s,a_2} - m\frac{1-r_2}{4}\right)}{\exp\left((\theta_0)_{s,a_1} + m\frac{1-r_2}{4}\right)}(1 - r_2) && (107)\\
&\geq \exp\left((\theta_0)_{s,a_2} - (\theta_0)_{s,a_1} - \frac{m}{2}\right)(1 - r_2), && (108)
\end{aligned}
$$

which is strictly positive and therefore concludes the proof. □

B.2.3 Convergence rates

Theorem 4 (**Asymptotic convergence rates under the softmax parametrization**). *With softmax parametrization, under A1, C4, and A8:*

A8. The optimal policy is unique: $\forall s,\ q_\star(s, a_1) = q_\star(s, a_2) = v_\star(s)$ *implies* $a_1 = a_2$*,*

the sequence of value functions q_t *converges asymptotically as follows:*

$$
\exists t_0,\ \textit{such that } \forall t \geq t_0, \quad v_\star(s) - v_t(s) \leq \frac{8|\mathcal{A}|(v_\top - v_\bot)}{(1-\gamma)\min_{s \in \textit{supp}(d_{\pi_\star,\gamma})} \delta(s) \sum_{t'=t_0}^{t-1} \eta_{t'} d_{t'}(s)}, \qquad (5)
$$

where $\delta(s) = v_\star(s) - \max_{a \in \mathcal{A}/\{\pi_\star(s)\}} q_\star(s, a)$ *is the gap with the best suboptimal action in state* s*,* $v_\top$ *(resp.* $v_\bot$*) is the maximal (resp. minimal) value, and* $\operatorname{supp}(d_{\pi_\star,\gamma})$ *denotes the support of the distribution of the optimal policy.*

Remark 1. *With A8, Thm. 4 provides bounds that improve the asymptotic rates that were stated before from* $\mathcal{O}(|\mathcal{A}|^3)$ *to* $\mathcal{O}(|\mathcal{A}|)$*).*

We note that we were not able to convince ourselves that [22] (the only existing convergence rate for the softmax policy) properly handles the case of multiple optimal policies either. We do believe that their proof works in the bandit setting, corresponding to their Section 3.2.1. However the general MDP setting studied in Section 3.2.2 is problematic. More specifically, their proof of Lemma 8 for multiple actions (only found in the Appendix) does not apply to the sum over all optimal actions, but to the sum over all greedy actions with respect to the current policy. This prevents a direct application of the rest of their results and raises the question of whether the rate does hold in that case.

We do not know whether the issue is structural, but we do believe that our proof technique, which ignores RL optimization properties, will not be able to prove a bound in $\mathcal{O}(1/t)$ *for problems with multiple optimal policies. Indeed, the problem arises when the optimization commits to one optimal action* $a_1^\top$ *at the expense of* $a_2^\top$ *in some state* s *(because e.g.* $q_t(s, a_2^\top) < v_t(s) < q_t(s, a_1^\top)$*), up to a point where* $\pi_t(s, a_1^\top) \gg \pi_t(s, a_2^\top)$*, and* $a_2^\top$ *later experiences a significant jump in value due to efficient optimization in its subsequent states, inducing* $q_t(s, a_1^\top) < v_t(s) < q_t(s, a_2^\top)$*. Then, the parameter update will be extremely slow: proportional to the product of* $\pi_t(s, a_2^\top)$ *and* $(1 - \pi_t(s, a_1^\top) + \pi_t(s, a_2^\top))$*, which are two very small quantities. The policy convergence to* $\mathcal{A}_s^\top$ *will be much slower, even go backwards for some time, since* $(\theta_t)_{s,a_1^\top}$ *will decrease and* $(\theta_t)_{s,a_2^\top}$ *will be too small to have a significant impact on* $\sum_{a^\top \in \mathcal{A}_s^\top} \pi_t(s, a^\top)$*.*

Proof sketch. We consider the partition between the optimal and the suboptimal actions in every state s:

$$
\mathcal{A}_s^\top = \{a \in \mathcal{A} | q_\star(s, a) = v_\star(s)\} \quad \text{and} \quad \mathcal{A}_s^\bot = \{a \in \mathcal{A} | q_\star(s, a) < v_\star(s)\}. \qquad (109)
$$

We define $\delta(s)$ as the gap with the best suboptimal action: $\delta(s) \doteq v_\star(s) - \max_{a \in \mathcal{A}_s^\bot} q_\star(s, a)$. Then, thanks to the uniform convergence and optimality proved in Corollary 1 and Thm. 2, we know that there exists t_0, such that $\forall t \geq t_0$, $\|q_\star - q_t\|_\infty \leq \frac{\delta(s)}{2}$. Then, we consider the sequence:

$$
(X_t)_{t \geq t_0} \doteq \begin{cases} X_{t_0} = \max_{a \in \mathcal{A}_s^\top} (\theta_{t_0})_{s,a} - \max_{a \in \mathcal{A}_s^\bot} (\theta_{t_0})_{s,a} \\ X_{t+1} = X_t + \frac{\delta(s)}{8} \eta_t d_t(s) e^{-X_t} \end{cases} \qquad (110)
$$

and prove that $\forall t \geq t_0$, $X_t \leq \max_{a\in\mathcal{A}_s^\top}(\theta_t)_{s,a} - \max_{a\in\mathcal{A}_s^\perp}(\theta_t)_{s,a}$. Further analysis allows us to prove the following rate in policy convergence to suboptimal actions:

$$\sum_{a\in\mathcal{A}_s^\perp} \pi_t(a|s) \leq \frac{8|\mathcal{A}|}{\delta(s)\sum_{t'=t_0}^{t-1}\eta_{t'}d_{t'}(s)}, \tag{111}$$

which allows us to prove the following upper bound on the value convergence rate:

$$v_\star(s) - v_t(s) \leq \frac{8|\mathcal{A}|(v_\top - v_\perp)}{(1-\gamma)\min_{s\in\text{supp}(d_{\pi_\star,\gamma})}\delta(s)\sum_{t'=t_0}^{t-1}\eta_{t'}d_{t'}(s)}. \qquad \square$$

Full proof. We consider the following partition of the actions in state s:

$$\mathcal{A}_s^\top = \{a \in \mathcal{A} | q_\star(s,a) = v_\star(s)\} \tag{112}$$
$$\mathcal{A}_s^\perp = \{a \in \mathcal{A} | q_\star(s,a) < v_\star(s)\}. \tag{113}$$

We define $\delta(s)$ as the gap with the best suboptimal action[7]:

$$\delta(s) \coloneqq v_\star(s) - \max_{a\in\mathcal{A}_s^\perp} q_\star(s,a). \tag{114}$$

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 $t \geq t_0$:

$$v_\star(s) - \xi \leq v_t(s) \leq v_\star(s) \tag{115}$$
$$q_\star(s,a) - \xi \leq q_t(s,a) \leq q_\star(s,a). \tag{116}$$

We fix $\xi \doteq \frac{\delta(s)}{2}$. We then write the advantage as:

$$\text{adv}_t(s,a) = \sum_{a'\neq a} \pi(a'|s)\left(q_t(s,a) - q_t(s,a')\right), \tag{117}$$

which gives the following bounds:

$$\begin{aligned}
\forall a \in \mathcal{A}_s^\top, \quad \text{adv}_t(s,a) &\geq (1 - \pi_t(\mathcal{A}_s^\top|s))\left(v_\star(s) - \xi - (v_\star(s) - \delta(s))\right) \\
&\quad + (\pi_t(\mathcal{A}_s^\top|s) - \pi_t(a|s))\left(v_\star(s) - \xi - v_\star(s)\right) && (118) \\
&= \frac{\delta(s)}{2}(1 - \pi_t(\mathcal{A}_s^\top|s)) - \frac{\delta(s)}{2}\left(\pi_t(\mathcal{A}_s^\top|s) - \pi_t(a|s)\right), && (119) \\
\forall a \in \mathcal{A}_s^\perp, \quad \text{adv}_t(s,a) &= q_t(s,a) - v_t(s) && (120) \\
&\leq q_\star(s,a) - (v_\star(s) - \xi) && (121) \\
&\leq -\frac{\delta(s)}{2}. && (122)
\end{aligned}$$

The last term in Eq. (119) may imply a negative advantage for actions in $\mathcal{A}_s^\top$. This term is null if $\mathcal{A}_s^\top$ contains only one element $a_s^\top$ and this is the assumption (A8) we need to make to progress further.

Therefore, we have for the parameters:

$$(\theta_{t+1})_{s,a_s^\top} \geq (\theta_t)_{s,a_s^\top} + \eta_t d_t(s)\pi_t(a_s^\top|s)\left(1 - \pi_t(a_s^\top|s)\right)\frac{\delta(s)}{2}, \tag{123}$$

$$\forall a \in \mathcal{A}/\{a_s^\top\}, \quad (\theta_{t+1})_{s,a} \leq (\theta_t)_{s,a} - \eta_t d_t(s)\pi_t(a|s)\frac{\delta(s)}{2} \leq (\theta_t)_{s,a}. \tag{124}$$

Let us consider the following sequence:

$$(X_t)_{t\geq t_0} \coloneqq \begin{cases} X_{t_0} = (\theta_{t_0})_{s,a_s^\top} - \max_{a\in\mathcal{A}/\{a_s^\top\}}(\theta_{t_0})_{s,a} \\ X_{t+1} = X_t + \frac{\delta(s)}{8}\eta_t d_t(s)e^{-X_t} \end{cases} \tag{125}$$

(126)

[7] If there is no suboptimal action, then $\sum_{a\in\mathcal{A}_s^\top}\pi(a|s) = 1$ and we proved what we wanted.

We prove by induction that for all $t \geq t_0$, $X_t \leq (\theta_t)_{s,a_s^\top} - \max_{a \in \mathcal{A}/\{a_s^\top\}} (\theta_t)_{s,a}$.

We have equality at initialization. Let us now assume the property true for t and prove it still holds for $t+1$. Let $a_t^\perp$ denote the action in $\mathcal{A}/\{a_s^\top\}$ for which the parameter is maximal at time t:

$$a_t^\perp \in \underset{a \in \mathcal{A}/\{a_s^\top\}}{\operatorname{argmax}} \, (\theta_t)_{s,a} = \underset{a \in \mathcal{A}/\{a_s^\top\}}{\operatorname{argmax}} \, \pi_t(a|s). \tag{127}$$

We recall that $t \geq t_0$ guarantees that $v_\star - v_t \leq \xi$, which implies that $\pi_t(a_s^\top|s) \geq \frac{1}{2}$. Moreover,

$$\begin{aligned} 1 - \pi_t(a_s^\top|s) &= \frac{\sum_{a \in \mathcal{A}/\{a_s^\top\}} e^{(\theta_t)_{s,a}}}{\sum_{a \in \mathcal{A}} e^{(\theta_t)_{s,a}}} && (128) \\ &\geq \frac{e^{(\theta_t)_{s,a_t^\perp}}}{2e^{(\theta_t)_{s,a_s^\top}}}. && (129) \end{aligned}$$

We now compute:

$$(\theta_{t+1})_{s,a_s^\top} - \max_{a \in \mathcal{A}/\{a_s^\top\}} (\theta_{t+1})_{s,a} \geq (\theta_t)_{s,a_s^\top} - (\theta_t)_{s,a_t^\perp} + \eta_t d_t(s) \pi_t(a_s^\top|s) \left(1 - \pi_t(a_s^\top|s)\right) \frac{\delta(s)}{2} \tag{130}$$

$$\begin{aligned} &\geq X_t + \eta_t d_t(s) \frac{1}{2} \frac{e^{(\theta_t)_{s,a_t^\perp}}}{2e^{(\theta_t)_{s,a_s^\top}}} \frac{\delta(s)}{2} && (131) \\ &\geq X_t + \frac{\delta(s)}{8} \eta_t d_t(s) e^{-(\theta_t)_{s,a_s^\top} + (\theta_t)_{s,a_t^\perp}} && (132) \\ &\geq X_t + \frac{\delta(s)}{8} \eta_t d_t(s) e^{-X_t} && (133) \\ &= X_{t+1} && (134) \end{aligned}$$

which concludes the induction.

Above, (130) is a direct application of (123) and (124), and (133) comes from the induction hypothesis and the fact that $x \mapsto x + \frac{\delta(s)}{8} \eta_t d_t(s) e^{-x}$ is increasing on $[\frac{\delta(s)}{8} \eta_t d_t(s), +\infty)$.

X_t belonging to that interval is easily guaranteed for t large enough as $X_t \to \infty$ and we assume $\eta_t d_t(s)$ to be bounded.

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):

$$\left\{ \begin{array}{l} f(t_0) = X_{t_0} \\ \frac{df(t)}{dt} = \frac{\delta(s)}{8} \eta_t d_t(s) e^{-f(t)}, \end{array} \right. \tag{135}$$

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 $\forall t \in \mathbb{N}, X_t \geq f(t)$. Additionally, we have:

$$f(t) = \log\left(\frac{\delta(s)}{8} \int_{t_0}^{t} \eta_{t'} d_{t'}(s) dt' + e^{X_{t_0}} \right). \tag{136}$$

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

$$X_t \geq \log\left(\frac{\delta(s)}{8} \int_{t_0}^{t} \eta_{t'} d_{t'}(s) dt' + e^{X_{t_0}} \right) = \log\left(\frac{\delta(s)}{8} \sum_{t'=t_0}^{t-1} \eta_{t'} d_{t'}(s) + e^{X_{t_0}} \right). \tag{137}$$

We can now write the following rate in policy convergence:

$$
\begin{aligned}
1-\pi_t(a_s^\top|s) &= \frac{\sum_{a\in\mathcal{A}/\{a_s^\top\}} e^{(\theta_t)_{s,a}}}{\sum_{a\in\mathcal{A}} e^{(\theta_t)_{s,a}}} && (138)\\
&\leq \frac{(|\mathcal{A}|-1)e^{(\theta_t)_{s,a_t^\perp}}}{e^{(\theta_t)_{s,a_s^\top}}} && (139)\\
&\leq (|\mathcal{A}|-1)e^{-X_t} && (140)\\
&\leq |\mathcal{A}|\frac{1}{\frac{\delta(s)}{8}\sum_{t'=t_0}^{t-1}\eta_{t'}d_{t'}(s)+e^{X_t}} && (141)\\
&\leq \frac{8|\mathcal{A}|}{\delta(s)\sum_{t'=t_0}^{t-1}\eta_{t'}d_{t'}(s)}. && (142)
\end{aligned}
$$

On the value side, we further get:

$$
\begin{aligned}
v_\star(s)-v_t(s) &= \mathbb{P}\left[A=a_s^\top|A\sim\pi_t(\cdot|s)\right]\times\gamma\mathbb{E}\left[v_\star(S')-v_t(S')|S'\sim p(\cdot|s,a_s^\top)\right] && (143)\\
&\quad+\mathbb{P}\left[A\in\mathcal{A}/\{a_s^\top\}|A\sim\pi_t(\cdot|s)\right]\left(v_\star(s)-\mathbb{E}\left[q_t(s,A)|A\sim\pi_t(\cdot|s)\cap\mathcal{A}/\{a_s^\top\}\right]\right)\\
&\leq \gamma\mathbb{E}\left[v_\star(S')-v_t(S')|S'\sim p(\cdot|s,a_s^\top)\right]+\frac{8|\mathcal{A}|(v_\star(s)-\min_{a\in\mathcal{A}}q_t(s,a))}{\delta(s)\sum_{t'=t_0}^{t-1}\eta_{t'}d_{t'}(s)} && (144)\\
&\leq \frac{8|\mathcal{A}|(v_\top-v_\perp)}{(1-\gamma)\min_{s\in\text{supp}(d_{\pi_\star,\gamma})}\delta(s)\sum_{t'=t_0}^{t-1}\eta_{t'}d_{t'}(s)}, && (145)
\end{aligned}
$$

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. □

Theorem 3 (Asymptotic convergence rates under the direct parametrization). *With direct parametrization, under A1 and C4, the sequence of value functions q_t converges to optimality in finite time:*

$$\exists t_0, \text{ such that } \forall t\geq t_0, \quad q_t=q_\star. \tag{4}$$

Proof. This is a direct consequence of Lemma 1 and Eq. (39) in Thm. 2a, which shows that the probability of choosing an optimal action grows by updates that sum to infinity until it hits its ceiling probability of 1. Therefore, this ceiling will be hit in a finite time t_1 in all states. □

Lemma 1. *Let $x\in\Delta_n$, the simplex of $\mathbb{R}^n$, and $y\in\mathbb{R}^n$. Assume that there exists $0<k<n$ and $\alpha>\beta$ such that $\forall i\leq k, y_i\geq\alpha$ and $\forall i>k, y_i\leq\beta$, then*

$$\sum_{i\leq k}\mathcal{P}_{\Delta_n}(x+y)_i\geq\min\left(1,\sum_{i\leq k}x_i+\frac{\alpha-\beta}{2}\right). \tag{146}$$

Proof. Let us assume without loss of generality that $\forall i, y_i\geq 0$ (by e.g. subtracting from each of them $\min y_i$). Let us also define the following sets:

$$
\begin{aligned}
\mathcal{A}^\top &= \{i\leq k\}, && (147)\\
\mathcal{A}^{\perp,+} &= \{i>k\,|\,\mathcal{P}_{\Delta_n}(x+y)_i>0\}, && (148)\\
\mathcal{A}^{\perp,-} &= \{i>k\,|\,\mathcal{P}_{\Delta_n}(x+y)_i=0\}. && (149)
\end{aligned}
$$

We assume that $\mathcal{A}^{\perp,+}\neq\emptyset$, as otherwise we would have $\sum_{i\leq k}\mathcal{P}_{\Delta_n}(x+y)_i=1$, and (146) would hold.

We are interested in the effect of $\mathcal{P}_{\Delta_n}$ on the mass of the coordinates of $x+y$ that belong to $\mathcal{A}^\top$. Given the assumption that y is coordinate-wise non-negative, the Euclidian projection onto Δ_n will remove from the coordinates of $x+y$ the mass $\sum_i y_i$ that x gained by adding y to it. That mass will be removed as uniformly as possible (corresponding to the equality case in Cauchy-Schwarz inequality), while ensuring that each coordinate remains positive. In our case, this means that $\mathcal{P}_{\Delta_n}$ will remove all the mass of the coordinates in $\mathcal{A}^{\perp,-}$, and then remove the remaining added mass uniformly from the other coordinates[8]:

- The mass added to the first k coordinates is $\sum_{i \in \mathcal{A}^\top} y_i$.
- The total mass to remove is $K = \sum_i y_i$.
- The mass removed from $\mathcal{A}^{\perp,-}$ is $K_{\mathcal{A}^{\perp,-}} \doteq \sum_{i \in \mathcal{A}^{\perp,-}} x_i + y_i$, which is larger than $\sum_{i \in \mathcal{A}^{\perp,-}} y_i$.
- The remaining mass $K_{\mathcal{A}^\top} + K_{\mathcal{A}^{\perp,+}} = K - K_{\mathcal{A}^{\perp,-}}$ is smaller than $\sum_{i \in \mathcal{A}^\top \cup \mathcal{A}^{\perp,+}} y_i$ and removed uniformly from the coordinates in $\mathcal{A}^\top \cup \mathcal{A}^{\perp,+}$.

Formally, this gives us:

$$\begin{aligned}
\sum_{i \in \mathcal{A}^\top} \mathcal{P}_{\Delta_n}(x+y)_i - \sum_{i \in \mathcal{A}^\top} x_i &= \sum_{i \in \mathcal{A}^\top} y_i - \frac{|\mathcal{A}^\top|}{|\mathcal{A}^\top| + |\mathcal{A}^{\perp,+}|} \left(K_{\mathcal{A}^\top} + K_{\mathcal{A}^{\perp,+}} \right) && (150) \\
&\geq \sum_{i \in \mathcal{A}^\top} y_i - \frac{|\mathcal{A}^\top|}{|\mathcal{A}^\top| + |\mathcal{A}^{\perp,+}|} \sum_{i \in \mathcal{A}^\top \cup \mathcal{A}^{\perp,+}} y_i && (151) \\
&= \frac{|\mathcal{A}^{\perp,+}|}{|\mathcal{A}^\top| + |\mathcal{A}^{\perp,+}|} \sum_{i \in \mathcal{A}^\top} y_i - \frac{|\mathcal{A}^\top|}{|\mathcal{A}^\top| + |\mathcal{A}^{\perp,+}|} \sum_{i \in \mathcal{A}^{\perp,+}} y_i && (152) \\
&\geq \frac{|\mathcal{A}^{\perp,+}||\mathcal{A}^\top|}{|\mathcal{A}^\top| + |\mathcal{A}^{\perp,+}|} \alpha - \frac{|\mathcal{A}^\top||\mathcal{A}^{\perp,+}|}{|\mathcal{A}^\top| + |\mathcal{A}^{\perp,+}|} \beta && (153) \\
&= \frac{|\mathcal{A}^{\perp,+}||\mathcal{A}^\top|}{|\mathcal{A}^\top| + |\mathcal{A}^{\perp,+}|} (\alpha - \beta). && (154)
\end{aligned}$$

$\frac{|\mathcal{A}^{\perp,+}||\mathcal{A}^\top|}{|\mathcal{A}^\top| + |\mathcal{A}^{\perp,+}|}$ is an increasing function of $|\mathcal{A}^{\perp,+}|$ and $|\mathcal{A}^\top|$ and both those quantities are larger than 1. Hence:

$$\sum_{i \in \mathcal{A}^\top} \mathcal{P}_{\Delta_n}(x+y)_i - x_i \geq \frac{\alpha - \beta}{2}, \tag{155}$$

which concludes the proof. □

B.3 Non-asymptotic convergence rates under A2

We now define an extra assumption that allows us to get non-asymptotic convergence rates under A2.

C7. There exists $d_\top \geq d_\perp > 0$ such that $\forall s, t, d_\top \geq d_t(s) \geq d_\perp$.

We will be needing $\left\| \frac{1}{p_0} \right\|_\infty$ and note that under A2, it is a well-defined quantity.

Finally, we let $M_t \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}||\mathcal{A}|}$ be the diagonal matrix defined by $M_t(s,a;s,a) = \frac{d_t(s)}{d_{\pi_t,\gamma}(s)}$ and 0 elsewhere. We note that:

$$U(\theta_t, d_t) = M_t \nabla_{\theta|\theta_t} \mathcal{J}(\pi_t). \tag{156}$$

We also denote by $\lambda_t = \min_s \frac{d_t(s)}{d_{\pi_t,\gamma}(s)}$ and $\Lambda_t = \max_s \frac{d_t(s)}{d_{\pi_t,\gamma}(s)}$ the smallest and largest eigenvalues of M_t, and by $\kappa_t = \frac{\Lambda_t}{\lambda_t}$ its condition number. We have the following inequalities:

$$\lambda_t \geq (1-\gamma) d_\perp, \qquad \Lambda_t \leq d_\top \left\| \frac{1}{p_0} \right\|_\infty, \qquad \kappa_t \leq \left\| \frac{1}{p_0} \right\|_\infty \frac{d_\top}{(1-\gamma) d_\perp}. \tag{157}$$

[8] We note that this can potentially result in some coordinates in $\mathcal{A}^\top$ becoming negative. As we are looking for an upper bound on the mass removed from $\mathcal{A}^\top$, this is not problematic (the mass removed in excess from $\mathcal{A}^\top$ will then be added back to it and removed from coordinates both in and out of $\mathcal{A}^\top$, resulting in a smaller decrease).

Theorem 7 (Convergence rate for the softmax parametrization under A2). *With a softmax parametrization, under A2, C7 and A8, and with* $\eta = \frac{(1-\gamma)^3}{8}\frac{1}{d_\top}\left\|\frac{1}{p_0}\right\|_\infty^{-1}$*, we have the following convergence rate for any* s*:*

$$v_\infty(s) - v_t(s) \le \frac{16|\mathcal{S}|C}{(1-\gamma)^7 t}\cdot\left\|\frac{d_{\pi_\star,\gamma}}{p_0}\right\|_\infty^2\left\|\frac{1}{p_0}\right\|_\infty^2\frac{d_\top}{d_\perp}, \tag{158}$$

where $C = \frac{1}{\inf_{s\in\mathcal{S},t\ge 1}\pi_t(\pi_\star(s)|s)^2}$ *and* $\pi_\star$ *is the unique (per A8) optimal policy .*

Proof. We note that this result is a generalization of the bound from [22] to our more general update rule. The extra $\left\|\frac{1}{p_0}\right\|_\infty\frac{d_\top}{(1-\gamma)d_\perp}$ term, coming from the upper bound on κ_t, is the price we pay for the generalization.

Our proof follows the logic from [22] and relies on the fact that M_t is diagonal and positive-definite. Hence it performs some form of conditioning on the gradient, which will affect the convergence rate, but does not change the overall evolution of the policies. More formally, we start by using the strong convexity of $\mathcal{J}$ to get:

$$|\mathcal{J}(\pi_{t+1}) - \mathcal{J}(\pi_t) - \langle\nabla_\theta\mathcal{J}(\pi_t), \theta_{t+1}-\theta_t\rangle| \le \frac{4}{(1-\gamma)^3}\|\theta_{t+1}-\theta_t\|^2. \tag{159}$$

Now, reusing notations from [22], we let $\delta_t = \mathcal{J}(\pi_*) - \mathcal{J}(\pi_t)$. Let us also remember that $U(\theta_t, d_t) = M_t\nabla_\theta\mathcal{J}(\pi_t)$, and that λ_t and Λ_t are respectively the smallest and largest eigenvalues of M_t. Then

$$\begin{aligned}
\delta_{t+1} - \delta_t &= \mathcal{J}(\pi_t) - \mathcal{J}(\pi_{t+1}) && (160)\\
&\le -\langle\nabla_\theta\mathcal{J}(\pi_t), \theta_{t+1}-\theta_t\rangle + \frac{4}{(1-\gamma)^3}\|\theta_{t+1}-\theta_t\|^2 && (161)\\
&= -\eta_t\langle\nabla_\theta\mathcal{J}(\pi_t), M_t\nabla_\theta\mathcal{J}(\pi_t)\rangle + \frac{4\eta_t^2}{(1-\gamma)^3}\|M_t\nabla_\theta\mathcal{J}(\pi_t)\|^2 && (162)\\
&= \sum_s\frac{d_t(s)}{d_{\pi_t,\gamma}(s)}\left(-\eta_t + \frac{4\eta_t^2}{(1-\gamma)^3}\frac{d_t(s)}{d_{\pi_t,\gamma}(s)}\right)\langle\nabla_{\theta_s}\mathcal{J}(\pi_t), \nabla_{\theta_s}\mathcal{J}(\pi_t)\rangle && (163)\\
&\le \sum_s\frac{d_t(s)}{d_{\pi_t,\gamma}(s)}\left(-\eta_t + \frac{4\eta_t^2}{(1-\gamma)^3}\Lambda_t\right)\langle\nabla_{\theta_s}\mathcal{J}(\pi_t), \nabla_{\theta_s}\mathcal{J}(\pi_t)\rangle && (164)\\
&\le -\sum_s\frac{d_t(s)}{d_{\pi_t,\gamma}(s)}\frac{(1-\gamma)^3}{16}\frac{1}{\Lambda_t}\langle\nabla_{\theta_s}\mathcal{J}(\pi_t), \nabla_{\theta_s}\mathcal{J}(\pi_t)\rangle \qquad (\eta_t = \frac{(1-\gamma)^3}{8}\frac{1}{\Lambda_t}) && (165)\\
&\le -\sum_s\frac{(1-\gamma)^3}{16}\frac{\lambda_t}{\Lambda_t}\langle\nabla_{\theta_s}\mathcal{J}(\pi_t), \nabla_{\theta_s}\mathcal{J}(\pi_t)\rangle && (166)\\
&= -\frac{(1-\gamma)^3}{16}\frac{\lambda_t}{\Lambda_t}\|\nabla_\theta\mathcal{J}(\pi_t)\|^2. && (167)
\end{aligned}$$

We now apply Lemma 8 from [22], which holds under A2 and A8, to get

$$\begin{aligned}
\delta_{t+1} - \delta_t &\le -\frac{(1-\gamma)^3}{16|\mathcal{S}|}\frac{1}{\kappa_t}\left\|\frac{d_{\pi_\star,\gamma}}{d_{\pi_t,\gamma}}\right\|_\infty^{-2}\cdot\left[\min_s\pi_t(a_\star(s)|s)\right]^2\cdot|\mathcal{J}(\pi_*) - \mathcal{J}(\pi_t)|^2 && (168)\\
&\le -\frac{(1-\gamma)^3}{16|\mathcal{S}|}\frac{1}{\kappa_t}\left\|\frac{d_{\pi_\star,\gamma}}{p_0}\right\|_\infty^{-2}\cdot\left[\inf_{t,s}\pi_t(a_\star(s)|s)\right]^2\cdot\delta_t^2, && (169)
\end{aligned}$$

where $a_\star(s)$ is the unique optimal action in s (per A8). We know from Thm. 2b that for all s, $\pi_t(a_\star(s)|s)\to 1$. Thus $\inf_{t,s}\pi_t(a_\star(s)|s) > 0$ and we can define $1/C \doteq \left[\inf_{t,s}\pi_t(a_\star(s)|s)\right]^2 > 0$.

Let us write $K_t \doteq \frac{(1-\gamma)^3}{16|\mathcal{S}|C}\frac{1}{\kappa_t}\left\|\frac{d_{\pi_\star,\gamma}}{p_0}\right\|_\infty^{-2}$. We have $\delta_{t+1} - \delta_t \leq -K_t\delta_t^2$. Exactly as in [22], we now proceed by induction to show that $\delta_t \leq \frac{1}{\min_{i\leq t} K_i t}$.

For $t = 2$, we clearly have $\delta_1 \leq \frac{1}{1-\gamma} \leq \frac{1}{2K_1}$. Now, $\delta_{t+1} \leq \delta_t - \min_{i\leq t} K_i\delta_t^2$. On $[0, \frac{1}{2\min_{i\leq t} K_i}]$, the function $x - \min_{i\leq t} K_i x^2$ is increasing, hence:

$$\delta_{t+1} \leq \frac{1}{\min_{i\leq t} K_i t} - \frac{\min_{i\leq t} K_i}{\min_{i\leq t} K_i^2 t^2} \leq \frac{1}{\min_{i\leq t} K_i(t+1)} \leq \frac{1}{\min_{i\leq t+1} K_i(t+1)}, \tag{170}$$

which concludes the induction.

In the end, we get:

$$\begin{aligned}\mathcal{J}(\pi_*) - \mathcal{J}(\pi_t) &\leq \frac{16|\mathcal{S}|C}{(1-\gamma)^3}\max_{i\leq t}\kappa_i\left\|\frac{d_{\pi_\star,\gamma}}{p_0}\right\|_\infty^2\frac{1}{t} &(171)\\ &\leq \frac{16|\mathcal{S}|C}{(1-\gamma)^4}\left\|\frac{1}{p_0}\right\|_\infty\frac{d_\top}{d_\perp}\left\|\frac{d_{\pi_\star,\gamma}}{p_0}\right\|_\infty^2\frac{1}{t}, &(172)\end{aligned}$$

where in the last line we used the bound on κ from Eq (157). Note that to match the definitions in the bound from [22], we can write is as:

$$\mathcal{J}(\pi_*) - \mathcal{J}(\pi_t) \leq \frac{16|\mathcal{S}|C}{(1-\gamma)^6 t}\left\|\frac{(1-\gamma)d_{\pi_\star,\gamma}}{p_0}\right\|_\infty^2\left\|\frac{1}{p_0}\right\|_\infty\frac{d_\top}{d_\perp}.$$

Finally, we move from $\mathcal{J}(\pi_*)$ and $\mathcal{J}(\pi_t)$ to $v_\infty(s)$ and $v_t(s)$ which introduces an extra $\frac{1}{1-\gamma}\left\|\frac{1}{p_0}\right\|_\infty$ and concludes the proof. □

We can use the above result to compute the time at which the rates from Thm. 4 become valid.

Corollary 2. *t_0 from Thm. 4 can be chosen as follows:*

$$t_0 = \frac{32|\mathcal{S}|C}{(1-\gamma)^7}\cdot\left\|\frac{d_{\pi_\star,\gamma}}{p_0}\right\|_\infty^2\cdot\left\|\frac{1}{p_0}\right\|_\infty^2\frac{d_\top}{d_\perp}\frac{1}{\inf_s\delta(s)}. \tag{173}$$

Proof. We know that the rates from Thm. 4 kick in for t_0 such that for any s, a and $t \geq t_0$:

$$\begin{aligned}v_\infty(s) - \frac{\delta(s)}{2} &\leq v_t(s) \leq v_\infty(s),\\ q_\infty(s,a) - \frac{\delta(s)}{2} &\leq q_t(s,a) \leq q_\infty(s,a).\end{aligned}$$

From Thm. 7, it is straightforward to show that it is verified for:

$$t_0 \geq \frac{32|\mathcal{S}|C}{(1-\gamma)^7 t}\cdot\left\|\frac{d_{\pi_\star,\gamma}}{p_0}\right\|_\infty^2\left\|\frac{1}{p_0}\right\|_\infty^2\frac{d_\top}{d_\perp}\frac{1}{\inf_s\delta(s)}.$$

□

C Domains

C.1 Chain domain

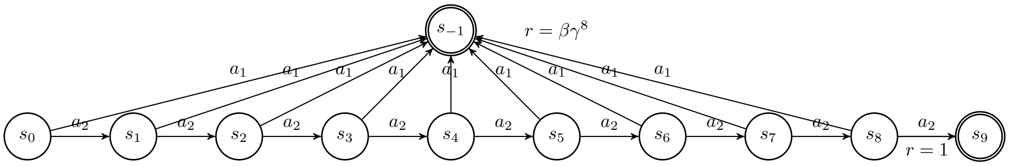


Figure 4: Deterministic chain MDP. Initial state is s_0. Reward is 0 everywhere except when accessing final states s_{-1} and s_9. Reward in s_{-1} is set such that $q(s_0, a_1) = \beta q_\star(s_0, a_2)$, with $\beta \in [0, 1)$.

The chain domain is designed to measure the ability of algorithms to overcome misleading rewards, *i.e.* immediate rewards that would guide the updates towards a suboptimal solution. In every state s_k, the agent has the opportunity to play action a_1 and receive an immediate reward of $\beta\gamma^{|\mathcal{S}|-2}$, or to play a_2 and progress to next state s_{k+1} without any immediate reward. A reward of 1 is eventually obtained when reaching state $s_{|\mathcal{S}|-1}$. Fig. 4 represents a chain of size 10 (both terminal states s_{-1} and $s_{\mathcal{S}-1}$ count as a single state). Formally, the deterministic MDP $\langle \mathcal{S}, \mathcal{A}, p, p_0, r, \gamma \rangle$ is constructed from two hyperparameters: the size $|\mathcal{S}|$ and the ratio β:

$$\mathcal{S} \doteq \{s_i\}_{i \in [\![0, |\mathcal{S}|-1]\!]} \tag{174}$$

$$\mathcal{A} \doteq \{a_1, a_2\} \tag{175}$$

$$p(\cdot|s,a) \doteq \begin{cases} \forall s,\ p(s_{|\mathcal{S}|-1}|s, a_1) = 1 \\ \forall i < |\mathcal{S}| - 1,\ p(s_{i+1}|s_i, a_2) = 1 \\ \text{the episode terminates when } s_{|\mathcal{S}|-1} \text{ is reached} \end{cases} \tag{176}$$

$$p_0(s_0) \doteq 1 \tag{177}$$

$$r(s,a) \doteq \begin{cases} \forall s,\ p(s, a_1) = \beta\gamma^{|\mathcal{S}|-2} \\ \forall i < |\mathcal{S}| - 2,\ r(s_i, a_2) = 0 \\ r(s_{|\mathcal{S}|-1}, a_2) = 1 \end{cases} \tag{178}$$

$$\gamma \doteq 0.99. \tag{179}$$

There are two remarkable policies:

- Low-hanging-fruit policy: $\begin{cases} \pi_\perp(a_1|s) = 1 \\ v_\perp(s) = \beta\gamma^{|\mathcal{S}|-2} \end{cases}$
- Optimal policy: $\begin{cases} \pi_\star(a_2|s) = 1 \\ v_\star(s_i) = \gamma^{|\mathcal{S}|-2-i} \end{cases}$

The policy gradient is attracted by the low-hanging-fruit policy because of its shorter horizon, even though it is sub-optimal.

In order to account for different experimental settings where the size of the chain $|\mathcal{S}|$ or the value ratio β differ, we report the experimental results in terms of normalized expected return, formally computed as:

$$\overline{\mathcal{J}}_\pi = \frac{\mathcal{J}_\pi - \mathcal{J}_\perp}{\mathcal{J}_\star - \mathcal{J}_\perp}, \tag{180}$$

where $\mathcal{J}_\star = v_\star(s_0)$ is the performance of the optimal policy and $\mathcal{J}_\perp = v_\perp(s_0)$ the performance of the low-hanging-fruit policy. Note in particular that $\overline{\mathcal{J}}_\star = 1$ and $\overline{\mathcal{J}}_\perp = 0$.

C.2 Random MDPs domain

The Random MDPs domain is designed to test the algorithms in situations where the exploration is not an issue. Indeed, by its design of highly stochastic transition functions, Random MDPs will have a non-null chance to visit every state whatever the behavioural policy is. This is therefore a domain where we would expect the policy gradient update to perform well, if not the best, and hope our modified updates still perform comparably. We reproduce the Random MDPs environment published in Section B.1.3 of [18] (also in [25, 40]), for which we recall the specifications hereafter.

The Random MDPs domain uses four parameters for the MDP generation: the number of states, the number of actions in each state, the connectivity of the transition function stating how many states are reachable after performing a given action in a given state, and the discount factor γ. In our experiments, we fix the number of actions to 4, the connectivity to 2, and the discount factor γ to 0.99.

The initial state is arbitrarily set to be s_0, we then search with dynamic programming the performance of the optimal policy for all potential terminal state $s_f \in \mathcal{S}/s_0$. We select the terminal state for which the optimal policy yields the smaller value function and set it as terminal: $r(s, a, s_f) = 1$ and $P(s|s_f, a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. The reward function is set to 0 everywhere else. We found that the optimal value-function is on average 0.6, amounting to an average horizon of 10, and has surprising low variance. Below, we write this environmental MDP $\langle \mathcal{S}, \mathcal{A}, p, p_0, r, \gamma \rangle$, its optimal state-action value function $q_\star$, its optimal performance $\mathcal{J}_\star$, and its uniform policy performance $\mathcal{J}_u$, where $\widetilde{\pi}$ denotes the uniform random policy: $\pi_u(a|s) = \frac{1}{|\mathcal{A}|}$ for all $s \in \mathcal{S}$ and all $a \in \mathcal{A}$.

In order to account for different random MDPs, we report the experimental results in terms of normalized expected return, formally computed as:

$$\overline{\mathcal{J}}_\pi = \frac{\mathcal{J}_\pi - \mathcal{J}_u}{\mathcal{J}_\star - \mathcal{J}_u}. \tag{181}$$

Note in particular that $\overline{\mathcal{J}}_\star = 1$ and $\overline{\mathcal{J}}_u = 0$.

D Full report of finite MDPs planning experiments (under A1)

D.1 Chain domain

In classic algorithms, the actor is conflicted between two objectives: explore and visit the unknown states, or exploit and follow the best policy under the current incomplete knowledge. The chain experiment depicted on Fig. 4 is intended to test actor-critic algorithms' ability to explore and find the optimal policy under misleading rewards, and then to commit to this optimal policy. In every state, the agent has the opportunity to choose action a_1 and receive a fair reward, or to go further without any reward signal and get a chance to eventually yield a higher return.

Our analysis underlines the importance of exploration and states the requirements for guaranteeing sufficient exploration. Our experiment validates these findings. We consider:

- the on-policy updates:
  - policy gradient (PG) distribution: $\frac{d_{\pi_t,\gamma}}{\|d_{\pi_t,\gamma}\|_1}$,
  - undiscounted distribution: $\frac{d_{\pi_t,1}}{\|d_{\pi_t,1}\|_1}$,
- uniform distribution: $d_u = \frac{1}{|\mathcal{S}|}$,
- and off-policy trade-offs between the uniform and policy gradient distributions: $o_t d_u + (1 - o_t)\frac{d_{\pi_t,\gamma}}{\|d_{\pi_t,\gamma}\|_1}$
  - with o_t constant: 0.1 and 0.48,
  - with $o_t \in \Theta(t^{-0.48})$: $o_t = 10t^{-0.48}$,
  - with $o_t \in \Theta(t^{-1})$: $o_t = 10t^{-1}$,

For a fair comparison, all update densities d_t are normalized: $\sum_s d_t(s) = 1$.

We consider four experiments:

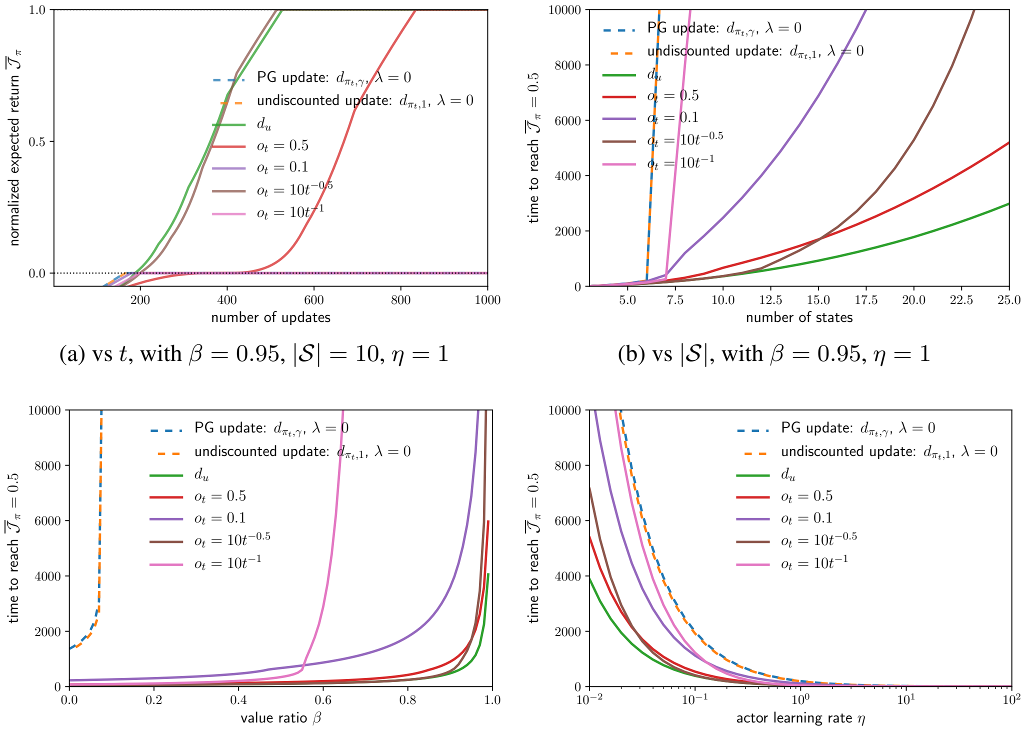
- **Exp1:** We observe instantaneous performance of each density schedule across iterations.
- We look at the speed of discovering the optimal policy. We measure it by tracking the number of updates required to reach a normalised return of 0.48 and its dependency on:
  - **Exp2:** the number of states in the chain,
  - **Exp3:** the value rate β between the low-hanging-fruit policy and the optimal one,
  - **Exp4:** the learning rate.

For the softmax parametrization, we also add a policy entropy regularizing term $-\lambda \log \pi(\cdot|s)$ and observe its impact in **Exp5** (in replacement to **Exp4** which delivers the same results as in the direct parametrization), as is often done in practice.

D.1.1 Direct parametrization

Exp1: Its results are displayed on Fig. 5a. We observe that on-policy updates converge very fast to the low-hanging fruit policy. Fast convergence can be desirable in easy-planning tasks, but in harder tasks such as the chain it leads to suboptimal behavior. In contrast, relying on a uniform state distribution for the actor updates is efficient and eventually discovers the optimal policy as long as the off-policiness o_t is strong enough. It is worth noting that even under the guarantees offered by having C4 satisfied (when $o_t \in \Omega(t^{-1})$, the policy can get stuck on the low-hanging-fruit policy for a long time (as for $o_t = 10t^{-1}$). However, once the optimal policy has been identified the convergence is fast and meets the optimal performance predicted by Thm. 3.

Exp2: Its results are displayed on Fig. 5b. We observe a failure mode of on-policy updates. When the chain is of size 7 or more, the policy converges to the suboptimal solution and will never get out of it. In contrast, C4 guarantees the policy will eventually be optimal. However, as illustrated by the $o_t = 10t^{-1}$ curve, it might be very long (possibly exponentially long) if the condition is only satisfied tightly. The convergence gets faster as the weight granted to the uniform density increases.



(a) vs t, with $\beta = 0.95$, $|\mathcal{S}| = 10$, $\eta = 1$

(b) vs $|\mathcal{S}|$, with $\beta = 0.95$, $\eta = 1$

(c) vs β, with $|\mathcal{S}| = 15$, $\eta = 1$

(d) vs η, with $\beta = 0.48$, $|\mathcal{S}| = 10$

Figure 5: Chain experiments with the direct parametrization under A1.

Exp3: Its results are displayed on Fig. 5c. We observe that the issues spotted in **Exp1** and **Exp2** is not due to the near-optimality of the low-hanging-fruit policy. Indeed, even with a reasonably short chain (15), and the low-hanging-fruit policy yielding an expected return 10 times smaller than optimal, the on-policy updates will still converge to it. In contrast, when the density requirement is satisfied, the updates reliably discover the optimal solution.

Exp4: Its results are displayed on Fig. 5d. We observe that the actor learning rate does not have any impact on the ability of the update to lead to an optimal policy[9]. Sill, making it larger speeds up convergence, so in the idealized setting we considered in this section where A1 is satisfied, the higher the actor learning rate, the better. And it has to be noted that setting it to infinity makes the update identical to a policy improvement step in the policy iteration algorithm.

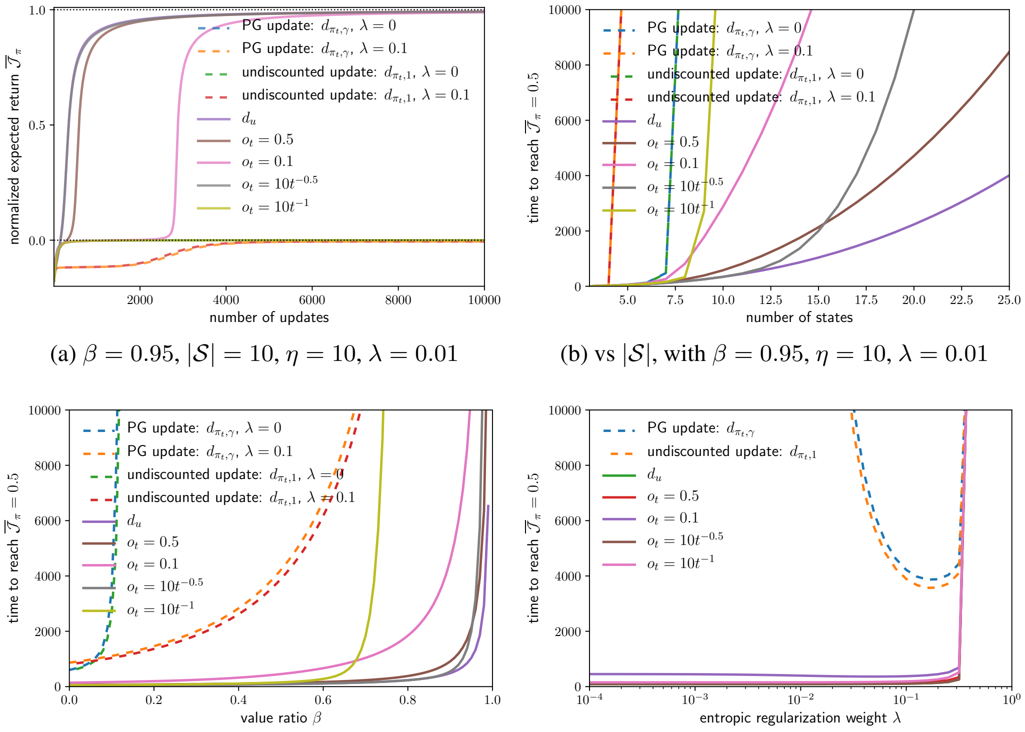
D.1.2 Softmax parametrization

Note that η has been multiplied by 10 in the softmax experiments as compared to the direct ones, in order to get a rate of convergence comparable to that of direct parametrization.

Exp1: Its results are displayed on Fig. 6a. We observe similar results than for the direct parametrization, except that the algorithms never completely converge to the optimal (or suboptimal) policies. We also observe an interesting behaviour of the policy entropy regularized versions of the on-policy updates, at first converging to a policy worse than the low-hanging-fruit one. This stems from the bias induced by the policy entropy regularization: during a first stage it converged to an entropy regularized policy close to the low-hanging-fruit policy, then, after update 2000, it discovered the low-hanging-fruit policy and slowly converged to it.

Exp2: Its results are displayed on Fig. 6b. We observe exactly the same results as in the direct parametrization.

[9] We do not show the results, but on-policy updates never find the optimal policy when $\beta = 0.7$ for any η.



(a) $\beta = 0.95$, $|\mathcal{S}| = 10$, $\eta = 10$, $\lambda = 0.01$

(b) vs $|\mathcal{S}|$, with $\beta = 0.95$, $\eta = 10$, $\lambda = 0.01$

(c) vs β, with $|\mathcal{S}| = 15$, $\eta = 10$, $\lambda = 0.01$

(d) vs λ, with $\beta = 0.48$, $|\mathcal{S}| = 15$, $\eta = 10$

Figure 6: Chain experiments with the softmax parametrization under A1.

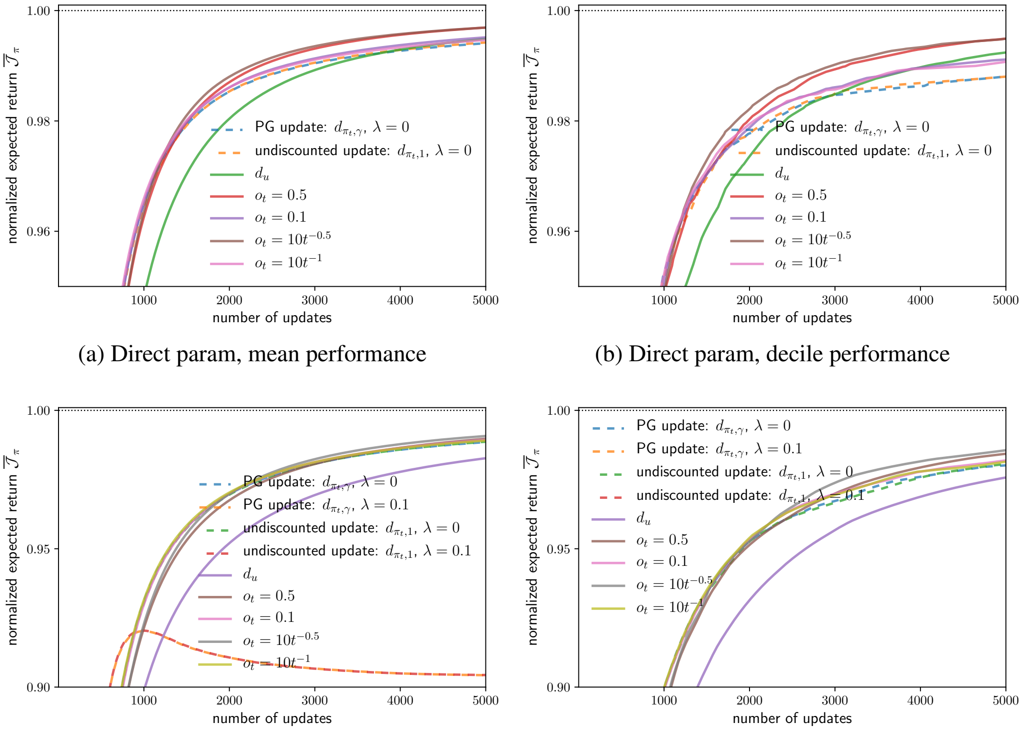
Exp3: Its results are displayed on Fig. 6c. We get results very similar to those of the direct parametrization. The policy entropy regularized updates behaviour is however quite interesting. It seems quite efficient at helping the on-policy updates find the best policy, but the introduced bias prevents learning it when β is too high. This last remark gets very clear by looking at Fig. 6b where policy entropy regularized updates perform terribly because β is high, even worse than the on-policy updates without the policy entropy regularization. It is also worth noting that this is the first experiment where there is a perceptible difference between on-policy updates, with the undiscounted update getting a little edge ($\gamma = 0.99$ is rather high).

Exp5: Its results are displayed on Fig. 6d. The first thing to notice is that, for all updates, when the policy entropy regularization gets large, $\lambda \approx 0.25$, the introduced bias is so high that the optimal regularized policy has a normalized performance lower than 0.48 (similar to what we observed on Fig. 6a). We also notice, that, for $\lambda < 0.25$, its impact on updates that satisfy C4 is near to null. Finally, for the on-policy updates, there is a sweet spot where it allows them to converge in the vicinity of the optimal policy reasonably fast, but it is rather narrow in terms of λ values, and slow in terms of time to find the optimal policy. And finally, let us recall that it will not converge to the optimal policy but to a stochastic policy that is optimal under the policy entropy regularized objective. We elaborate this last point further in Appendix A.5.

D.2 Random MDPs domain

The Random MDPs experiment described in Section C.2 is intended to test the updates in random MDPs, that have not been designed for the on-policy updates to fail, and where they are known to be efficient. We will consider a single experiment consisting in repeating the following process for more than 100 runs:

1. Generate a Random MDP.
2. Run the policy optimization in this MDP with the same densities as those described in Section D.1.



(a) Direct param, mean performance

(b) Direct param, decile performance

(c) Softmax param, mean performance

(d) Softmax param, decile performance

Figure 7: Random MDPs experiments (50 states, 4 actions) under A1.

We then report the mean (resp. decile) of the normalized expected return over the runs, as the mean (resp. decile) performance. We run the same experiment with both the direct and softmax parametrizations.

D.2.1 Direct parametrization

The mean performance is displayed on Fig. 7a. All algorithms perform almost equally good. Indeed, in absence of purposely designed deceptive rewards, the policy gradient and the undiscounted update are rather reliable at finding strong policies. Still, we start observing from update 2,000 on that the updates with low off-policy o_t updates (*i.e.* small weight on the uniform policy) start losing ground to the ones with high off-policiness o_t. The notable exception is d_u that suffered some lateness at the start of curve, but is catching up and overtaking the low off-policy o_t curves around the 4,000th update. This suggests that a proper scheduling of o_t might be worth studying in the future: in early stages, follow the true gradient and increase exploration when the value improvement slows down.

The decile performance is displayed on Fig. 7b. We observe that, in addition to yielding a higher mean performance, high off-policy o_t updates are also very stable: their decile performance is very close to their mean performance. In contrast, the decile performance of the on-policy updates seem to stagnate around 0.99 at the end of the curve.

D.2.2 Softmax parametrization

The mean performance is displayed on Fig. 7c. The results are similar to that with the direct parametrization except that:

- the mean performance remains significantly further from the optimal (around 0.99 here vs 0.997 with direct parametrization). This is due to the fact that softmax parametrization takes longer to converge.

- the gap between high and low off-policy o_t updates is much thinner. We believe this might also be due to the slowness of the softmax parametrization convergence. As a consequence, d_u remains far from the other updates.
- the policy entropy regularized versions of the on-policy updates perform much worse. Indeed, the bias introduced by the policy entropy regularization impairs the convergence to a policy that is near optimal with respect to the normalized expected return (which is our objective function). Nevertheless, it has to be noted that it produced the faster improvement in the early stages (until update 1,000). We do not have an explanation for that.

The decile performance is displayed on Fig. 7d. We observe the same behavior as with the direct parametrization, with the exception of the narrower gaps already discussed above.

D.3 General conclusion for this set of experiments

We test the performance across time, actor learning rate η, MDP parameters $|\mathcal{S}|$ and β, off-policiness o_t, and policy entropy regularization λ, with both direct and softmax parametrizations, on both the chain and random MDPs. The full report is available in Appendix D.

On the chain domain, we empirically confirm that enforcing updates with d_t including an off-policy component helps path discovery and policy planning, while on-policy updates fail at converging to the optimal policy, even with policy entropy regularization, in a reasonable amount of time. By playing with the chain domain hyperparameters β and $|\mathcal{S}|$, we observe that on-policy updates are sensitive to both: even with a small $\beta = 0.1$, a reasonable sized chain $|\mathcal{S}| = 15$ cannot be solved. In contrast, updates that are partly off-policy allow to converge fast to optimality even with $\beta = 0.95$ and $|\mathcal{S}| = 25$.

On the random MDP domain, we empirically observe that discounted and undiscounted updates perform well but still oftentimes stagnate at 99% of optimal performance, while d_t including an off-policy component achieves even closer to optimality. Finally, we also note that a purely uniform d_t slows down training in the random MDPs experiment. We conclude that it is best to include both on-policy and off-policy components in d_t. Experiments also allowed us to observe the biased convergence implied by the policy entropy regularization.

E Full report of finite MDPs reinforcement learning experiments

E.1 Algorithms design

This section provides the full implementation of the algorithms on the testbed.

```
Input: UCB reward bonus weight $\nu$ and policy entropy regularization weight $\lambda$.
Input: Initialization of the value $q_0$, set actor learning rate $\eta$ and critics learning rate $\eta_c$.
Initialize parameters $\theta = 0$ of the actor $\pi \doteq \pi_\theta$ and the critic $q \doteq q_0$.
for $t = 0$ to $\infty$ do
    Sample a transition $\tau_t = \langle s_t, a_t \sim \pi_b(\cdot|s_t), s_{t+1} \sim p(\cdot|s_t, a_t), r_t \sim r(\cdot|s_t, a_t) \rangle$.
    for once, do a single on-policy update:
        Update critic $q$ with a SARSA update on $\tau_t$:
            $$q(s_t, a_t) \leftarrow q(s_t, a_t) + \eta_c \left( r + \gamma \sum_{a' \in \mathcal{A}} q(s', a') \pi(s', a') - q(s, a) \right) \tag{182}$$
        Perform a stochastic update step in state $s$ on Dr Jekyll's actor $\theta$:
            $$\theta_{s_t,a_t} \leftarrow \theta_{s_t,a_t} + \eta d \left( r + \nu \sqrt{\frac{\log t}{n_{s_t,a_t}}} + \lambda \log \pi(a_t|s_t) + \sum_{a' \in \mathcal{A}} \pi(s'_y, a') q(s'_t, a') \right) \nabla_\theta \pi(a_t|s_t) \tag{183}$$
    end for
end for
```

Algorithm 2: On-policy algorithms for experiments in finite MDPs (by default $\nu = 0$ and $\lambda = 0$).

We first formally describe the on-policy algorithms in Algorithm 2. PG update has d equal to γ at the power of the episode's timestep, while $d = 1$ for undiscounted updates. Unless specified otherwise UCB exploration bonus parameter ν and policy entropy regularization parameter λ are set to 0.

We then formally describe the implementation of J&H in Algorithm 3.

Not shown in the algorithms: in order to compare equally each update (for instance, discounted updates would observe less update amplitude than undiscounted ones), we normalize the empirical update steps d by the average of their values in history. This normalizing step is not useful for the algorithms themselves, its only use is to control potential side effects of the update choice.

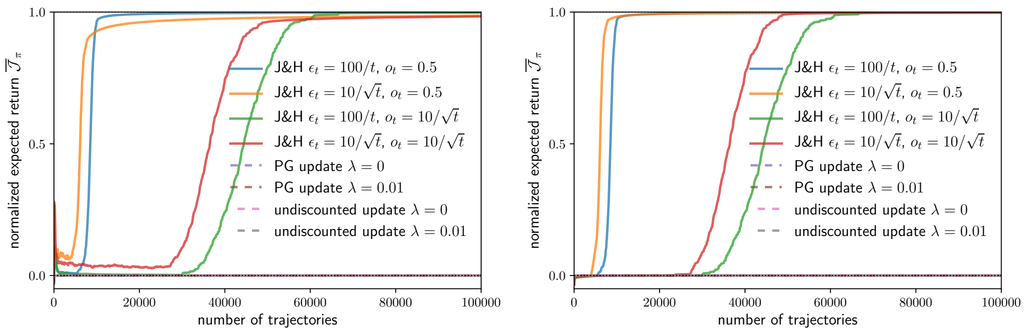
In order to prepare for its deep RL implementation, we extensively analyse the J&H algorithm, but only on the softmax parametrization. Indeed, there is no true equivalent to direct parametrization with neural networks. We analysed the performance of the algorithms against time (number of updates), the off-policiness o_t, the actor learning rate η, and the parameters of the chain domain: its size and the value ratio β. We will look at the same hyperparameters in the Reinforcement Learning setting, but also at the newly introduced hyperparameters: the critic learning rate η_c, the q-function initialization q_0, and the exploration schedule ϵ_t.

But first let us specify the default hyperparameter settings:

- Environment hyperparameters:
  - Chain domain: $|\mathcal{S}| = 10$, $|\mathcal{A}| = 2$, $\gamma = 0.99$, $\beta = 0.8$.
  - Random MDPs domain: $|\mathcal{S}| = 100$, $|\mathcal{A}| = 4$, $\gamma = 0.99$.
- Algorithms hyperparameters:
  - All algorithms: $\eta = 1$, $\eta_c = 0.1$, $q_0 = 0$, $\lambda = 0$, $\nu = 0$.
  - J&H specific: ϵ_t and o_t are always shown.
  - In all the experiments, Mr Hyde is trained with Q-learning on a γ discounted objective from UCB rewards: $\tilde{r}(s, a) \doteq \frac{1}{\sqrt{n_{s,a}}}$.

```
Input: Scheduling of exploration $(\epsilon_t) = \frac{\epsilon_1}{t^{\alpha_\epsilon}}$, and of off-policiness $(o_t) = \frac{o_1}{t^{\alpha_o}}$.
Input: Initialization of the value $q_0$, set actor learning rate $\eta$ and critics learning rate $\eta_c$.
Initialize $\mathring{D} = \emptyset$, parameters $\mathring{\theta} = 0$ of Dr Jekyll's actor $\mathring{\pi} \doteq \pi_{\mathring{\theta}}$, and critic $\mathring{q} \doteq q_0$.
Initialize $\tilde{D} = \emptyset$ and Mr Hyde's value to $\tilde{q} = q_0$, and policy $\tilde{\pi}$ to uniform.
Set the behavioural policy to Dr Jekyll: $\pi_b \leftarrow \mathring{\pi}$ and the working replay buffer $D_b \leftarrow \mathring{D}$.
for $t = 0$ to $\infty$ do
    Sample a transition $\tau_t = \langle s_t, a_t \sim \pi_b(\cdot|s_t), s_{t+1} \sim p(\cdot|s_t, a_t), r_t \sim r(\cdot|s_t, a_t)\rangle$
    Add it to the working replay buffer $D_b \leftarrow D_b \cup \{\tau_t\}$.
    if $\tau$ was terminal, then $(\pi_b, D_b) \leftarrow (\tilde{\pi}, \tilde{D})$ w.p. $\epsilon_t$, $(\pi_b, D_b) \leftarrow (\mathring{\pi}, \mathring{D})$ otherwise.
    for once, do a single update:
        $\tau \doteq \langle s, a, s', r\rangle \sim \tilde{D}$ w.p. $o_t$, $\tau \doteq \langle s, a, s', r\rangle \sim \mathring{D}$ otherwise.
        Update Mr Hyde's critic $\tilde{q}$ with a $q$-learning update on $\tau$, then $\tilde{\pi}$ is greedy on $\tilde{q}$:
            $\tilde{q}(s,a) \leftarrow \tilde{q}(s,a) + \eta_c \left(\frac{1}{\sqrt{n_{s,a}}} + \gamma \max_{a' \in \mathcal{A}} \tilde{q}(s',a') - \tilde{q}(s,a)\right) \quad \text{then} \quad \tilde{\pi} \leftarrow \operatorname*{argmax}_{a \in \mathcal{A}} \tilde{q} \quad (184)$
        Update Dr Jekyll's critic $\mathring{q}$ with a SARSA update on $\tau$:
            $\mathring{q}(s,a) \leftarrow \mathring{q}(s,a) + \eta_c \left(r + \gamma \sum_{a' \in \mathcal{A}} \mathring{q}(s',a') \mathring{\pi}(s',a') - \mathring{q}(s,a)\right) \quad (185)$
        Perform an expected update step in state $s$ on Dr Jekyll's actor $\mathring{\theta}$:
            $\forall b \in \mathcal{A}, \qquad \mathring{\theta} \leftarrow \mathring{\theta} + \eta \mathring{q}(s,b) \nabla_{\mathring{\theta}} \mathring{\pi}(b|s) \quad (186)$
    end for
end for
```

Algorithm 3: Dr Jekyll & Mr Hyde algorithm for experiments in finite MDPs.



(a) Mean global performance

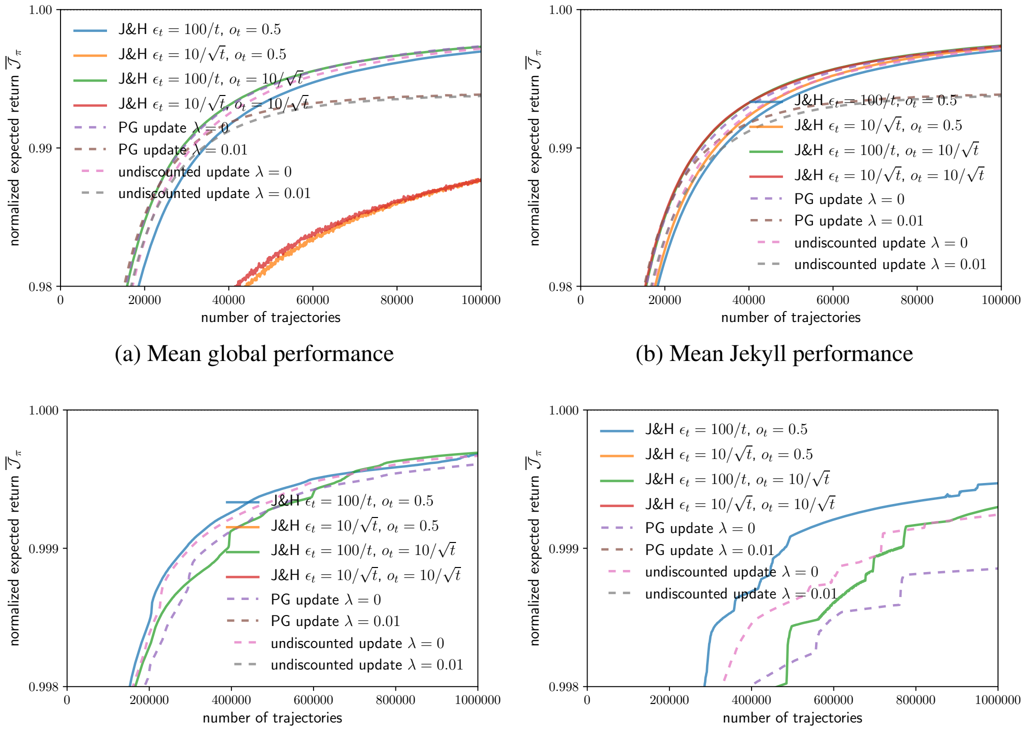
(b) Mean Jekyll performance

Figure 8: Chain MDPs experiments vs. number of trajectories (200+ runs).

For J&H, we generally report the *global* performance, *i.e.* the expected performance of the mixture of Dr Jekyll and Mr Hyde, but also sometimes, when mentioned, we also look at the performance of the Dr Jekyll policy, because it is the best policy known to the algorithm at a given time.

E.2 Vs. time (number of trajectories):

In the chain experiment, we expect J&H to perform much better than on-policy updates (policy gradient and undiscounted), since they are known to easily converge to suboptimal solutions. This is indeed what we observe on Fig. 8a. All versions of J&H eventually converge, in all runs, to the optimal policy, while on-policy updates, with or without entropic regularization, never converged to the optimal solution. Looking more closely at the different J&H variants, we notice that the off-policiness is instrumental to make the convergence faster, more than the exploration ratio ϵ_t, because this latter cannot be set too large in order not to compromise the global performance. Indeed, we



(a) Mean global performance

(b) Mean Jekyll performance

(c) Mean global performance (long)

(d) Decile global performance (long)

Figure 9: Random MDPs experiments vs. number of trajectories.

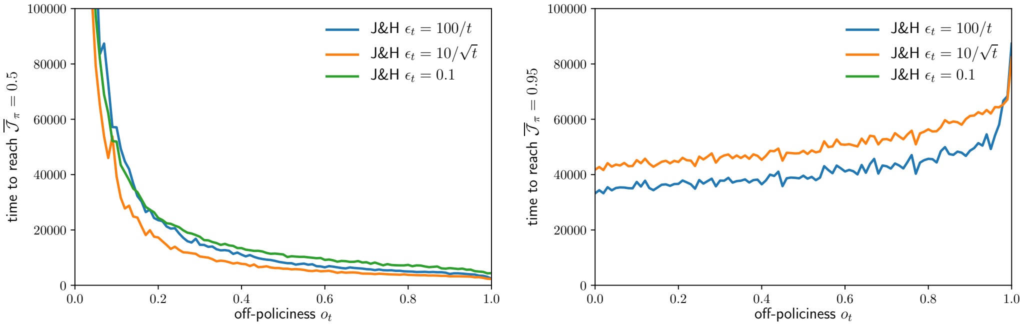
already observe with $\epsilon_t = \frac{10}{\sqrt{t}}$ that exploration prevents from converging closely to optimal in global performance. We may confirm that Dr Jekyll converges well to optimal on Fig. 8b.

In the random MDPs domain, we do not expect to see any improvement, the on-policy updates (policy gradient and undiscounted) are not *tricked* to converge to a suboptimal solution. Moreover, the high stochasticity of the domain, makes it so that every state is visited a lot of times even when following a deterministic policy. Fig. 9a confirms that J&H can keep up with the speed of the true gradient. Its versions with $\epsilon_t = \frac{10}{\sqrt{t}}$ suffer a low global performance because of its exploration that remains high, as Fig. 9b confirms. We notice that entropic regularization does not help training faster and also biases the training objective which results into a plateaued performance. Finally, we may observe that the choice of o_t has a slight impact: using half Dr Jekyll, half Mr Hyde samples to train offline slows a bit the convergence in the first 100k trajectories, however, looking at a longer horizon on Fig. 9c shows that it gets better later[10]. It is also interesting to notice that the gap to optimal performance has decreased by a factor 10 with 10 times more data, which corroborates our theoretical findings that the regret should be in $\mathcal{O}(t^{-1})$. Finally, the decile performance across time shown on Fig. 9d suggests that the use of strong off-policiness mitigates the risk and improves the worst case performance.

E.3 Vs. off-policiness o_t:

Off-policiness being a hyperparameter of J&H alone, we report the results for J&H only. On Fig. 10a, we observe that the higher the off-policiness, the best better it is in the chain experiment. However, it is quite the opposite in the random MDPs experiments results shown on Fig. 10b, even though it is less critical. The scheduling of o_t would be an interesting focus for further research. At this point of our knowledge we conjecture that it is better to train on-policy when the expected value improvement are high, and train more off-policy when it is small.

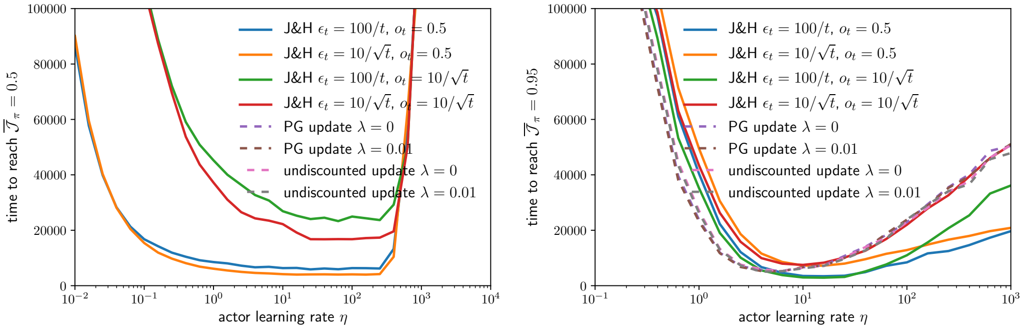
[10] Note that despite using more than 100 runs to compute these curves, there remains some instability and the visible results are probably not significant enough to elect a winner, however they are enough to see that they perform comparably.



(a) Chain experiment

(b) Random MDPs experiment

Figure 10: Experiments vs. off-policiness o_t: average time to reach a global performance objective.



(a) Chain experiment

(b) Random MDPs experiment

Figure 11: Experiments vs. actor learning rate η: average time to reach a global performance objective.

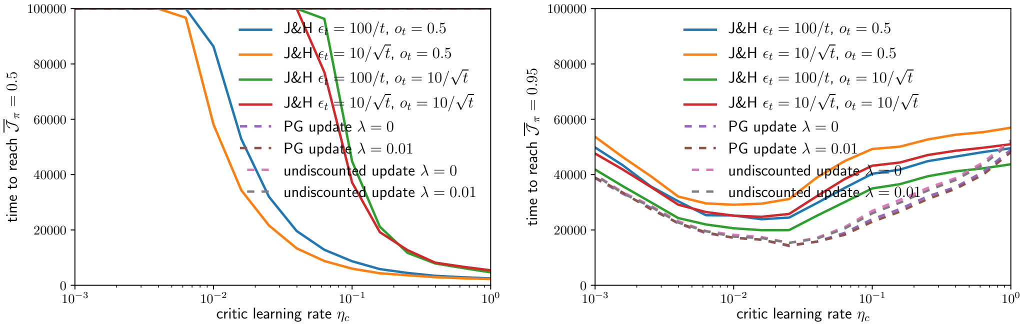
E.4 Vs. learning rates η and η_c:

The study of η on the chain domain (results on Fig. 11a reveals that the performance of J&H improves until some value where it breaks. It looks like a numerical stability issue but after checking it does not seem like anything of the sort, rather that the softmax parameters go faster away than they can go back. Indeed, as our theoretical analysis shows, there is a factor in $\pi(1-\pi)$ the updates that slows down the recovery from a run that converged too far into a suboptimal policy. Actually, this phenomenon has been observed before on policy gradients, *e.g.* on Fig. 1.d of [22]. Note that the value of this hyperparameter does not allow to help the on-policy algorithms to find the optimal solution. We made the similar experiment on easier settings (lower β and size), and this did not improve (not shown). Note also that the default value for the actor learning rate has been set to 1 which is not its optimal value, but it was a better value to make the critic/actor learning rates consistent with each other.

The experiment on the Random MDPs shown on Fig. 11b is one of the most interesting empirical results for many reasons[11]. Firstly, we observe that all updates admit an optimal value for the actor learning rate in the range [2,30]: rather high, but not too high. This shows that there is a value for not updating too fast the policy. Secondly, off-policiness is more tolerant for higher η values: its optimal value is obtained for $\eta \approx 20$, while on-policy updates are best with $\eta \approx 5$. Thirdly, when η is optimally set for each update, J&H significantly outperforms on-policy updates ($\approx$ 2,500 vs. 3,500). Finally, it is satisfying to observe that $\eta \approx 20$ is a also optimal in the chain experiment, suggesting that it may not be too much domain dependent.

Since the chain domain is deterministic, we would expect a high critic learning rate η_c to be better and this is what we observe, but even a maximal critic learning rate of 1 is not enough for the on-policy updates with or without policy entropy regularization to solve the chain is the default settings

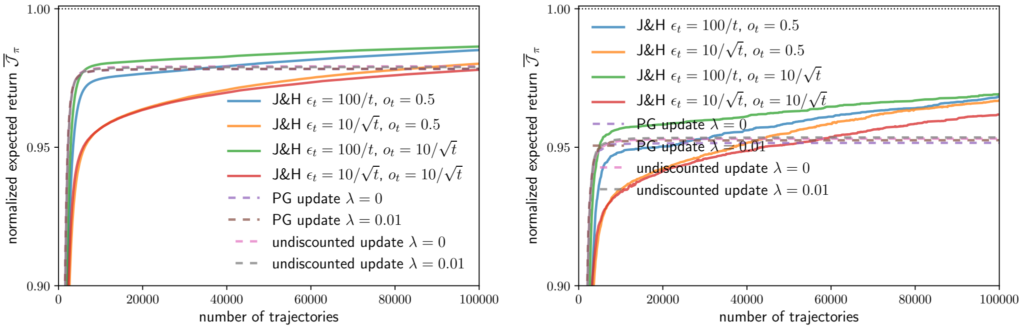
[11] We notice that the default value we used for this hyperparameter is not favorable for J&H.



(a) Chain experiment

(b) Random MDPs experiment

Figure 12: Experiments vs. critic learning rate η_c: average time to reach a global performance objective.



(a) Mean performance

(b) Decile performance

Figure 13: Random MDPs experiments with $\eta = 10$ and $\eta_c = 0.025$.

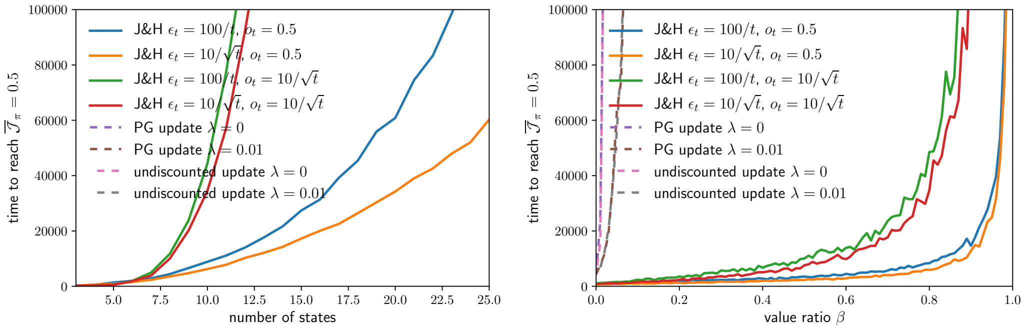
(size of 10 and $\beta = 0.8$). We also not that J&H is much more tolerant to a small critic learning rate when the off-policiness is kept high. The random MDP experiment (Fig. 12b) illustrates the risk of setting a too high critic learning rate. A small learning rate implies a longer but steadier time to converge, while a high learning rate might compromise the convergence at all.

One must keep in mind that these insights are for reaching a normalized performance of 0.95 in the random MDPs experiments. We tried a run with $\eta = 10$ and $\eta_c = 0.025$ for all algorithms, since they seem to all perform well with this setting in our hyperparameter search we have just presented. Fig. 13 displays the results. It clearly appears that reaching $\overline{\mathcal{J}}_\pi = 0.99$ (let alone 0.999) would take much longer than with our default setting. We judged better to stick with our default settings for this reason.

E.5 Vs. parameters of the chain domain:

On Fig. 14a, we vary the size of the chain, and observe that J&H algorithm scales well to it as long as the off-policiness is maintained long enough for it to unlearn to suboptimal policy and move on to the optimal policy. It is worth noting that the on-policy updates fail at finding even with its easiest hyperparameters, indeed the q-function being initialized to 0 and the critic learning rate being of 0.1 makes it that the algorithms already converged to the suboptimal policy when the later states values are properly estimated. In fact, the policy entropy regularized versions of the on-policy updates reach $\overline{\mathcal{J}}_\pi = 0.48$ around the 150,000th episode.

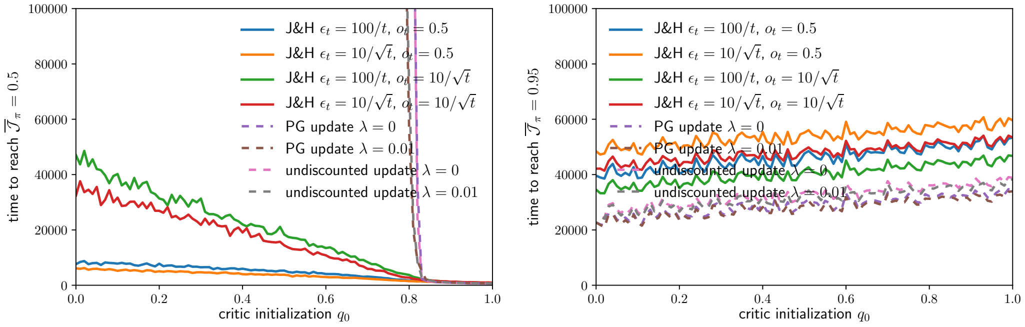
On Fig. 14a, we vary the performance ratio β and observe that the on-policy updates manage to make it work even for a chain of size 10 when β is sufficiently small, and that the policy entropy regularization helps a bit to push further the limit, but not by much. In general, the delay occurred by the time necessary for the critic to be trained is fatal to these algorithms, hence the importance of



(a) vs. chain size

(b) vs. performance ratio β

Figure 14: Experiments vs. chain hyperparameters: average time to reach a global performance objective.



(a) Chain experiment

(b) Random MDPs experiment

Figure 15: Experiments vs. q-function initialization q_0: average time to reach a global performance objective.

prioritizing the replay of exploratory experiences. Fig. 14a also validates the reliability of the J&H algorithms, in particular with a strong off-policiness.

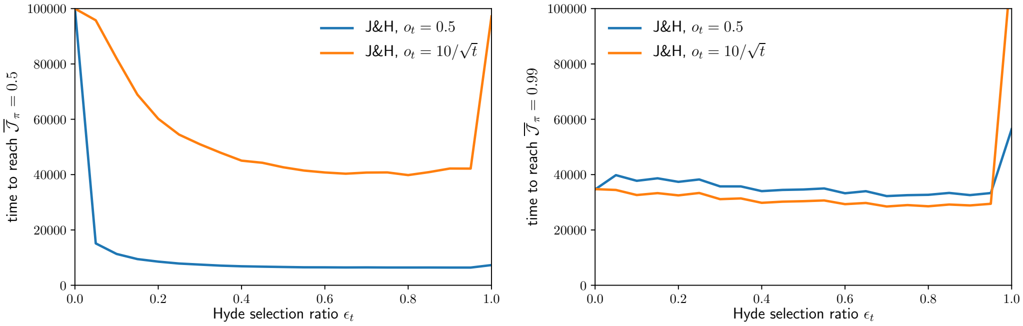
E.6 Vs. q-function initialization q_0:

Initializing the q-function to high values has been a historic exploration scheme for q-learning, and we observe on Fig. 12a that is also works in the actor-critic world. As soon as q_0 gets larger than the value of the low-hanging-fruit policy, all updates push to explore until discovering the end of the chain. In contrast, J&H is not as much sensitive to the critic initialization, in particular when the off-policiness is high enough to fight against a hostile one.

E.7 Vs. exploration schedule ϵ_t:

Exploration schedule being a hyperparameter of J&H alone, we report the results for J&H only. In order to remove the dependence of the performance on its exploratory behaviour, contrarily to all the other experiments where we report the performance of J&H's dyad, we report here the time for Jekyll's policy to reach the normalized value.

On Fig. 16a, we observe that having a strong off-policiness alleviates the lack of exploration to a certain extent. We however want to reiterate that the chain domain is deterministic, and as such, performing off-policy updates is equivalent to exploring. In a stochastic environment such as the random MDPs, exploration cannot be replaced with off-policy updates, because statistical significance is required to find the optimal policy. With $\epsilon_t = 0$, unsurprisingly, the algorithm fails at finding the optimal reward and therefore finding the goal. From $\epsilon_t = 0.05$ to $\epsilon_t = 0.6$, the speed improves from 15000 to 6500. From $\epsilon_t = 0.6$ to $\epsilon_t = 0.95$, the speed remains more of less constant in the range [6400, 6500]. With $\epsilon_t = 1$, more surprisingly, the performance degrades quite significantly to 7300.



(a) Chain experiment

(b) Random MDPs experiment

Figure 16: Experiments vs. exploration schedule ϵ_t: average time to reach a global performance objective.

The random MDPs experiment (Fig. 16b) displays similar behaviour at a different scale. From $\epsilon_t = 0$ to $\epsilon_t = 0.95$, the speed linearly improves from 35000 to 29000. With $\epsilon_t = 1$, again surprisingly, the performance degrades catastrophically to more than 100000.

We analyse the combined results as follows: introducing Hyde's exploratory behavior generates a large performance improvement due to a need for exploration. This need is correlated with the planning difficulty of the task, hence the much stronger effect in the chain experiment. Performance then slowly improves as a function of ϵ_t: this is where the statistical significance matters. Finally, when ϵ_t equals 1, the on-policy updates do not exist anymore, slowing down significantly the convergence speed.

E.8 General conclusion for this set of experiments

We test the performance across time, actor η and critic η_c learning rates, MDP parameters $|\mathcal{S}|$ and β, off-policiness o_t and exploration ϵ_t, and critic initialization q_0 with the softmax parametrization on both the chain and random MDPs. The full report is available in Appendix E.

In the chain experiments, we observe that J&H shows the same ability to converge to the optimal policy, while the on-policy updates, with or without policy entropy regularization, always fail, except when $q_0 > \mathcal{J}_\perp$ because it induces an exploratory behaviour that solves its limits. In contrast, J&H is efficient whatever q_0. More specifically about J&H, while the exploration ϵ_t is necessary at a small amount, the amplitude of off-policiness o_t is what matters the most. Since the chain domain is deterministic, the critic learning rate η_c is optimal set at 1. Regarding the actor learning rate, it is best set high too, but until some limit where it breaks the convergence.

Regarding the random MDPs experiments, the results are much closer, the on-policy updates being helped by the high stochasticity of the environment, providing indeed a sort of exploration. We still observe that all algorithms provide similar performance, J&H performing slightly worse when o_t or ϵ_t increase. Similarly, since exploration is not necessary in this domain, the performance slightly decrease with q_0 for all algorithms. Regarding the critic learning rate, it is best chosen around $\eta_c = 0.025$. Maybe the most interesting result concern the dependency with respect to the actor learning rate. It seems that the off-policiness granted by J&H updates allows for higher η, which allows faster convergence when this hyperparameter is optimized independently for each algorithm. While expected, another interesting observation is that off-policiness seems to improve the worst case scenarios by allowing to recover more easily from converging to a suboptimal policy.

F Full report of deep reinforcement learning experiments

F.1 The Four Rooms environment

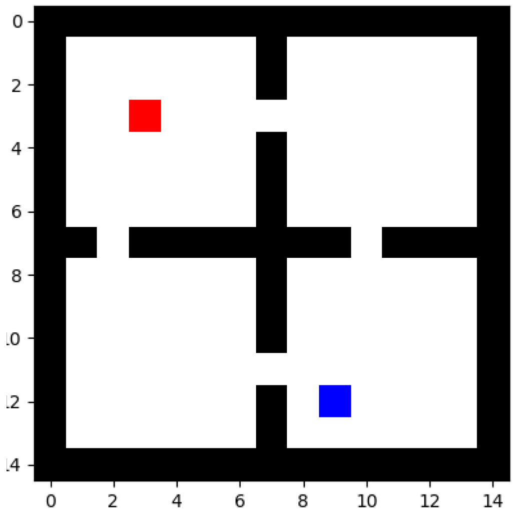


Figure 17: The Four Rooms environment, the agent (in blue) needs to navigate to the goal (in red).

Figure 17 shows an example of the environment. The agent (in blue) starts at a random place in one of the rooms. In Level 1, the initial state can be in any room, in Level 2, the initial state cannot be in the goal room. The agent then needs to navigate to the goal, using four actions, North, East, West, South. At each step in the environment, the agent receives a negative reward of -0.1. When reaching the goal, the agent receives a reward of 90 and the episode terminates. If the agent hasn't reached the goal in 90 steps, the episode terminates. In the plots, we report the undiscounted sum of rewards, a quantity that belongs to $[-9, 90]$. The state consists of a random 20-dimensional embedding for each distinct position in the environment. The discount factor is set to 0.9. After 10 episodes of training, we evaluate each agent during 10 episodes and report the average performance. Each agent was evaluated on 10 seeds, we report the average and one standard deviation over those 10 seeds. We trained the agents on an internal GPU cluster of P40 cards. Each training run took approximately 30 minutes.

F.2 Implementation, architectures and hyperparameters

We based our code on the repo [8], that contains PyTorch implementations of various deep reinforcement learning algorithms. We used the default architectures from the repo, the Adam optimizer and performed small hyperparameters sweeps on learning rates for each algorithm (described below).

DDQN We used the default implementation provided by the repo [8] for the Four Rooms environment. The q-network is a two-hidden layer MLP with 30 and 10 neurons in each. The replay buffer size is 40k and we used soft decay of the epsilon greedy exploration as well as soft updates of the target network. In terms of learning rate, we swept over $\{0.06, 0.03, 0.01, 0.006, 0.003, 0.001\}$ and found that 0.01 performed best. We also found that decreasing the epsilon greedy decay compared to the default value in the repo gave better performance on Level 2 (it did not affect Level 1 much). We used the following probability for picking a random action: $100/e$, where e is the number of episodes trained on so far (compared to 10 for the default).

Soft-Actor Critic The actor and critics are two-hidden layers MLPs with 64 neurons in each. Two critics were trained, and the mininum between the two was used in the backup of the critics, as well as to train the actor. The entropy regularization hyperparameter was automatically tuned. The implementation is the standard one provided in [8]. We also swept over the same learning rate set and found that similarly 0.01 performed best.

RND For the RND target network, we used a one-hidden layer MLP with 20 neurons, outputing a 20-dimensional feature vector. The predictor network contains (as is customary in RND implementations) an additional hidden layer with 20 neurons as well. We did not search over architectures. The learning rate is set to 0.003, which was found after a sweep on the set $\{0.1, 0.03, 0.01, 0.003, 0.001\}$.

Dr Jekyll & Mr Hyde Dr Jekyll's architecture is the same as the Soft-Actor Critic one described above. Its learning rate was set to 0.001, which was found to perform best out of the set $\{0.06, 0.03, 0.01, 0.006, 0.003, 0.001, 0.0006\}$. Mr Hyde's architecture and learning rate are the same as the DDQN architecture described above. The decay of the probability of picking Mr. Hyde (parameter ϵ_1 in algorithm 4) is set to $4k$ for Level 1 and $40k$ for Level 2. Note that this corresponds to the number of steps in the environment, not the number of episodes played. Those values correspond approximately to the decay of the exploration in DDQN.

DDQN+RND and SAC+RND In the case of DDQN+RND, we simply augmented the reward of the environment using the exploration bonus provided by the RND network. For SAC+RND, we trained an additional critic on the exploration bonus and added it to SAC's actor update. In both cases, we weighted the exploration bonus by 1 afer sweeping over the set $\{0.1, 0.3, 0.6, 1.0, 1.3, 1.6\}$ and finding that it performed best in both cases.

F.3 The detailed algorithm

Algorithm 4 describes the implementation used to produce Figure 3.

```
**Input:** Scheduling of exploration $(\epsilon_t) = \frac{\epsilon_1}{t}$.
Initialize Dr Jekyll's actor $\mathring{\pi} \doteq \pi_{\mathring{\theta}}$ and critic $\mathring{q} = q_{\mathring{\theta}}$.
Initialize Mr Hyde's q-network $\tilde{q} = q_{\tilde{\theta}}$ (its policy $\tilde{\pi}$ is greedy with respect to $\tilde{q}$).
Initialize the RND predictor $p_{RND}$ and target $t_{RND}$ networks.
Initialize a buffer $F = \emptyset$ of capacity $K = 128$ and a buffer $\tilde{D} = \emptyset$ of capacity $40k$.
Set the behavioural policy to Dr Jekyll: $\pi_b \leftarrow \mathring{\pi}$ and the working replay buffer $D_b \leftarrow \mathring{D}$.
**for** $t = 0$ to $\infty$ **do**
    Sample a transition $\tau_t = \langle s_t, a_t \sim \pi_b(\cdot|s_t), s_{t+1} \sim p(\cdot|s_t, a_t), r_t \sim r(\cdot|s_t, a_t)\rangle$.
    **if** Mr Hyde is in control ($\pi_b = \tilde{\pi}$), **then** add $\tau_t$ to Mr Hyde's replay buffer $\tilde{D}$ .
    Add $\tau_t$ to buffer $F$.
    **if** $\tau_t$ was terminal, **then** $\pi_b \leftarrow \tilde{\pi}$ wp $\epsilon_t$, $\pi_b \leftarrow \mathring{\pi}$ otherwise.
    Initialize minibatch with FIFO buffer: $B \leftarrow F$
    **for** $i = 1 \rightarrow K$
        Add $\tau_i \doteq \langle s, a, s', r\rangle \sim \tilde{D}$ to minibatch $B$
    **end for**
    For each transition $\tau_i \in B$, compute RND reward $\tilde{r}_{\tau_i} = \|p_{RND}(s') - t_{RND}(s')\|^2$.
    Update Mr Hyde's q-network $\tilde{q}$ with a DDQN update on $B$ where $\tilde{r}$ is used instead of $r$
        $\tilde{q}(s,a) \leftarrow \tilde{q}(s,a) + \eta_c \left(\tilde{r}(s,a) + \gamma \max_{a' \in \mathcal{A}} \tilde{q}(s',a') - \tilde{q}(s,a)\right).$   (187)
    Update Dr Jekyll's critic $\mathring{q}$ with a SARSA update on $B$:
        $\mathring{q}(s,a) \leftarrow \mathring{q}(s,a) + \eta_c \left(r + \gamma \sum_{a' \in \mathcal{A}} \mathring{q}(s',a')\mathring{\pi}(s',a') - \mathring{q}(s,a)\right).$   (188)
    Perform an expected update step in each state $s$ of $B$ on Dr Jekyll's actor $\mathring{\theta}$:
        $\forall b \in \mathcal{A}, \qquad \mathring{\theta} \leftarrow \mathring{\theta} + \eta \mathring{q}(s,b) \nabla_{\mathring{\theta}} \mathring{\pi}(b|s)$   (189)
    Update $p_{RND}$ by gradient descent on $\sum_{s' \in B} \|p_{RND}(s') - t_{RND}(s')\|^2$.
**end for**
```

Algorithm 4: Dr Jekyll & Mr Hyde algorithm for experiments with a deep architecture.

G Code description

The code for all our experiments can be found at https://github.com/microsoft/Dr-Jekyll-and-Mr-Hyde-The-Strange-Case-of-Off-Policy-Policy-Updates.

G.1 Finite MDPs code

It relies on two main files:

- `main_config.py` to run experiments in a fixed setting and report average/quantile performance across time,
- and `main_hyperparam.py` to run experiments where a hyperparameter varies, and report the time to reach a fixed normalized performance target $\tilde{\mathcal{J}}$ (see Eq. (180) and (181)).

They both use the same configuration file determining the setting the experiment (the hyperparameter search overwrites the setting of this hyperparameter during the run): which setting (planning or RL), which environment (chain or random MDPs), which algorithms, with which hyperparameters, for how many steps, and for how many runs. Such configuration files may be found in the `expes` folder, where the settings of all reported experiments may be retrieved as well as their figures (and some more). Nevertheless, we deleted the experiment result files because they are voluminous (10s of Go), but we have them and can provide them on demand, or better: their random seeds.

The two environments are implemented in python files:

- `chain.py`: the chain environment described in App. C.1,
- and `garnets.py`: the random MDPs environment described in App. C.2 [18].

Two utility python files are used:

- `utils.py` contains all utility files,
- except `proj_simplex.py` implementing the projection on the simplex, that has been isolated in order to account for their authors [5].

Finally, all the algorithmic innovation that we claim belongs to `gradient_ascent.py`, including the implementation of J&H (Alg. 3).

G.2 Deep J&H implementation

The implementation is based on the repository [8], released under the MIT License and which contains various PyTorch implementations of standard deep reinforcement learning algorithms. The entry point to train the agents is the `results/Four_Rooms.py` file. To train a given one, simply run:

```
python results/Four_Rooms.py -agent {agent} -level {level}
```

where agent is chosen in [JH_Discrete, SAC_Discrete, SAC_DiscreteRND, DDQN, DDQNRND] and level is 1 or 2.