Leverage the Average: an Analysis of Regularization in RL

Nino Vieillard [1 2] **Tadashi Kozuno** [3 *] **Bruno Scherrer** [2] **Olivier Pietquin** [1] **Rémi Munos** [4] **Matthieu Geist** [1]

Abstract

Building upon the formalism of regularized Markov decision processes, we study the effect of Kullback-Leibler (KL) and entropy regularization in reinforcement learning. Through an equivalent formulation of the related approximate dynamic programming (ADP) scheme, we show that a KL penalty amounts to averaging q-values. This equivalence allows drawing connections between *a priori* disconnected methods from the literature, and proving that a KL regularization indeed leads to averaging errors made at each iteration of value function update. With the proposed theoretical analysis, we also study the interplay between KL and entropy regularization. When the considered ADP scheme is combined with neural-network-based stochastic approximations, the equivalence is lost, which suggests a number of different ways to do regularization. Because this goes beyond what we can analyse theoretically, we extensively study this aspect empirically.

1. Introduction

Regularization in Reinforcement Learning (RL) usually amounts to adding a penalty term to the greedy step of a dynamic programming scheme. For example, soft Q-learning (Fox et al., 2016; Schulman et al., 2017; Haarnoja et al., 2017) uses a Shannon entropy regularization in a Value Iteration (VI) scheme, while Soft Actor Critic (SAC) (Haarnoja et al., 2018) uses it in a Policy Iteration (PI) scheme. Other approaches penalize the divergence between consecutive policies. Trust Region Policy Optimization (TRPO) (Schulman et al., 2015) is such a PI-scheme, with the greedy step being penalized with a Kullback-Leibler (KL) divergence. Maximum a Posteriori Policy Optimization (Abdolmaleki et al., 2018) is derived from a rather different principle, but the resulting algorithm is quite close, the main difference lying in how the greedy step is approximated. The generic regularized Dynamic Programming (DP) scheme we consider in this paper encompasses (variations of) these approaches, among others.

Algorithms making use of regularization enjoy good empirical performances. However, the reason of this efficiency is not well understood. Some authors (*e.g.*, Schulman et al. (2017)) advocate that having a higher entropy helps exploration. Ahmed et al. (2019) claim that its benefit comes also from its effect on the optimization landscape. The KL penalty of TRPO was introduced as a practical proxy for the stochastic mixture of the theoretically sound Conservative Policy Iteration approach (Kakade & Langford, 2002). Geist et al. (2019) formalized and analyzed a large set of regularized RL algorithms. However, their analysis does not show why regularization helps. In this paper, we propose an alternative explanation of the benefit of using regularization in RL. We show, under some assumptions, that using a KL penalty (penalizing the new policy from being too far from the previous one), possibly in conjunction with an entropy penalty (penalizing near deterministic policies), allows for an averaging of the errors made by estimating value functions over iterations.

To do so, we build upon the formalism introduced by Geist et al. (2019). We study a variation of their Mirror Descent Modified PI (MD-MPI) framework, more restrictive in some sense (only entropy or KL) but more general in others (we can consider both at the same time). We frame an equivalent Dual Averaging MPI (DA-MPI) framework, inspired by the equivalence between mirror descent and dual averaging, in some cases, for convex optimization (*e.g.*, McMahan (2010)). It will be used for the theoretical analysis, restricted to a VI scheme, its extension to a general MPI scheme remaining an open question.

A limitation of our analysis is that we account for errors in the evaluation step (value function estimation), while we do not in the greedy step (policy improvement). Yet, practical neural-network-based implementations of the general considered regularized ADP scheme involve both errors. In this case, the formal equivalence between MD-MPI and DA-MPI no longer holds. As this goes beyond our theory, we provide an extensive empirical comparison of the different ways of doing regularization.

[1] Google Research, Brain Team [2] Université de Lorraine, CNRS, Inria, IECL, F-54000 Nancy, France [3] Okinawa Institute of Science and Technology, Okinawa, Japan [4] DeepMind *Work done during an internship at Deepmind.

2. Background and Notations

We write Δ_X the set of probability distributions over a finite set X and Y^X the set of applications from X to the set Y. A Markov Decision Process (MDP) is a tuple $\{\mathcal{S}, \mathcal{A}, P, r, \gamma\}$ with $\mathcal{S}$ the finite state space, $\mathcal{A}$ the finite set of actions, $P \in \Delta_{\mathcal{S}}^{\mathcal{S}\times\mathcal{A}}$ the Markovian transition kernel, $r \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ the reward function uniformly bounded by r_{max}, and $\gamma \in (0, 1)$ the discount factor. For $\tau \geq 0$, we write $v_{\text{max}}^{\tau} = \frac{r_{\text{max}} + \tau \ln |\mathcal{A}|}{1-\gamma}$ and simply $v_{\text{max}} = v_{\text{max}}^0$. We also write $\mathbf{1} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ the vector whose components are all equal to 1.

A policy $\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}$ associates a distribution over actions to each state. Its (state-action) value function is defined as

$$q_\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t) \middle| S_0 = s, A_0 = a \right],$$

$\mathbb{E}_\pi$ being the expectation over trajectories induced by π. An optimal policy is one with maximal value function, $\pi_* \in \operatorname{argmax}_{\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}} q_\pi$ (all scalar operators applied on vectors should be understood point-wise), and $q_* = q_{\pi_*}$.

The following notations will be useful. For $q_1, q_2 \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$,

$$\langle q_1, q_2 \rangle = \left(\sum_a q_1(s, a) q_2(s, a) \right)_s \in \mathbb{R}^{\mathcal{S}}.$$

We write P_π the stochastic kernel induced by π, and for $q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ we have

$$P_\pi q = \left(\sum_{s'} P(s'|s, a) \sum_{a'} \pi(a'|s') q(s', a') \right)_{s,a} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}.$$

For $v \in \mathbb{R}^{\mathcal{S}}$, we also define

$$Pv = \left(\sum_{s'} P(s'|s, a) v(s') \right)_{s,a} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}.$$

With these notations[1], we have $P_\pi q = P\langle \pi, q \rangle$.

The Bellman evaluation operator is $T_\pi q = r + \gamma P_\pi q$, its unique fixed point being q_π. The set of greedy policies w.r.t. $q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ is $\mathcal{G}(q) = \operatorname{argmax}_{\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}} \langle q, \pi \rangle$. A classical approach to compute an optimal policy is Modified Policy Iteration (MPI) (Puterman & Shin, 1978),

$$\begin{cases} \pi_{k+1} \in \mathcal{G}(q_k) \\ q_{k+1} = (T_{\pi_{k+1}})^m q_k + \epsilon_{k+1} \end{cases},$$

which reduces to VI (for $m = 1$) and PI (for $m = \infty$) as special cases. The term ϵ_{k+1} accounts for errors made when applying the Bellman operator. The classic use of m-step rollouts in (deep) RL actually usually corresponds to an MPI scheme with $m > 1$. In the next section, we add regularization to this scheme.

3. Regularized MPI

In this work, we consider the entropy and the KL divergence:

$$\mathcal{H}(\pi) = -\langle \pi, \ln \pi \rangle \in \mathbb{R}^{\mathcal{S}},$$
$$\mathrm{KL}(\pi_1 || \pi_2) = \langle \pi_1, \ln \pi_1 - \ln \pi_2 \rangle \in \mathbb{R}^{\mathcal{S}}.$$

First, we introduce a variation of the MD-MPI scheme originally proposed by Geist et al. (2019), who have not considered entropy and KL at the same time.

3.1. Mirror Descent MPI

For $q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ and an associated policy $\mu \in \Delta_{\mathcal{A}}^{\mathcal{S}}$, we define the regularized greedy policy as

$$\mathcal{G}_\mu^{\lambda,\tau}(q) = \operatorname*{argmax}_{\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}} \left(\langle \pi, q \rangle - \lambda \, \mathrm{KL}(\pi || \mu) + \tau \mathcal{H}(\pi) \right).$$

Observe that with $\lambda = \tau = 0$, we get the classic greediness. Notice also that with $\lambda = 0$, the KL term disappears, so does the dependency to μ. In this case we will write $\mathcal{G}^{0,\tau}$. One can also account or not for the regularization in the Bellman evaluation operator. Recall that the classic operator is $T_\pi q = r + \gamma P\langle \pi, q \rangle$. Given the form of the regularized greediness, it is natural to replace the term $\langle \pi, q \rangle$ by the regularized one, giving $T_{\pi|\mu}^{\lambda,\tau} q = r + \gamma P \left(\langle \pi, q \rangle - \lambda \, \mathrm{KL}(\pi || \mu) + \tau \mathcal{H}(\pi) \right)$. We refer to this as the Bellman operator of type 1, following the taxonomy of Geist et al. (2019):

$$T_{\pi|\mu}^1 q = T_{\pi|\mu}^{\lambda,\tau} q = T_\pi q - \gamma P \left(\lambda \, \mathrm{KL}(\pi || \mu) - \tau \mathcal{H}(\pi) \right)$$

Type 2 ignores the regularization in the evaluation step:

$$T_{\pi|\mu}^2 q_k = T_\pi q_k.$$

These lead to the following MD-MPI$_{1\text{-}2}$(λ,τ) scheme, the subscript denoting the type of evaluation. It is initialized with $q_0 \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ such that $\|q_0\|_\infty \leq v_{\text{max}}$ and with π_0 the uniform policy, without much loss of generality (notice that the greedy policy is unique whenever $\lambda > 0$ or $\tau > 0$):

$$\begin{cases} \pi_{k+1} = \mathcal{G}_{\pi_k}^{\lambda,\tau}(q_k) \\ q_{k+1} = (T_{\pi_{k+1}|\pi_k}^{1-2})^m q_k + \epsilon_{k+1} \end{cases}. \tag{1}$$

The term ϵ_k stands for the approximation error of the value function (such as the error due to learning the value function with a neural network and using a multi-step temporal difference).

Compared to the MD-MPI of Geist et al. (2019), we consider q-functions rather than value functions, but we handle a more specific case: they consider either a convex regularizer or the Bregman divergence generated by it, while we only consider the negative entropy or its associated KL divergence[2]. Yet, we consider also a more general case, as

[1] As $\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}} \subset \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, for $q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, we write $\langle \pi, q \rangle = (\sum_a \pi(a|s) q(s, a)))_s$.

[2] Note that our analysis could be adapted to a Bregman divergence generated by a convex regularizer of the Legendre type.

we handle the KL divergence and the entropy in the same scheme, while they consider them separately[3].

Scheme (1) encompasses a number of approaches (the following statements are justified in Appx. B.1). For example, SAC and soft Q-learning are variations of MD-MPI$_1$(0,τ), and softmax DQN (Song et al., 2019) is a variation of MD-MPI$_2$(0,τ). TRPO and MPO are variations of MD-MPI$_2$(λ,0). Dynamic Policy Programming (DPP) is almost a reparametrization of MD-MPI$_1$(λ,0), and Conservative Value Iteration (CVI) (Kozuno et al., 2019) is a reparametrization of MD-MPI$_1$(λ,τ), which consequently also generalizes Advantage Learning (AL) (Baird III, 1999; Bellemare et al., 2016).

3.2. Dual Averaging MPI

We provide an equivalent formulation of scheme (1). This will be the basis of our analysis, and it also allows us to draw connections to other algorithms, originally not introduced as doing a KL regularization. All the technical details are provided in the Appendix, but we give an intuition here, for the case $\tau = 0$ (no entropy). Let $\pi_{k+1} = \mathcal{G}^{\lambda,0}_{\pi_k}(q_k)$. This optimization problem can be solved analytically, yielding $\pi_{k+1} \propto \pi_k \exp \frac{q_k}{\lambda}$. By direct induction, π_0 being uniform,

$$\pi_{k+1} \propto \pi_k \exp \frac{q_k}{\lambda} \propto \cdots \propto \exp \frac{\sum_{j=0}^{k} q_j}{\lambda}.$$

This means that penalizing the greedy step with a KL divergence provides a policy being a softmax over the scaled sum of all past q-functions. This is reminiscent of dual averaging in convex optimization, hence the name.

We now introduce the Dual Averaging MPI (DA-MPI) scheme. Contrary to MD-MPI, we have to distinguish the cases $\tau = 0$ and $\tau \neq 0$. We consider also type 1 and type 2 variations for evaluation. DA-MPI$_{1\text{-}2}$(λ,0) is given by

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k) \\ q_{k+1} = (T^{1-2}_{\pi_{k+1}|\pi_k})^m q_k + \epsilon_{k+1} \\ h_{k+1} = \frac{k+1}{k+2} h_k + \frac{1}{k+2} q_{k+1} \end{cases}, \qquad (2)$$

with $h_0 = q_0$. For $\tau > 0$, DA-MPI$_{1\text{-}2}$(λ,τ) is given by

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{0,\tau}(h_k) \\ q_{k+1} = (T^{1-2}_{\pi_{k+1}|\pi_k})^m q_k + \epsilon_{k+1} \\ h_{k+1} = \beta h_k + (1-\beta) q_{k+1} \text{ with } \beta = \frac{\lambda}{\lambda+\tau} \end{cases}, \qquad (3)$$

with $h_{-1} = 0$. The following result is proven in Appx. C.1.

Proposition 1. *For type $t \in \{1, 2\}$, for any $\lambda > 0$, MD-MPI$_t$(λ,0) and DA-MPI$_t$(λ,0) are equivalent (but the equivalence does not hold in the limit $\lambda \to 0$). Moreover, for any $\tau > 0$, MD-MPI$_t$(λ,τ) and DA-MPI$_t$(λ,τ) are equivalent.*

Schemes (2) and (3) also encompass a number of approaches (the following statements being justified in Appx. B.2). Politex (Abbasi-Yadkori et al., 2019) is a PI scheme for the average reward case, motivated by the prediction with expert advice problem. In the discounted case, it is DA-MPI$_2$(λ,0). Momentum Value Iteration (MoVI) (Vieillard et al., 2019) is a limit case of DA-MPI$_2$(λ,0) as $\lambda \to 0$, and its practical extension to deep RL momentum DQN is a limit case of DA-MPI$_2$(λ,τ). Speedy Q-learning (SQL) (Azar et al., 2011) is a limit case of DA-MPI$_1$(λ, 0) as $\lambda \to 0$.

4. Theoretical Analysis

Here, we analyze the propagation of errors of MD-MPI, through the equivalent DA-MPI. We provide componentwise bounds that assess the quality of the learned policy, depending on $\tau = 0$ or not and on type $1 - 2$. From these, ℓ_p-norm bounds could be derived (Scherrer et al., 2015, Lemma 5). We also restrict our analysis to a VI scheme, its extension to $m > 1$ remaining an open question.

4.1. Analysis of DA-VI$_1$(λ,0)

This is the following special case of scheme (2):

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k) \\ q_{k+1} = T^{\lambda,0}_{\pi_{k+1}|\pi_k} q_k + \epsilon_{k+1} \\ h_{k+1} = \frac{k+1}{k+2} h_k + \frac{1}{k+2} q_{k+1} \end{cases}. \qquad (4)$$

The following Thm. is proved in Appx. C.2.

Theorem 1. *Define $E_k = -\sum_{j=1}^{k} \epsilon_j$ the sum of iterations' errors, $A_k^1 = (I - \gamma P_{\pi_*})^{-1} - (I - \gamma P_{\pi_k})^{-1}$ and $g^1(k) = \frac{4}{1-\gamma} \frac{v_{max}^{\lambda}}{k}$. Assume that $\|q_k\|_\infty \leq v_{max}$. We have*

$$q_* - q_{\pi_k} \leq \left| A_k^1 \frac{E_k}{k} \right| + g^1(k)\mathbf{1}.$$

Remark 1. *The assumption $\|q_k\|_\infty \leq v_{max}$ is not strong. It would be true without errors, and with errors it can be obtained by clipping the result of the evaluation step in $[-v_{max}, v_{max}]$. See also Appx. C.3.*

We first discuss the error term $\frac{1}{k}E_k$. For example, in a tabular setting with access to a generative model (the setting of Azar et al. (2011)), the sequence of estimation errors is a martingale difference w.r.t. its natural filtration, and the error term $\frac{1}{k}E_k$ vanishes asymptotically, by the law of large numbers, contrary to classic ADP. Even beyond this ideal case, this means that there can be a compensation of errors over iterations (variance reduction). We also have only a linear dependency on the horizon $\frac{1}{1-\gamma}$ for the cumulative error term E_k, contrary to a square dependency for classic ADP, this being tight (Scherrer & Lesner, 2012). We think that these reasons, compensation of errors and associated

[3] That said, we acknowledge that their analysis of MD-MPI could easily be extended to an additional fixed regularizer.

linear dependency to the horizon, could explain the good empirical results of using a KL regularization in RL.

To illustrate the above discussion, we can express an ℓ_∞ bound as a direct corollary of Thm 1:

$$\|q_* - q_{\pi_k}\|_\infty \leq \mathcal{O}\left(\frac{1}{1-\gamma}\left\|\frac{1}{k}\sum_{j=1}^k \epsilon_j\right\|_\infty + \frac{1}{1-\gamma}\frac{v_{\max}^\lambda}{k}\right).$$

The first term is the error term, and the second term essentially expresses how fast an initialization error vanishes. This is to be compared to the classic propagation of errors of Approximate VI (AVI) (*e.g.*, Scherrer et al. (2015)):

$$\|q_* - q_{\pi_k}\|_\infty \leq \mathcal{O}\left(\frac{\max_{j\leq k}\|\epsilon_j\|_\infty}{(1-\gamma)^2} + \frac{1}{1-\gamma}\gamma^k v_{\max}\right).$$

The second term (linked to initialization) vanishes more quickly (γ^k instead of $\frac{1}{k}$). This was to be expected: regularization forces changes of a policy to be small, which is clearly unnecessary when no error is involved. However, regarding the error term, they pay for the worst error among past iterations[4], instead of paying for the average of past errors. Also, this error term is multiplied by a square horizon, instead of a linear one, and it is known that it cannot be improved (Scherrer & Lesner, 2012).

DA-VI$_1$(λ,0) is not the first algorithm that benefits from a compensation of errors; DPP and SQL also have this property. Their bounds are similar, and can be framed as

$$\|q_* - q_{\pi_k}\|_\infty \leq \mathcal{O}\left(\frac{\max_{j\leq k}\|E_j\|_\infty}{(1-\gamma)^2 k} + \frac{1}{(1-\gamma)^2}\frac{v_{\max}^\lambda}{k}\right).$$

We retrieve the compensation of errors, but it suffers from a squared dependency to the horizon, instead of a linear one in our case. Yet, both DPP and SQL being special cases of DA-VI$_1$(λ, 0), our (better) bound should apply to them too.

4.2. Analysis of DA-VI$_1$(λ,τ)

This is the following special case of scheme (3):

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{0,\tau}(h_k) \\ q_{k+1} = T^{\lambda,\tau}_{\pi_{k+1}|\pi_k} q_k + \epsilon_{k+1} \\ h_{k+1} = \beta h_k + (1-\beta) q_{k+1} \text{ with } \beta = \frac{\lambda}{\lambda+\tau} \end{cases}.$$

Due to the entropy term, this scheme cannot converge to the unregularized optimal q-function. Yet, without errors and with $\lambda = 0$, it would converge to the solution of the MDP regularized by the scaled entropy (that is, considering the reward augmented by the scaled entropy). Our bound will show that adding a KL penalty does not change this. To do so, we introduce a few notations. All the following claims come from Geist et al. (2019). We already have defined the operator $T^{0,\tau}_\pi$. It has a unique fixed point, that we write q^τ_π. The unique optimal q-function is $q^\tau_* = \max_\pi q^\tau_\pi$. We write $\pi^\tau_* = \mathcal{G}^{0,\tau}(q^\tau_*)$ the associated unique optimal policy, and $q^\tau_{\pi^\tau_*} = q^\tau_*$. The next result is proven in Appx. C.4.

Theorem 2. *For a sequence of policies $\pi_0, \dots, \pi_k$, we define $P_{k:j} = P_{\pi_k}P_{\pi_{k-1}}\dots P_{\pi_j}$ if $j \leq k$, $P_{k:j} = I$ else. We define $A^2_{k:j} = P^{k-j}_{\pi^\tau_*} + (I-\gamma P_{\pi_{k+1}})^{-1}P_{k:j+1}(I-\gamma P_{\pi_j})$. We define $g^2(k) = \gamma^k(1+\frac{1-\beta}{1-\gamma})\sum_{j=0}^k(\frac{\beta}{\gamma})^j v^\tau_{max}$. Finally, we define $E^\beta_k = (1-\beta)\sum_{j=1}^k \beta^{k-j}\epsilon_j$. With these notations:*

$$q^\tau_* - q^\tau_{\pi_{k+1}} \leq \sum_{j=1}^k \gamma^{k-j}\left|A^2_{k:j}E^\beta_j\right| + g^2(k)\mathbf{1}.$$

This is quite close to the bound of CVI (despite a quite different proof technique). It is not surprising, CVI being a reparametrization of DA-VA$_1$(λ,τ). To ease the discussion, we express an ℓ_∞ bound as a direct corollary of Thm. 2:

$$\|q^\tau_* - q^\tau_{\pi_{k+1}}\|_\infty \leq \mathcal{O}\left(\frac{1}{1-\gamma}\sum_{j=1}^k \gamma^{k-j}\|E^\beta_j\|_\infty + g^2(k)\right).$$

First, we discuss the error term. It is a moving average of the errors made at each iteration, $E^\beta_k = \beta E^\beta_{k-1} + (1-\beta)\epsilon_k$. In the ideal case where the sequence of these errors is a martingale difference with respect to the natural filtration, this term no longer vanishes, contrary to $\frac{1}{k}E_k$ (and the dependency to the horizon is square here, instead of linear before). However, it can reduce the variance. For simplicity, assume that the ϵ_j's are i.i.d. of variance 1. In this case, it is easy to see that the variance of E^β_k is bounded by $\beta\frac{1-\beta}{1+\beta} < 1$, that tends toward 0 for β close to 1. Therefore, we advocate that DA-VI$_1$(λ,τ) allows for a better control of the error term than classic AVI. The bound of CVI has the same error term (up to the normalizing constant), and Kozuno et al. (2019) provide further discussions about it.

Second, we discuss the term $g^2(k)$, which tells how fast the initialization of the algorithm vanishes. If $\beta = \gamma$, we have $g^2(k) = 2(k+1)\gamma^k v^\tau_{\max}$. If $\beta \neq \gamma$, we have that $g^2(k) = (1 + \frac{1-\beta}{1-\gamma})\frac{\beta^{k+1}-\gamma^{k+1}}{\beta-\gamma}$. In all cases, we have that $\lim_{k\to\infty} g(k) = 0$. The asymptotic rate of convergence is always faster than $\mathcal{O}(\frac{1}{k})$ (the rate we have without entropy), but can be slower than $\mathcal{O}(\gamma^k)$ (the rate of AVI).

Third, we discuss the interplay between the KL and the entropy terms. The l.h.s. of the bound of Thm. 2 solely depends on the entropy scale τ, while the r.h.s. solely depends on the term $\beta = \frac{\lambda}{\lambda+\tau}$. With DA-VI$_1$($\lambda$,$\tau$), we approximate the optimal policy of the regularized MDP, while we are usually interested in the solution of the original MDP. We have that (*e.g.*, Geist et al. (2019)) $\|q_* - q_{\pi^\tau_*}\|_\infty \leq \frac{\tau \ln|\mathcal{A}|}{1-\gamma}$.

[4] This could be refined, old errors being less important, but the conclusion would remain the same.

So, this bias can be controlled by setting an (arbitrarily) small τ. This does not affect the r.h.s. of the bound, as long as the scale of the KL term follows (such that $\frac{\lambda}{\lambda+\tau}$ remains fixed to the chosen value). So, Thm. 2 suggests to set τ to a very small value and to choose λ such that we have a given value of β (for example, close to 1 to reduce the variance). Moreover, in the limiting case $\tau, \lambda \to 0$ with fixed β, the bound of DA-VI$_1$(λ,0) is better (both regarding the error and the horizon). However, adding an entropy term has been proven efficient empirically, be it with arguments of exploration and robustness (*e.g.* Haarnoja et al. (2018)) or regarding the optimization landscape (Ahmed et al., 2019). Our analysis does not cover these aspects.

4.3. Analysis of DA-VI$_2$(λ,0)

This is the following special case of scheme (2):

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k) \\ q_{k+1} = T_{\pi_{k+1}} q_k + \epsilon_{k+1} \\ h_{k+1} = \frac{k+1}{k+2} h_k + \frac{1}{k+2} q_{k+1} \end{cases}.$$

The change might appear tiny, compared to DA-VI$_1$(λ,0), as only the evaluation step is modified. Yet, it impacts the theoretical analysis. See Appx. C.5 for the proof.

Theorem 3. *Define E_k as in Thm. 1, and the weighted sum of errors as $\mathcal{E}_{j,k} = -\sum_{i=1}^{j} P_{i+k-j:i+1}(I - \gamma P_{\pi_i})\epsilon_i$, with $P_{i:j}$ as in Thm. 2. Define $A_*^3 = (I - \gamma P_{\pi_*})^{-1}$, $A_k^3 = (I - \gamma P_{\pi_k})^{-1}$ and $g^3(k) = \frac{2}{1-\gamma}\frac{v_{max}^{\lambda}}{k}$. Assume $\|q_k\|_\infty \leq v_{max}$ (see Rk. 1) and $q_0 = 0$ (to simplify the bound). We have*

$$q_* - q_{\pi_k} \leq \left| A_*^3 \frac{E_k}{k} - A_k^3 \frac{1}{k} \sum_{j=1}^{k-1} \gamma^{k-1-j} \mathcal{E}_{j,k-1} \right| + g^3(k)\mathbf{1}.$$

To ease the discussion of this result, we again provide an ℓ_∞ bound as a direct corollary of Thm. 3:

$$\|q_* - q_{\pi_k}\|_\infty \leq \mathcal{O}\left(\frac{\|E_k\|_\infty + \frac{\max_{j \leq k-1} \|\mathcal{E}_{j,k-1}\|_\infty}{1-\gamma} + v_{\text{max}}^{\lambda}}{(1-\gamma)k} \right).$$

The terms $\frac{1}{1-\gamma}\frac{\|E_k\|_\infty}{k}$ and $\frac{1}{1-\gamma}\frac{v_{\text{max}}^{\lambda}}{k}$ are the same as in the bound of DA-VI$_1$(λ,0) and were already discussed in Sec. 4.1. There is also an additional error term, $\frac{1}{(1-\gamma)^2}\frac{\max_{j \leq k-1} \|\mathcal{E}_{j,k-1}\|_\infty}{k}$, that scales with the square of the horizon. The weighted cumulative error $\mathcal{E}_{j,k}$ is the same as the one appearing in the analysis of MoVI. This is not surprising, as MoVI is a limiting case of DA-VI$_2$(λ,0), and our bound generalizes the one of Vieillard et al. (2019).

The term $\mathcal{E}_{j,k}$ is a sum of errors, but weighted by a product of transition kernels $P_{i:j}$. If these kernels were arbitrary, this could further reduce the variance (by averaging the errors over the state-action space, in addition to averaging them over iterations). However, these kernels are not arbitrary, they depend on the error they weight, and this dependency is hard to quantify. However, despite this, this term $\mathcal{E}_{j,k}$ behaves empirically much like E_k, at least in the ideal case of the sequence of errors being a martingale difference with respect to the natural filtration. See Vieillard et al. (2019) for further discussion about this error term.

Thereby, if the bound of DA-VI$_2$(λ,0) shows advantage compared to the one of AVI (because of the variance reduction induced by averaging the errors), it is worse than the one of DA-VI$_1$(λ,0). This can be an artifact of the analysis, we do not know if these bounds are tight (but we suspect the bound of DA-VI$_1$(λ,0) to be close to, see the proof). However, it is worth noticing that all deep RL algorithms we are aware of, that regularize with a KL and/or an entropy (*e.g.*, TRPO or MPO), never consider regularizing the evaluation step. Our analysis suggests that it is worth doing it, so we will compare both approaches empirically (see Sec. 6).

4.4. The issue with DA-VI$_2$(λ,τ)

When regularizing the greediness solely with a KL penalty, one can choose to regularize (type 1) or not (type 2) the evaluation step. The resulting bounds are not the same, but without error both algorithmic schemes converge toward the solution of the original MDP. However, when considering also an entropy penalty in the greedy step, the asymptotic solution is necessarily biased (recall Thm. 2, without error DA-VI$_1$(λ,τ) converges toward π_τ^*). In this case, one cannot ignore safely the regularization of the evaluation operator.

To illustrate this, consider DA-VI$_2$(0,τ), without error. It is equivalent to applying repeatedly the following operator (see Appx. C.6 for the derivation): $q_{k+1} = T_\tau q_k$ with $T_\tau q = r + \gamma P \langle \frac{\exp \frac{q}{\tau}}{\langle \mathbf{1}, \exp \frac{q}{\tau} \rangle}, q \rangle$.This is sometimes called the softmax Bellman operator, as the evaluation operator for the policy being softmax w.r.t. q. This operator is not necessarily a contraction in ℓ_∞-norm, it can have multiple fixed points, and it can be practically unstable (Asadi & Littman, 2017)[5]. The regularized Bellman operator is a key to show convergence, or for studying the propagation of errors. This explains why we do not provide a bound. Yet, we would highlight again that the regularization of the evaluation operator is often ignored in the literature.

While the scheme $q_{k+1} = T_\tau q_k$ is not convergent, Song et al. (2019) have shown that the superior and inferior limits are bounded, the gap between both decaying exponentially fast as $\tau \to 0$ (in the limit, we retrieve the classic Bellman optimality operator). Yet, this analysis is done in the exact case, and it is not enough to study the propagation of errors.

[5] As a side note, the mellowmax policy introduced by Asadi & Littman (2017) to circumvent these issues can be seen as an indirect regularization of the evaluation step, see Appx. C.6.

There is a possible workaround to make the scheme convergent, that consists in introducing a third type of evaluation, without KL regularization but with entropy regularization. It is possible to provide a bound in this case, that mixes the moving average aspect of Thm. 2 with the weighted error of Thm. 3. We think that if one considers regularizing the evaluation, it is worth doing the effort to regularize with both the KL and the entropy, so we postpone the full presentation of the workaround and the related theorem to Appx. C.6.

4.5. Limitation of our analysis

In our analysis, we considered errors in the evaluation step (update of q), but not in the greedy step (update of π) or in the averaging step (update of h). This is a limitation of our analysis. With errors in the greedy step, the equivalence between MD-MPI and DA-MPI would no longer hold. Moreover, adding errors in the averaging step or in the greedy step for DA-VI would change our bounds, with additional error terms maybe not compensating.

With a linear parameterization and discrete actions, it might be reasonable to consider errors only in the evaluation step (the averaging can be done analytically by averaging the parameters, and the policies can be computed analytically as functions of the q-functions). We provide an empirical illustration of the bounds in an ideal case (tabular case, generative model) in Appx. D. However, with neural networks it is no longer the case. Therefore, we provide extensive empirical comparison of MD-VI and DA-VI in Sec. 6. We also think important to compare type 1 and type 2, that is regularizing or not the evaluation step, because doing this regularization is rarely envisioned in the literature, while the analysis suggests that it could be better.

Before this, we explain how to derive practical off-policy deep actor-critics from the abstract MD-VI and DA-VI schemes, and present the variations that we'll consider.

5. Practical algorithms

DA-VI and MD-VI are extensions of VI. One of the most prevalent VI-based deep RL algorithm is probably DQN (Mnih et al., 2015). Thus, our approach consists in modifying the DQN algorithm to study regularization. We present the different variations we consider with a high level viewpoint, all practical details being in Appx. E.1.

DQN maintains a replay buffer and a target network q_k, and computes q_{k+1} by minimizing the loss ('t2' is for type 2):

$$\mathcal{L}_{\text{t2}}(q) = \hat{\mathbb{E}}_{s,a}\left[\left([\hat{T}_{\pi_{k+1}} q_k](s,a) - q(s,a)\right)^2\right], \qquad (5)$$

with q a neural network, $\pi_{k+1} \in \mathcal{G}(q_k)$ the greedy policy computed analytically from q_k, $[\hat{T}_{\pi_{k+1}} q_k](s,a) = r(s,a) + \gamma\langle \pi_{k+1}, q_k\rangle(s')$ the sampled Bellman operator (with $s' \sim P(\cdot|s,a)$), and where the empirical expectation $\hat{\mathbb{E}}_{s,a}$ is according to the transitions in the buffer. DQN is an optimistic AVI scheme, in the sense that only a few steps of stochastic gradient descent are performed before updating the target network. We modify DQN by adding a policy network and possibly modifying the evaluation step. For the moment, we consider $\tau > 0$.

Greedy step. As explained before, when the greedy step is approximated, MD-VI and DA-VI are no longer equivalent. We start with MD-VI. A natural way to learn the policy network is to optimize directly for the greedy step. Let π_k be the target policy network and q_k the target q-network, it corresponds to ('dir' stands for direct):

$$\mathcal{L}_{\text{dir}}(\pi) = \hat{\mathbb{E}}_s\left[\langle \pi, q_k\rangle(s) - \lambda\,\text{KL}(\pi||\pi_k)(s) + \tau\mathcal{H}(\pi)(s)\right].$$

Maximizing this loss over networks gives π_{k+1}. This is reminiscent of TRPO (see Appx. B.1). One can also compute analytically the policy π_{k+1} (see Appx. A), but it would require remembering all past networks. Thus, another solution is to approximate this analytical solution by a neural network ('ind' stands for indirect):

$$\mathcal{L}_{\text{ind}}(\pi) = \hat{\mathbb{E}}_s\left[\text{KL}(\pi^*_{k+1}||\pi)(s)\right] \text{ with } \pi^*_{k+1} \propto \pi_k^\beta \exp\frac{\beta q_k}{\lambda}.$$

Minimizing this loss over networks gives π_{k+1}. This is reminiscent of MPO[6] (see Appx. B.1). When considering DA-VI, the policy can be computed analytically, $\pi_{k+1} = \mathcal{G}^{0,\tau}(h_k)$, but h_k has to be approximated (and can be seen as the logits of the policy). With h_{k-1} and q_k the target networks:

$$\mathcal{L}_{\text{da}}(h) = \hat{\mathbb{E}}_{s,a}\left[([\beta h_{k-1} + (1-\beta)q_k](s,a) - h(s,a))^2\right].$$

Minimizing this loss over networks h gives h_k. This is reminiscent of momentum-DQN (see Appx. B.2).

Evaluation step. Given one of the three ways of doing the greedy step, one can choose between type 1 and type 2 for the evaluation step. Type 2 is already depicted in Eq. (5) (changing the considered policy) and type 1 is given by

$$\mathcal{L}_{\text{t1}}(q) = \hat{\mathbb{E}}_{s,a}\left[\left([\hat{T}^{\lambda,\tau}_{\pi_{k+1}} q_k](s,a) - q(s,a)\right)^2\right].$$

So combining one of the two evaluation steps (type 1 or type 2) with one of the three greedy steps (MD-dir, MD-ind or DA), we get six variations. We discuss also the limit case without entropy.

When $\tau = 0$. For MD-VI, one can set $\tau = 0$. However, recall that for DA-VI, the resulting algorithm is different.

[6] One could also consider the KL in the other sense, which could be interesting for ignoring the partition function.

DA-VI(λ, 0) (see Eq. (4)) is not practical in a deep learning setting, as it requires averaging over iterations. Indeed, updates of target networks are too fast to consider them as new iterations, and a moving average is more convenient. Vieillard et al. (2019) used a decay on β to mimic this behavior, but this is a heuristic that needs to be tuned. Therefore, for DA-VI we will only consider the limit case $\lambda + \tau \to 0$ with $\beta = \frac{\lambda}{\lambda+\tau}$ kept constant (that is, momentum-DQN with fixed β). In this case, type 1 and 2 are equivalent. We offer additional visualisations in Appx. E.3.

6. Experiments

The rationale of the following experimental study is to compare the different ways of doing regularization, which would be equivalent in a linear setting, and to study the effect of regularizing the evaluation operator, something largely ignored in the literature, especially with a KL regularization. To do so, we start from a reasonably tuned version of DQN, from Dopamine (Castro et al., 2018), and modify it to obtain the six variations depicted in Sec. 5, as well as their limit cases. We keep the meta-parameters fixed for all variations (*e.g.*, the optimizer is the same for all networks), the ones working well for DQN. All details are in Appx. E.2. We will only vary the parameters linked to the entropy and the KL, which are our subject of study.

Regarding the parameters λ, τ and $\beta = \frac{\lambda}{\lambda+\tau}$, one is redundant. Our analysis (Sec. 4.2) suggests that τ and β are important. Moreover, the fact that $\beta \in (0, 1)$ is convenient. So, we consider a (τ, β) space: τ scales the entropy, β gives a relative scale between the entropy and the KL. If $\tau = 0$, we vary λ for MD-VI and β for DA-VI (see end of Sec. 5).

6.1. Empirical study

Visualisation. For each considered environment, we present results as a table, the rows corresponding to the type of evaluation, the columns to the kind of greedy step. Each element of this table is a grid, varying β for the rows and τ for the columns. One element of the grid is, for the considered variation and (τ, β) couple, the average undiscounted return per episode obtained during training, also averaged over a number of seeds that depends on the environment. On the bottom of this table, we show the limit cases with the same principle, varying with λ for MD-VI and with β for DA-VI (with only one type of evaluation for DA-VI, as explained end of Sec. 5). The scale of colors is common to all these subplots, and the performance of DQN is indicated on this scale for comparison.

Environments. We provide results for five environments. We consider two light environments from Gym (Brockman et al., 2016), Cartpole (Fig. 1) and Lunar lander (Fig. 2), to allow for a large sweep over the parameters (their values being visible on the figures). For these environment, each cell of each grid is averaged over 10 seeds. We also consider three Atari games (Bellemare et al., 2013), with sticky actions, namely Asterix (Fig. 3), Breakout (Fig. 4) and Seaquest (Fig. 5), to assess the effect of regularization on more challenging problems. The sweep over parameters is smaller (see the figures for the values), and each cell of each grid is averaged over 3 seeds.

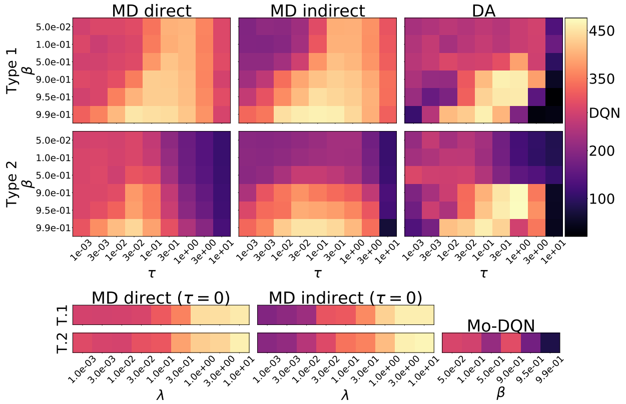

Figure 1. Cartpole.

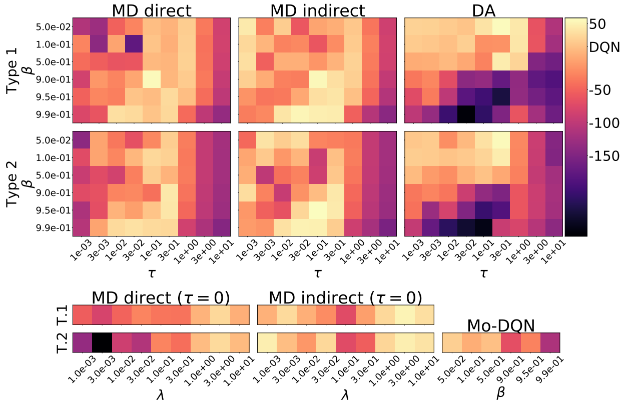

Figure 2. Lunar lander.

6.2. Discussion

Evaluation. We compare the two types of evaluation, with (type 1) or without (type 2) regularization, *i.e.* the rows of our figures. The theory suggests that type 1 should lead to better convergence. On the gym environments, it generally helps, with a larger area of working parameters. On the Atari environments, it is less clear: the type of evaluation has only a small impact on which parameters work best. Still, it never performs worse than type 2. Type 2 is mostly used in the literature, and it shows here to be efficient enough (it is competitive with type 1), but our analysis combined with these experiments would suggest using the more theoretically grounded type 1 instead.

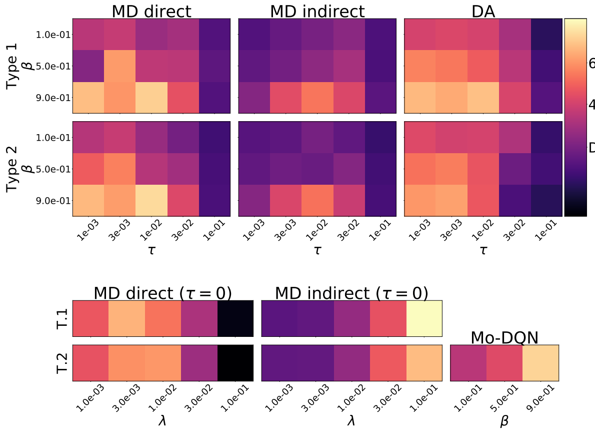

Figure 3. Asterix.

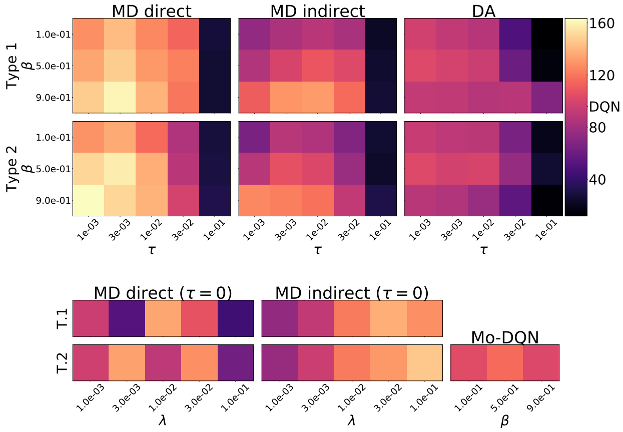

Figure 4. Breakout.

Greediness. The three approaches show different behaviour on the environments, confirming that the equivalence is lost. As expected, the two MD approaches still share a similar behaviour. Choosing the most efficient greedy step is not obvious, as it seems to depend heavily on the environment, but in general, MD-dir benefits from a lower sensitivity to parameters and higher maximal performance.

On β and τ. All the results suggest a similar choice of parameters: a value of β close to 1, and a not-too-small value for τ. This is consistent with the theory for β, as commented in Sec. 4.2. The optimal value for τ would be 0, but we see that it still helps on some environments (both Gyms, and Seaquest). This could be explained by the other effects of regularization mentioned in the introduction, such as smoothing the optimization landscape or enhancing exploration, while a too large value of τ can kill performance by producing too random policies. However, the specific values are dependent on the environment, and for example the effect of τ on Atari is not that clear. This could be caused by the different magnitudes of q-functions.

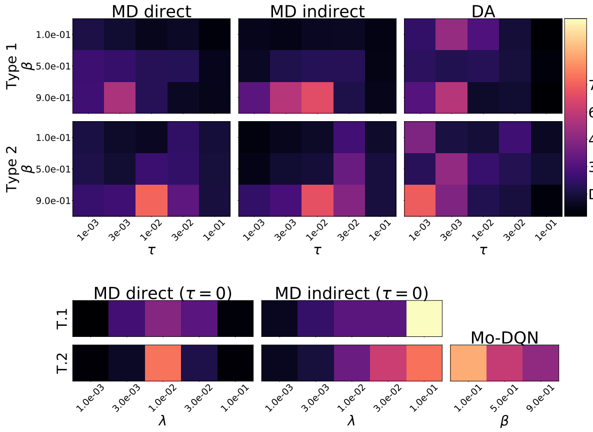

Figure 5. Seaquest.

Behavior of limit cases. On the gym environments, the two MD approaches clearly benefit from a regularization. Although the theory would suggest to choose a λ close to 0, it appears beneficial to use a higher value when using stochastic approximation. On Atari, the results are more surprising, as MD-dir and MD-ind do not share the same behaviour w.r.t. λ. Indeed, MD-dir exhibits a somewhat chaotic behaviour, even opposite to MD-ind on Asterix. We acknowledge this could simply be an artifact of the variance between random seeds, but it could also hide some yet unexplained effect. The DA limit case (Mo-DQN) was already tested on Atari by Vieillard et al. (2019), and with the additional results on Gym, we can draw a similar conclusion, that is that the optimal value of β is game-dependent.

7. Conclusion

In this paper, we provided an explanation of the effect of regularization in RL. We conducted this study through the lens of regularized ADP, a framework that encompasses a number of recent and successful approaches making use of regularization. We showed an equivalence between regularizing the greedy step with a KL divergence and averaging the successive q-values. With this equivalence, we were able to prove that this type of regularization averages the errors made when approximating the evaluation step. Although regularization can have other effects not covered in this work, the compensation of errors is a newly described phenomenon that can explain why it is efficient in practice. To complete this theoretical analysis, we conducted an intensive empirical study, comparing different deep actor-critics inspired by the ADP framework. The results of this study allowed us to observe to what extent our analysis stood in the absence of several assumptions, and also showed that more work is required to understand precisely regularization in a practical setting. These results also confirmed that regularization can improve significantly vanilla DQN.

References

Abbasi-Yadkori, Y., Bartlett, P., Bhatia, K., Lazic, N., Szepesvári, C., and Weisz, G. Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning (ICML)*, 2019.

Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations (ICLR)*, 2018.

Ahmed, Z., Le Roux, N., Norouzi, M., and Schuurmans, D. Understanding the impact of entropy on policy optimization. In *International Conference on Machine Learning (ICML)*, 2019.

Archibald, T., McKinnon, K., and Thomas, L. On the generation of markov decision processes. *Journal of the Operational Research Society*, 46(3):354–361, 1995.

Asadi, K. and Littman, M. L. An alternative softmax operator for reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2017.

Azar, M. G., Munos, R., Ghavamzadeh, M., and Kappen, H. J. Speedy q-learning. In *Advances in neural information processing systems (NIPS)*, 2011.

Azar, M. G., Gómez, V., and Kappen, H. J. Dynamic policy programming. *Journal of Machine Learning Research (JMLR)*, 13(Nov):3207–3245, 2012.

Baird III, L. C. *Reinforcement Learning Through Gradient Descent*. PhD thesis, US Air Force Academy, US, 1999.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Bellemare, M. G., Ostrovski, G., Guez, A., Thomas, P. S., and Munos, R. Increasing the action gap: New operators for reinforcement learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. 2016.

Castro, P. S., Moitra, S., Gelada, C., Kumar, S., and Bellemare, M. G. Dopamine: A research framework for deep reinforcement learning. *arXiv preprint arXiv:1812.06110*, 2018.

Fox, R., Pakman, A., and Tishby, N. Taming the noise in reinforcement learning via soft updates. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2016.

Geist, M., Scherrer, B., and Pietquin, O. A theory of regularized markov decision processes. In *International Conference on Machine Learning (ICML)*, 2019.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning (ICML)*, 2017.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, 2018.

Hiriart-Urruty, J.-B. and Lemaréchal, C. *Fundamentals of convex analysis*. Springer Science & Business Media, 2012.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2002.

Kozuno, T., Uchibe, E., and Doya, K. Theoretical analysis of efficiency and robustness of softmax and gap-increasing operators in reinforcement learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

McMahan, H. B. A unified view of regularized dual averaging and mirror descent with implicit updates. *arXiv preprint arXiv:1009.3240*, 2010.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529, 2015.

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

Puterman, M. L. and Shin, M. C. Modified policy iteration algorithms for discounted markov decision problems. *Management Science*, 24(11):1127–1137, 1978.

Scherrer, B. and Lesner, B. On the use of non-stationary policies for stationary infinite-horizon markov decision processes. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

Scherrer, B., Ghavamzadeh, M., Gabillon, V., Lesner, B., and Geist, M. Approximate modified policy iteration and its application to the game of tetris. *Journal of Machine Learning Research (JMLR)*, 16:1629–1676, 2015.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International Conference on Machine Learning (ICML)*, 2015.

Schulman, J., Chen, X., and Abbeel, P. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017.

Song, Z., Parr, R., and Carin, L. Revisiting the softmax bellman operator: New benefits and new perspective. In *International Conference on Machine Learning (ICML)*, 2019.

Vieillard, N., Scherrer, B., Pietquin, O., and Geist, M. Momentum in reinforcement learning. *arXiv preprint arXiv:1910.09322*, 2019.

Content. This Appendix provides additionnal details. Appx. B justifies the connections drawn in Sec. 3 between MD-MPI or DA-MPI and the literature. Appx. C provides the proofs of all stated theoretical results, as well as some necessary lemmata. Appx. D illustrates the bounds empirically, in an ideal case (tabular case, generative model). Appx. E provides additional details regarding the practical algorithms and the experiments. Before that, as a warm-up, we state in Appx. A a few facts about the Legendre-Fenchel transform that will be useful all along the derivations.

A. Convex Conjugacy for KL and Entropy Regularization

Let $q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ and $\mu \in \Delta_{\mathcal{A}}^{\mathcal{S}}$, and consider the general greedy step $\pi' \in \mathcal{G}_{\mu}^{\lambda,\tau}$, the optimization being understood here state-wise.

$$\pi' \in \operatorname*{argmax}_{\pi\in\Delta_{\mathcal{A}}^{\mathcal{S}}} \left(\langle \pi, q\rangle - \lambda\,\mathrm{KL}(\pi||\mu) + \tau\mathcal{H}(\pi)\right). \tag{6}$$

The function $\lambda\,\mathrm{KL}(\pi||\mu) - \tau\mathcal{H}(\pi)$ being convex in π, this optimization problem is related to the Legendre-Fenchel transform (*e.g.*, Hiriart-Urruty & Lemaréchal (2012, Ch. E)), or convex conjugate (which is the maximum rather than the maximizer). First, we consider a simple case, $\lambda = 0$ and $\tau = 1$. It is well known in this case that the maximum (the convex conjugate) is the log-sum-exp function and the maximizer (the gradient of the convex conjugate) is the softmax (*e.g.*, Boyd & Vandenberghe (2004, Ex. 3.25)):

$$\max_{\pi\in\Delta_{\mathcal{A}}^{\mathcal{S}}} \left(\langle \pi, q\rangle + \mathcal{H}(\pi)\right) = \ln\langle \mathbf{1}, \exp q\rangle \in \mathbb{R}^{\mathcal{S}},$$
$$\operatorname*{argmax}_{\pi\in\Delta_{\mathcal{A}}^{\mathcal{S}}} \left(\langle \pi, q\rangle + \mathcal{H}(\pi)\right) = \frac{\exp q}{\langle \mathbf{1}, \exp q\rangle} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}},$$

with $\mathbf{1} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ the vector of which all components are equal to 1. We made use of the notations introduced in Sec. 2, and overload $v \in \mathbb{R}^{\mathcal{S}}$ to $v \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ as $v(s,a) = v(s)$. To make things clear, it gives

$$\left[\ln\langle \mathbf{1}, \exp q\rangle\right](s) = \ln \sum_{a\in\mathcal{A}} \exp q(s,a)$$
$$\text{and } \left[\frac{\exp q}{\langle \mathbf{1}, \exp q\rangle}\right](s,a) = \frac{\exp q(s,a)}{\sum_{a'\in\mathcal{A}} q(s,a')}.$$

Notice also that a direct consequence of this is that

$$\ln\langle \mathbf{1}, \exp q\rangle = \langle \pi', q\rangle + \mathcal{H}(\pi') \text{ with } \pi' = \frac{\exp q}{\langle \mathbf{1}, \exp q\rangle}.$$

From this simple case, we can easily handle the general case. We have

$$\begin{aligned}\langle \pi, q\rangle - \lambda\,\mathrm{KL}(\pi||\mu) + \tau\mathcal{H}(\pi) &= \langle \pi, q\rangle - \lambda\langle \pi, \ln\pi - \ln\mu\rangle - \tau\langle \pi, \ln\pi\rangle \\ &= \langle \pi, q + \lambda\ln\mu\rangle - (\lambda+\tau)\langle \pi, \ln\pi\rangle \\ &= (\lambda+\tau)\left(\left\langle \pi, \frac{q+\lambda\ln\mu}{\lambda+\tau}\right\rangle + \mathcal{H}(\pi)\right).\end{aligned}$$

From this, we can deduce directly that the maximum of (6) is

$$\begin{aligned}\max_{\pi\in\Delta_{\mathcal{A}}^{\mathcal{S}}} \left(\langle \pi, q\rangle - \lambda\,\mathrm{KL}(\pi||\mu) + \tau\mathcal{H}(\pi)\right) &= (\lambda+\tau)\ln\left\langle 1, \exp\frac{q+\lambda\ln\mu}{\lambda+\tau}\right\rangle = (\lambda+\tau)\ln\left\langle \mu^{\frac{\lambda}{\lambda+\tau}}, \exp\frac{q}{\lambda+\tau}\right\rangle \\ &= (\lambda+\tau)\left(\ln\sum_{a\in\mathcal{A}} \mu(a|s)^{\frac{\lambda}{\lambda+\tau}} \exp\frac{q(s,a)}{\lambda+\tau}\right)_{s\in\mathcal{S}},\end{aligned} \tag{7}$$

and that the maximizer of (6) is

$$\begin{aligned}\operatorname*{argmax}_{\pi\in\Delta_{\mathcal{A}}^{\mathcal{S}}} \left(\langle \pi, q\rangle - \lambda\,\mathrm{KL}(\pi||\mu) + \tau\mathcal{H}(\pi)\right) &= \frac{\exp\frac{q+\lambda\ln\mu}{\lambda+\tau}}{\langle \mathbf{1}, \exp\frac{q+\lambda\ln\mu}{\lambda+\tau}\rangle} = \frac{\mu^{\frac{\lambda}{\lambda+\tau}}\exp\frac{q}{\lambda+\tau}}{\langle \mathbf{1}, \mu^{\frac{\lambda}{\lambda+\tau}}\exp\frac{q}{\lambda+\tau}\rangle} \\ &= \left(\frac{\mu(a|s)^{\frac{\lambda}{\lambda+\tau}}\exp\frac{q(s,a)}{\lambda+\tau}}{\sum_{a'\in\mathcal{A}} \mu(a'|s)^{\frac{\lambda}{\lambda+\tau}}\exp\frac{q(s,a')}{\lambda+\tau}}\right)_{(s,a)\in\mathcal{S}\times\mathcal{A}}\end{aligned} \tag{8}$$

Again, the relationship between the maximum and the maximizer gives

$$(\lambda+\tau)\ln\left\langle \mu^{\frac{\lambda}{\lambda+\tau}}, \exp\frac{q}{\lambda+\tau}\right\rangle = \langle \pi', q\rangle - \lambda\,\mathrm{KL}(\pi'||\mu) + \tau\mathcal{H}(\pi') \text{ with } \pi' = \frac{\mu^{\frac{\lambda}{\lambda+\tau}}\exp\frac{q}{\lambda+\tau}}{\langle \mathbf{1}, \mu^{\frac{\lambda}{\lambda+\tau}}\exp\frac{q}{\lambda+\tau}\rangle}. \tag{9}$$

B. Connections to existing algorithms

In this section, we justify the connections stated in Sec. 3 between the considered regularized DP schemes and the literature.

B.1. Connection of MD-MPI$_{1\text{-}2}$(λ,τ) to other algorithms

Connection to SAC. We stated that SAC (Haarnoja et al., 2018) is a variation of MD-MPI$_1$(0,τ). SAC was introduced as PI scheme ($m=\infty$), while it is practically implemented as VI scheme ($m=1$). We keep the VI viewpoint for this discussion. The MD-VI$_1$(0,τ) scheme is given by

$$\begin{cases}\pi_{k+1} = \mathcal{G}^{0,\tau}(q_k)\\ q_{k+1} = T^{0,\tau}_{\pi_{k+1}}q_k + \epsilon_{k+1}\end{cases}. \tag{10}$$

The regularized Bellamn operator can be rewritten as follows:

$$T^{0,\tau}_{\pi_{k+1}}q_k = T_{\pi_{k+1}}q_k + \gamma P\tau\mathcal{H}(\pi_{k+1}) = r + \gamma P\left(\langle \pi_{k+1}, q_k\rangle - \tau\langle \pi_{k+1}, \ln\pi_{k+1}\rangle\right) = r + \gamma P\langle \pi_{k+1}, q_k - \tau\ln\pi_{k+1}\rangle.$$

This is exactly the Bellman operator considered in SAC. For the greedy step, we have directly from Eq. (8) that $\pi_{k+1}\propto \exp\frac{q_k}{\tau}$. In SAC, continuous actions are considered, so the policy cannot be computed (due to the partition function). Therefore, it is approximated with a neural network by minimizing a reverse KL divergence (that allows getting rid of the partition function) between the neural policy and the target policy (the solution of the original greedy step):

$$\pi_{k+1} = \operatorname*{argmin}_{\pi_\theta}\mathbb{E}_s[\mathrm{KL}(\pi_\theta||\pi^*_{k+1})] = \operatorname*{argmin}_{\pi_\theta}\mathbb{E}_s[\mathrm{KL}(\pi_\theta||\exp\frac{q}{\tau})] \text{ with } \pi^*_{k+1} = \frac{\exp\frac{q_k}{\tau}}{\langle \mathbf{1}, \exp\frac{q_k}{\tau}\rangle}.$$

Connection to Soft Q-learning. We stated that Soft Q-learning (Fox et al., 2016; Haarnoja et al., 2017) is also a variation of MD-MPI$_1$(0,τ). It is indeed a VI scheme, so a variation of MD-VI$_1$(0,τ) depicted in Eq. (10). As a direct consequence of Eq. (9), $\pi_{k+1}\propto\exp\frac{q_k}{\tau}$ being the maximizer, we have

$$\langle \pi_{k+1}, q_k\rangle + \tau\mathcal{H}(\pi_{k+1}) = \tau\ln\langle \mathbf{1}, \exp\frac{q_k}{\tau}\rangle.$$

This allows rewriting the evaluation step as follows:

$$\begin{aligned} q_{k+1} &= T^{0,\tau}_{\pi_{k+1}}q_k + \epsilon_{k+1}\\ &= r + \gamma P\left(\langle \pi_{k+1}, q_k\rangle + \tau\mathcal{H}(\pi_{k+1})\right) + \epsilon_{k+1}\\ \Leftrightarrow q_{k+1} &= r + \gamma P\left(\tau\ln\langle \mathbf{1}, \exp\frac{q_k}{\tau}\rangle\right) + \epsilon_{k+1}. \end{aligned} \tag{11}$$

Eq. (11) is equivalent to Eq. (10), and it is the Bellman operator upon which Soft Q-learning is built (replacing the hard maximum by the log-sum-exp). Haarnoja et al. (2017) additionally handle continuous actions, which requires some refinements.

Connection to Softmax DQN. We stated that Softmax DQN (Song et al., 2019) is a variation of MD-MPI$_2$(0,τ). More precisely, it is an MD-VI$_2$(0,τ) scheme:

$$\begin{cases}\pi_{k+1} = \mathcal{G}^{0,\tau}(q_k)\\ q_{k+1} = T_{\pi_{k+1}}q_k + \epsilon_{k+1}\end{cases}.$$

Given that $\pi_{k+1}\propto\exp\frac{q_k}{\tau}$, this amounts to iterating the following so called softmax operator

$$\begin{aligned} q_{k+1} &= T_{\pi_{k+1}}q_k + \epsilon_{k+1}\\ &= r + \gamma P\left\langle \frac{\exp\frac{q_k}{\tau}}{\langle \mathbf{1}, \exp\frac{q_k}{\tau}\rangle}, q_k\right\rangle + \epsilon_{k+1}, \end{aligned}$$

which is the core update rule of softmax DQN. Notice that this operator might not be a contraction (depending on the value of τ), and that it can have multiple fixed points (Asadi & Littman, 2017).

Connection to TRPO. We stated that TRPO (Schulman et al., 2015) is a variation of MD-MPI$_2$(λ, 0). More precisely, it is a variation of MD-PI$_2$(λ, 0):

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{\lambda,0}(q_k) \\ q_{k+1} = T^{\infty}_{\pi_{k+1}} q_k + \epsilon_k = q_{\pi_{k+1}} + \epsilon_k \end{cases} . \tag{12}$$

In TRPO, the q-function is evaluated using Monte Carlo rollouts. The greedy policy is approximated with a neural network by directly solving the expected greedy step:

$$\pi_{k+1} = \operatorname*{argmin}_{\pi_\theta} \mathbb{E}_s[\langle \pi_\theta, q_k \rangle - \lambda \operatorname{KL}(\pi_\theta || \pi_k)].$$

TRPO is indeed a bit different, as it uses importance sampling to sample actions according to π_k (which is especially useful for continuous actions, but does not change the objective function), it uses a constraint based on the KL rather than a regularization, and it considers the KL in the other direction:

$$\pi_{k+1} = \operatorname*{argmin}_{\pi_\theta : \mathbb{E}_s[\operatorname{KL}(\pi_k || \pi_\theta)] \leq \epsilon} \mathbb{E}_s[\mathbb{E}_{a \sim \pi_k(.|s)}[\langle \frac{\pi_\theta}{\pi_k}, q_k \rangle]].$$

However, from an abstract viewpoint, TRPO is close to scheme (12).

Connection to MPO. We stated that MPO (Abdolmaleki et al., 2018) is also a variation of MD-MPI$_2$(λ, 0):

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{\lambda,0}(q_k) \\ q_{k+1} = T^{m}_{\pi_{k+1}} q_k + \epsilon_k \end{cases} . \tag{13}$$

The evaluation step is done by combining a TD approach with eligibility traces (a geometric average of m-step returns), rather than using m-step returns (that amounts to using the T^m_π operator). For the greedy step, the analytic solution can be computed for any state-action couple, and generalized to the whole state-action space by minimizing a KL between this analytical solution and a neural network:

$$\pi_{k+1} = \operatorname*{argmin}_{\pi_\theta} \mathbb{E}_s[\operatorname{KL}(\pi^*_{k+1} || \pi_\theta)] = \operatorname*{argmax}_{\pi_\theta} \mathbb{E}_s[\mathbb{E}_{a \sim \pi^*_{k+1}(.|s)}[\ln \pi_\theta(a|s)]] \text{ with } \pi^*_{k+1} = \frac{\pi_k \exp \frac{q_k}{\lambda}}{\langle \mathbf{1}, \pi_k \exp \frac{q_k}{\lambda} \rangle}.$$

The greedy step of MPO is indeed a bit different, the algorithm being derived from an expectation-maximization principle based on a probabilistic inference view of RL. The term λ is not fixed but learnt by the minimization of a convex dual function (coming from viewing the KL term as a constraint rather than a regularization), and an additional KL penalty is added (not necessarily redundant with the initial one, as the KL there is in the other direction):

$$\pi_{k+1} = \operatorname*{argmax}_{\pi_\theta : \mathbb{E}_s[\operatorname{KL}(\pi_k || \pi_\theta)] \leq \epsilon} \mathbb{E}_s[\mathbb{E}_{a \sim \pi^*_{k+1}(.|s)}[\ln \pi_\theta(a|s)]].$$

However, from an abstract viewpoint, MPO is close to scheme (13).

Connection to DPP. We stated that DPP (Azar et al., 2012) is a variation of MD-MPI$_1$(λ, 0). More precisely, it is close to be a reparameterization of MD-VI$_1$(λ, 0), the difference being mainly the error term:

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{\lambda,0}(q_k) \\ q_{k+1} = T^{\lambda,0}_{\pi_{k+1}} q_k + \epsilon_k \end{cases} . \tag{14}$$

To derive the DPP update rule from Eq. (14), we consider $\epsilon_k = 0$. The greedy policy is, according to (8),

$$\pi_{k+1} = \frac{\pi_k \exp \frac{q_k}{\lambda}}{\langle \mathbf{1}, \pi_k \exp \frac{q_k}{\lambda} \rangle}.$$

Define v_{k+1} as (the second equality coming from Eq. (9))

$$v_{k+1} = \langle \pi_{k+1}, q_k \rangle - \lambda \operatorname{KL}(\pi_{k+1} || \pi_k) = \lambda \ln \langle \pi_k, \exp \frac{q_k}{\lambda} \rangle.$$

With this, we have

$$q_{k+1} = T^{\lambda,0}_{\pi_{k+1}} q_k = r + \gamma P(\langle \pi_{k+1}, q_k\rangle - \lambda \operatorname{KL}(\pi_{k+1}||\pi_k)) = r + \gamma P v_{k+1}$$

Let us define $\psi_{k+1} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ as

$$\psi_{k+1} = \lambda \ln\left(\pi_k \exp \frac{q_k}{\lambda}\right) = r + \gamma P v_k + \lambda \ln \pi_k. \tag{15}$$

Thus, we have

$$\pi_k = \frac{\exp \frac{\psi_k}{\lambda}}{\langle \mathbf{1}, \exp \frac{\psi_k}{\lambda}\rangle} \tag{16}$$

$$\text{and } v_k = \lambda \ln\langle \mathbf{1}, \frac{\psi_k}{\lambda}\rangle. \tag{17}$$

Injecting Eqs. (16) and (17) into (15), we get

$$\psi_{k+1} = r + \gamma P \lambda \ln\langle \mathbf{1}, \frac{\psi_k}{\lambda}\rangle + \psi_k - \lambda \ln\langle \mathbf{1}, \frac{\psi_k}{\lambda}\rangle.$$

This is how DPP is justified from a DP viewpoint (Azar et al., 2012, Appx. A). It is a bit different from the DPP algorithm analyzed by Azar et al. (2012), for which $\ln\langle \mathbf{1}, \frac{\psi_k}{\lambda}\rangle$ is replaced by $\langle \pi_k, \psi_k\rangle$ (both terms being equal in the limit $\lambda \to 0$), and that consider an estimation error ϵ'_{k+1}:

$$\psi_{k+1} = r + \gamma P\langle \pi_k, \psi_k\rangle + \psi_k - \langle \pi_k, \psi_k\rangle + \epsilon'_{k+1}.$$

We advocate that the error ϵ'_k is usually harder to control than ϵ_k (or equivalently that q_k is easier to estimate than ψ_k), because the function ψ_* (the optimal ψ-function for the MDP) is equal to $-\infty$ for any suboptimal action (Azar et al., 2012, Cor. 4).

Connection to CVI. We stated that CVI is a reparametrization of MD-VI$_1$(λ,τ), that we recall (without the error term, to do the reparameterization):

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{\lambda,\tau}(q_k) \\ q_{k+1} = T^{\lambda,\tau}_{\pi_{k+1}} q_k \end{cases}.$$

We now show how to derive the CVI update rule from this. The regularized greedy policy is, thanks to Eq. (8), and writing $\beta = \frac{\lambda}{\lambda+\tau}$:

$$\pi_{k+1} = \frac{\pi_k^\beta \exp \frac{\beta q_k}{\lambda}}{\langle \mathbf{1}, \pi_k^\beta \exp \frac{\beta q_k}{\lambda}\rangle}.$$

Similarly to DPP, we can define v_{k+1} as (still using Eq. (9) for the second equality):

$$v_{k+1} = \langle \pi_{k+1}, q_k\rangle - \lambda \operatorname{KL}(\pi_{k+1}||\pi_k) + \tau\mathcal{H}(\pi_{k+1}) = \frac{\lambda}{\beta} \ln\langle \pi_k^\beta, \exp \frac{\beta q_k}{\lambda}\rangle.$$

With this, we have

$$q_{k+1} = T^{\lambda,0}_{\pi_{k+1}} q_k = r + \gamma P(\langle \pi_{k+1}, q_k\rangle - \lambda \operatorname{KL}(\pi_{k+1}||\pi_k) + \tau\mathcal{H}(\pi_{k+1})) = r + \gamma P v_{k+1}$$

Let us define $\psi_{k+1} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ as

$$\psi_{k+1} = \frac{\lambda}{\beta} \ln\left(\pi_k^\beta \exp \frac{\beta q_k}{\lambda}\right) = r + \gamma P v_k + \lambda \ln \pi_k. \tag{18}$$

Thus, we have

$$\pi_k = \frac{\exp \frac{\beta\psi_k}{\lambda}}{\langle \mathbf{1}, \exp \frac{\beta\psi_k}{\lambda}\rangle} \tag{19}$$

$$\text{and } v_k = \frac{\lambda}{\beta} \ln\langle \mathbf{1}, \frac{\beta\psi_k}{\lambda}\rangle. \tag{20}$$

Injecting Eqs. (19) and (20) into (18), we get

$$\psi_{k+1} = r + \gamma P \frac{\lambda}{\beta} \ln\langle \mathbf{1}, \frac{\beta\psi_k}{\lambda}\rangle + \beta(\psi_k - \frac{\lambda}{\beta}\ln\langle \mathbf{1}, \frac{\beta\psi_k}{\lambda}\rangle).$$

This is exactly the CVI update rule. Notice that setting $\beta = 1$, i.e., $\tau = 0$ (no entropy term), we retrieve DPP (which was to be expected). As we obtain CVI, by considering $\lambda + \tau \to 0$ while keeping $\beta = \frac{\lambda}{\lambda+\tau}$ constant, we retrieve advantage learning in the limit (Baird III, 1999; Bellemare et al., 2016), that DA-VI$_1$(λ,τ) thus generalizes.

B.2. Connection of DA-MPI$_{1\text{-}2}$(λ,τ) to other algorithms

Connection to Politex. Politex (Abbasi-Yadkori et al., 2019) addresses the average reward criterion. It is a PI scheme, up to the fact that the policy, instead of being greedy according to the last q-function, is softmax according to the sum of all past q-function. In the discounted reward case considered here, this is exactly DA-PI$_2$(λ,0):

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k) \\ q_{k+1} = T^\infty_{\pi_{k+1}} q_k + \epsilon_{k+1} = q_{\pi_{k+1}} + \epsilon_{k+1} \\ h_{k+1} = \frac{k+1}{k+2} h_k + \frac{1}{k+2} q_{k+1} \end{cases}$$

Indeed, by definition $h_k = \frac{1}{k+1}\sum_{j=0}^k q_j$ and the greedy policy is

$$\pi_{k+1} = \mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k) = \frac{\exp\frac{(k+1)h_k}{\lambda}}{\langle \mathbf{1}, \exp\frac{(k+1)h_k}{\lambda}\rangle} = \frac{\exp\frac{\sum_{j=0}^k q_j}{\lambda}}{\langle \mathbf{1}, \exp\frac{\sum_{j=0}^k q_j}{\lambda}\rangle}.$$

This is exactly the Politex algorithm, but for the discounted reward case (that changes how the q-function is defined, and thus estimated).

Connection to MoVI. MoVI (Vieillard et al., 2019) is a VI scheme, up to the fact that the policy, instead of being greedy according to the last q-function, is greedy according to the average of past q-functions. It is indeed is a limiting case of DA-VI$_2$(λ, 0), that we recall:

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k) \\ q_{k+1} = T_{\pi_{k+1}} q_k + \epsilon_{k+1} \\ h_{k+1} = \frac{k+1}{k+2} h_k + \frac{1}{k+2} q_{k+1} \end{cases}.$$

It is well known that the limit of a softmax, when the temperatures goes to zero, is the greedy policy: $\mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k) \to \mathcal{G}(h_k)$ as $\lambda \to 0$. So, DA-VI$_2$($\lambda \to 0$, 0) is the following scheme,

$$\begin{cases} \pi_{k+1} = \mathcal{G}(h_k) \\ q_{k+1} = T_{\pi_{k+1}} q_k + \epsilon_{k+1} \\ h_{k+1} = \frac{k+1}{k+2} h_k + \frac{1}{k+2} q_{k+1} \end{cases},$$

that is exactly MoVI. Notice that it is different from MD-VI$_2$($\lambda \to 0$, 0), which is AVI (see also Prop. 1).

Connection to momentum DQN. Momentum DQN (Vieillard et al., 2019) was introduced as a practical heuristic to MoVI, changing the exact average by a moving average (more amenable to optimization with deep networks). We show below that it is indeed a limiting case of DA-VI$_2$(λ,τ), which we recall:

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{0,\tau}(h_k) \\ q_{k+1} = T_{\pi_{k+1}} q_k + \epsilon_{k+1} \\ h_{k+1} = \beta h_k + (1-\beta) q_{k+1} \text{ with } \beta = \frac{\lambda}{\lambda+\tau} \end{cases}.$$

Fix $\beta \in (0, 1)$, we can consider $\lambda, \tau \to 0$ with $\beta = \frac{\lambda}{\lambda+\tau}$ kept constant. In this case, the regularized greedy operator tends to the usual greedy one: $\mathcal{G}^{0,\tau}(h_k) \to \mathcal{G}(h_k)$ as $\tau \to 0$. In the limit, we obtain the following scheme,

$$\begin{cases} \pi_{k+1} = \mathcal{G}(h_k) \\ q_{k+1} = T_{\pi_{k+1}} q_k + \epsilon_{k+1} \\ h_{k+1} = \beta h_k + (1-\beta) q_{k+1} \end{cases},$$

for a chosen β, which is exactly momentum DQN with fixed β.

Connection to Speedy Q-learning. We stated that Speedy Q-learning (Azar et al., 2011) is a limiting case of DA-VI$_1$(λ,0), which we recall (without the error term here):

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k) \\ q_{k+1} = T^{\lambda,0}_{\pi_{k+1}} q_k \\ h_{k+1} = \frac{k+1}{k+2} h_k + \frac{1}{k+2} q_{k+1} \end{cases}.$$

As shown in Lemma 2 in Appx. C.2, we have

$$T^{\lambda,0}_{\pi_{k+1}|\pi_k} q_k = (k+1) T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k - k T^{0,\frac{\lambda}{k}}_{\pi_k} h_{k-1}.$$

With this, DA-VI$_1$(λ,0) can be expressed solely in terms of h_k and π_k:

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k) \\ h_{k+1} = \frac{k+1}{k+2} h_k + \frac{1}{k+2} \left((k+1) T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k - k T^{0,\frac{\lambda}{k}}_{\pi_k} h_{k-1} \right) \end{cases}. \tag{21}$$

As before, as $\lambda \to 0$, the regularized greedy step tends to the greedy step, $\mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k) \to \mathcal{G}(h_k)$. Regarding the evaluation step, we can write, by definition of the regularized Bellman operator and using Eq. (9),

$$\begin{aligned} T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k &= r + \gamma P \left(\langle \pi_{k+1}, h_k \rangle + \frac{\lambda}{k+1} \mathcal{H}(\pi_{k+1}) \right) \\ &= r + \gamma P \left(\frac{\lambda}{k+1} \ln \langle \mathbf{1}, \exp \frac{(k+1) h_k}{\lambda} \rangle \right). \end{aligned}$$

It is a classical result that the convex conjugate of the entropy tends to the hard maximum as the associated temperature goes to zero. For any $s \in \mathcal{S}$,

$$\lim_{\lambda \to 0} \frac{\lambda}{k+1} \ln \sum_{a \in \mathcal{A}} \exp \frac{(k+1) h_k(s,a)}{\lambda} = \frac{1}{k+1} \max_{a \in \mathcal{A}} ((k+1) h_k(s,a)) = \max_{a \in \mathcal{A}} h_k(s,a).$$

Writing T_* the Bellman optimality operator, defined as $T_* q = \max_\pi T_\pi q$, we thus have

$$\lim_{\lambda \to 0} T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k = T_* h_k.$$

Thus, writing the limit of scheme (21) as $\lambda \to 0$, we obtain

$$\begin{cases} \pi_{k+1} = \mathcal{G}(h_k) \\ h_{k+1} = (1 - \frac{1}{k+2}) h_k + \frac{1}{k+2} \left((k+1) T_* h_k - k T_* h_{k-1} \right), \end{cases}$$

which is exactly the Speedy Q-learning update rule.

C. Proofs of Theoretical Results

In this section, we prove the results stated in the paper.

C.1. Proof of Proposition 1

We start by proving the equivalence for the case $\tau = 0$. Recall that we assumed, with little loss of generality, that π_0 is the uniform policy. We recall MD-MPI$_{1\text{-}2}$(λ,0):

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{\lambda,0}_{\pi_k}(q_k) \\ q_{k+1} = (T^{1-2}_{\pi_{k+1}|\pi_k})^m q_k + \epsilon_{k+1} \end{cases}. \tag{22}$$

Let us define $h_0 = q_0$ and h_k for $k \geq 1$ as the average of past q-functions.

$$h_k = \frac{1}{k+1}\sum_{j=0}^{k} q_j = \frac{k}{k+1}h_{k-1} + \frac{1}{k+1}q_k.$$

As a direct consequence of Eq. (8), we have that $\pi_{k+1} \propto \pi_k \exp \frac{q_k}{\lambda}$. By direct induction, we obtain

$$\pi_{k+1} \propto \pi_k \exp \frac{q_k}{\lambda} \propto \pi_{k-1} \exp \frac{q_k + q_{k-1}}{\lambda} \propto \dots \propto \exp \frac{\sum_{j=0}^{k} q_j}{\lambda} = \exp \frac{(k+1)h_k}{\lambda}.$$

Still thanks to Eq. (8), this means that π_{k+1} satisfies

$$\pi_{k+1} = \operatorname*{argmax}_{\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}} \left(\langle \pi, h_k \rangle + \frac{\lambda}{k+1}\mathcal{H}(\pi) \right) = \mathcal{G}^{0, \frac{\lambda}{k+1}}(h_k).$$

This shows that Eq. (22) is equivalent to

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k) \\ q_{k+1} = (T^{1-2}_{\pi_{k+1}|\pi_k})^m q_k + \epsilon_{k+1} \\ h_{k+1} = \frac{k+1}{k+2}h_k + \frac{1}{k+2}q_{k+1} \end{cases}, \tag{23}$$

which is DA-MPI$_{1\text{-}2}$(λ,0), and this shows the first part of the result. In the limit $\lambda \to 0$, the regularized greediness becomes the classic greediness (hard maximum over q-values) and the (regularized) evaluation operator becomes the classic one. However, notice that schemes are not equivalent in the limit: scheme (22) tends to classic VI, while scheme (23) tends to momentum VI (Vieillard et al., 2019).

Next, we prove the equivalence for the case $\tau > 0$. We recall MD-MPI$_{1\text{-}2}$(λ,τ):

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{\lambda,\tau}_{\pi_k}(q_k) \\ q_{k+1} = (T^{1-2}_{\pi_{k+1}|\pi_k})^m q_k + \epsilon_{k+1} \end{cases}. \tag{24}$$

Thanks to Eq. (8), we have that $\pi_{k+1} \propto \exp \frac{q_k + \lambda \ln \pi_k}{\lambda+\tau}$. We define $\beta = \frac{\lambda}{\lambda+\tau}$ (and thus $1-\beta = \frac{\tau}{\lambda+\tau}$ and $\frac{\beta}{\lambda} = \frac{1}{\lambda+\tau}$). By induction, we have (writing cst any function depending solely on states, not necessarily the same for different lines):

$$\begin{aligned} \ln \pi_{k+1} &= \frac{\beta}{\lambda}q_k + \beta \ln \pi_k + \text{cst} \\ &= \frac{\beta}{\lambda}\left(q_k + \beta q_{k-1} + \beta^2 q_{k-2} + \dots\right) + \text{cst} \\ &= \frac{\beta}{\lambda(1-\beta)}\left((1-\beta)(q_k + \beta q_{k-1} + \beta^2 q_{k-2} + \dots)\right) + \text{cst}\,. \end{aligned}$$

We now define h_k as the moving average of past q-values, with $h_{-1} = 0$:

$$h_k = \beta h_{k-1} + (1-\beta)q_k = (1-\beta)\sum_{j=0}^{k}\beta^{k-j}q_j. \tag{25}$$

Noticing also that $\frac{\beta}{\lambda(1-\beta)} = \frac{1}{\tau}$, this shows that

$$\pi_{k+1} \propto \exp \frac{h_k}{\tau}$$

As before, this means that π_{k+1} is the solution of an entropy regularized greedy step with respect to h_k:

$$\pi_{k+1} = \operatorname*{argmax}_{\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}} \left(\langle \pi, h_k \rangle + \tau\mathcal{H}(\pi) \right) = \mathcal{G}^{0,\tau}(h_k).$$

This means that Eq. (24) is equivalent to

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{0,\tau}(h_k) \\ q_{k+1} = (T^{1-2}_{\pi_{k+1}|\pi_k})^m q_k + \epsilon_{k+1} \\ h_{k+1} = \beta h_k + (1-\beta)q_{k+1} \text{ with } \beta = \frac{\lambda}{\lambda+\tau} \end{cases},$$

which is the DA-MPI$_{1\text{-}2}$(λ,τ) scheme. This concludes the proof.

C.2. Proof of Theorem 1

Here, we provide the bound for DA-VI$_1$(λ,0), which we recall:

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k) \\ q_{k+1} = T^{\lambda,0}_{\pi_{k+1}|\pi_k} q_k + \epsilon_{k+1} \\ h_{k+1} = \frac{k+1}{k+2} h_k + \frac{1}{k+2} q_{k+1} \end{cases} . \tag{26}$$

We start by stating a useful lemma.

Lemma 1. *For any $q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ and $\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}$, we have*

$$q_\pi - q = (I - \gamma P_\pi)^{-1}(T_\pi q - q).$$

Proof. This result is classic, and appears many times in the literature (*e.g.*, Kakade & Langford (2002)). We provide a one line proof for completeness, relying on basic properties of the Bellman operator:

$$q_\pi - q = T_\pi q_\pi - T_\pi q + T_\pi q - q = \gamma P_\pi (q_\pi - q) + T_\pi q - q \Leftrightarrow q_\pi - q = (I - \gamma P_\pi)^{-1}(T_\pi q - q).$$

□

The aim is to bound the quantity $q_* - q_{\pi_{k+1}}$, the difference between the optimal value function and the value function computed by DA-VI$_1$(λ,0). Thanks to Lemma 1, we can decompose this term as

$$\begin{aligned} q_* - q_{\pi_{k+1}} &= q_* - h_k + h_k - q_{\pi_{k+1}} \\ &= (I - \gamma P_{\pi_*})^{-1}(T_{\pi_*} h_k - h_k) - (I - \gamma P_{\pi_{k+1}})^{-1}(T_{\pi_{k+1}} h_k - h_k). \end{aligned} \tag{27}$$

Notice that $q_* = q_{\pi_*}$ for any optimal policy π_*. There exists an optimal deterministic policy (Puterman, 2014), so we will consider a deterministic π_*. As for any deterministic policy, $\mathcal{H}(\pi_*) = 0$. Using the definition of π_{k+1}, we have

$$\begin{aligned} \pi_{k+1} = \mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k) &\Rightarrow \langle \pi_{k+1}, h_k \rangle + \frac{\lambda}{k+1}\mathcal{H}(\pi_{k+1}) \geq \langle \pi_*, h_k \rangle + \frac{\lambda}{k+1} \underbrace{\mathcal{H}(\pi_*)}_{=0} \\ &\Rightarrow r + \gamma P \left(\langle \pi_{k+1}, h_k \rangle + \frac{\lambda}{k+1}\mathcal{H}(\pi_{k+1}) \right) \geq r + \gamma P \langle \pi_*, h_k \rangle \\ &\Rightarrow T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k = T_{\pi_{k+1}} h_k + \gamma \frac{\lambda}{k+1} P \mathcal{H}(\pi_{k+1}) \geq T_{\pi_*} h_k. \end{aligned}$$

Injecting this into Eq. (27), we obtain, using the fact that for any π the matrix $(I - \gamma P_\pi)^{-1} = \sum_{t\geq 0} \gamma^t P_\pi^t$ is positive,

$$q_* - q_{\pi_{k+1}} \leq (I - \gamma P_{\pi_*})^{-1}(T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k - h_k) - (I - \gamma P_{\pi_{k+1}})^{-1}(T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k - h_k - \gamma \frac{\lambda}{k+1} P \mathcal{H}(\pi_{k+1})). \tag{28}$$

So, what we have to do is to control the residual $T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k - h_k$.

To do so, the following lemma will be useful.

Lemma 2. *For any $k \geq 1$, we have that*

$$T^{\lambda,0}_{\pi_{k+1}|\pi_k} q_k = (k+1) T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k - k T^{0,\frac{\lambda}{k}}_{\pi_k} h_{k-1}.$$

For $k = 0$, we have

$$T^{\lambda,0}_{\pi_1|\pi_0} q_0 = T^{0,\lambda}_{\pi_1} h_0 - \gamma \lambda P \mathcal{H}(\pi_0).$$

Proof. To prove this result, we will start by working on the optimization problem related to the regularized greedy step $\mathcal{G}^{\lambda,0}_{\pi_k} q_k$:

$$\langle \pi, q_k \rangle - \lambda \operatorname{KL}(\pi || \pi_k) = \langle \pi, q_k \rangle - \lambda \langle \pi, \ln \pi - \ln \pi_k \rangle = \langle \pi, q_k + \lambda \ln \pi_k \rangle - \lambda \langle \pi, \ln \pi \rangle.$$

For DA-VI$_1$(λ,0), $\pi_{k+1} \in \mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k)$ (see Eq. (26)), so according to Eq. (8), $\pi_{k+1} \propto \exp \frac{(k+1)h_k}{\lambda}$. Therefore, we have, using also the definition of h_k

$$\begin{aligned}q_k + \lambda \ln \pi_k &= q_k + \lambda(\frac{k}{\lambda} h_{k-1} - \ln\langle 1, \exp \frac{k h_{k-1}}{\lambda}\rangle)\\ &= (k+1)h_k - \lambda \ln\langle 1, \exp \frac{k h_{k-1}}{\lambda}\rangle.\end{aligned}$$

Therefore, we have

$$\langle \pi, q_k\rangle - \lambda \operatorname{KL}(\pi||\pi_k) = \langle \pi, (k+1)h_k\rangle - \lambda\langle \pi, \ln \pi\rangle - \lambda \ln\langle \mathbf{1}, \exp \frac{k h_{k-1}}{\lambda}\rangle.$$

The maximizer is π_{k+1}, obviously. It is also the maximizer of $\langle \pi, (k+1)h_k\rangle - \lambda\langle \pi, \ln \pi\rangle$ (the third term not depending on π), and the associated maximum is, according to Eq. (7), $\lambda \ln\langle \mathbf{1}, \exp \frac{(k+1)h_k}{\lambda}\rangle$. This gives

$$\begin{aligned}\langle \pi_{k+1}, q_k\rangle - \lambda \operatorname{KL}(\pi_{k+1}||\pi_k) &= \lambda \ln\langle \mathbf{1}, \exp \frac{(k+1)h_k}{\lambda}\rangle - \lambda \ln\langle \mathbf{1}, \exp \frac{k h_{k-1}}{\lambda}\rangle\\ &= (k+1)\frac{\lambda}{k+1} \ln\langle \mathbf{1}, \exp \frac{(k+1)h_k}{\lambda}\rangle - k\frac{\lambda}{k} \ln\langle \mathbf{1}, \exp \frac{k h_{k-1}}{\lambda}\rangle.\end{aligned}$$

Still from Eq. (7), we know that $\frac{\lambda}{k+1} \ln\langle 1, \exp \frac{(k+1)h_k}{\lambda}\rangle$ is the maximum of $\langle \pi, h_k\rangle + \frac{\lambda}{k+1}\mathcal{H}(\pi)$, the associated maximizer being again π_{k+1}, so using Eq. (9), we can conclude that

$$\langle \pi_{k+1}, q_k\rangle - \lambda \operatorname{KL}(\pi_{k+1}||\pi_k) = (k+1)\left(\langle \pi_{k+1}, h_k\rangle + \frac{\lambda}{k+1}\mathcal{H}(\pi_{k+1})\right) - k\left(\langle \pi_k, h_{k-1}\rangle + \frac{\lambda}{k}\mathcal{H}(\pi_k)\right).$$

Noticing that $r = (k+1)r - kr$, we have the first part of the result:

$$T^{\lambda,0}_{\pi_{k+1}|\pi_k} q_k = (k+1) T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k - k T^{0,\frac{\lambda}{k}}_{\pi_k} h_{k-1}.$$

This only holds for $k \geq 1$. For $k = 0$, using the fact that $h_0 = q_0$,

$$\begin{aligned}T^{\lambda,0}_{\pi_1|\pi_0} q_0 &= r + \gamma P(\langle \pi_1, q_0\rangle - \lambda \operatorname{KL}(\pi_1||\pi_0))\\ &= r + \gamma P(\langle \pi_1, h_0\rangle - \lambda\langle \pi_1, \ln \pi_1 - \ln \pi_0\rangle)\\ &= r + \gamma P(\langle \pi_1, h_0\rangle + \lambda\mathcal{H}(\pi_1) + \lambda\langle \pi_1, \ln \pi_0\rangle)\\ &= T^{0,\lambda}_{\pi_1} h_0 - \gamma\lambda P\mathcal{H}(\pi_0),\end{aligned}$$

where we used in the last line the fact that, π_0 being uniform,

$$\langle \pi_1, \ln \pi_0\rangle = \langle \pi_1, \ln \frac{1}{|\mathcal{A}|}\rangle = -\ln|\mathcal{A}|\langle \pi_1, 1\rangle = -\ln|\mathcal{A}| = -\mathcal{H}(\pi_0).$$

This concludes the proof. □

Using this lemma, we can provide a Bellman-like induction on h_k.

Lemma 3. *Define $E_k = -\sum_{j=1}^k \epsilon_j$. For any $k \geq 1$, we have that*

$$h_{k+1} = \frac{k+1}{k+2} T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k + \frac{1}{k+2}\left(q_0 - E_{k+1} - \gamma\lambda P\mathcal{H}(\pi_0)\right).$$

Proof. Using the definition of h_k, Lemma 2, the fact that $q_{k+1} = T^{\lambda,0}_{\pi_{k+1}|\pi_k} q_k + \epsilon_{k+1}$, and the definition $E_k = -\sum_{j=1}^k \epsilon_j$, we have

$$\begin{aligned}
(k+2)h_{k+1} &= \sum_{j=0}^{k+1} q_j \\
&= q_0 + q_1 + \sum_{j=1}^{k} q_{j+1} \\
&= q_0 + T^{\lambda,0}_{\pi_1|\pi_0} q_0 + \epsilon_1 + \sum_{j=1}^{k} \left(T^{\lambda,0}_{\pi_{j+1}} q_j + \epsilon_{j+1} \right) \\
&= q_0 + T^{0,\lambda}_{\pi_1} h_0 - \gamma\lambda P\mathcal{H}(\pi_0) + \sum_{j=1}^{k} \left((j+1) T^{0,\frac{\lambda}{j+1}}_{\pi_{j+1}} h_j - j T^{0,\frac{\lambda}{j}}_{\pi_j} h_{j-1} \right) - E_{k+1} \\
&= q_0 - E_{k+1} - \gamma\lambda P\mathcal{H}(\pi_0) + (k+1) T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k \\
\Leftrightarrow h_{k+1} &= \frac{k+1}{k+2} T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k + \frac{1}{k+2} \left(q_0 - E_{k+1} - \gamma\lambda P\mathcal{H}(\pi_0) \right).
\end{aligned}$$

□

We know have the tools to work on the residual of interest. Starting from Lemma 3, and using the fact that $(k+2)h_{k+1} = (k+1)h_k + q_{k+1}$,

$$\begin{aligned}
h_{k+1} &= \frac{k+1}{k+2} T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k + \frac{1}{k+2} \left(q_0 - E_{k+1} - \gamma\lambda P\mathcal{H}(\pi_0) \right) \\
\Leftrightarrow (k+1)h_k + q_{k+1} &= q_0 - E_{k+1} - \gamma\lambda P\mathcal{H}(\pi_0) + (k+1) T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k \\
\Leftrightarrow T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k - h_k &= \frac{1}{k+1} \left(q_{k+1} - q_0 + E_{k+1} + \gamma\lambda P\mathcal{H}(\pi_0) \right).
\end{aligned}$$

Injecting this last result into decomposition (28), we get

$$\begin{aligned}
q_* - q_{\pi_{k+1}} &\leq (I - \gamma P_{\pi_*})^{-1} (T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k - h_k) - (I - \gamma P_{\pi_{k+1}})^{-1} (T^{0,\frac{\lambda}{k+1}}_{\pi_{k+1}} h_k - h_k - \gamma\lambda P\mathcal{H}(\pi_{k+1})) \\
&= (I - \gamma P_{\pi_*})^{-1} \left(\frac{1}{k+1} \left(q_{k+1} - q_0 + E_{k+1} + \gamma\lambda P\mathcal{H}(\pi_0) \right) \right) \\
&\quad - (I - \gamma P_{\pi_{k+1}})^{-1} \left(\frac{1}{k+1} \left(q_{k+1} - q_0 + E_{k+1} + \gamma\lambda P\mathcal{H}(\pi_0) \right) - \gamma \frac{\lambda}{k+1} P\mathcal{H}(\pi_{k+1}) \right) \\
&\leq (I - \gamma P_{\pi_*})^{-1} \left(\frac{1}{k+1} \left(q_{k+1} - q_0 + E_{k+1} + \gamma\lambda P\mathcal{H}(\pi_0) \right) \right) \\
&\quad - (I - \gamma P_{\pi_{k+1}})^{-1} \left(\frac{1}{k+1} \left(q_{k+1} - q_0 + E_{k+1} - \gamma\lambda P\mathcal{H}(\pi_{k+1}) \right) \right),
\end{aligned}$$

where we used for the last inequality the fact that $-(I - \gamma P_{\pi_*})^{-1} P\mathcal{H}(\pi_0) \leq 0$. Next, using the fact that $q_* - q_{\pi_{k+1}} \geq 0$ and rearranging terms, we have

$$\begin{aligned}
q_* - q_{\pi_{k+1}} \leq & \left| \left((I - \gamma P_{\pi_*})^{-1} - (I - \gamma P_{\pi_{k+1}})^{-1} \right) \frac{E_{k+1}}{k+1} \right| \\
& + (I - \gamma P_{\pi_*})^{-1} \left| \frac{q_{k+1} - q_0 + \gamma\lambda P\mathcal{H}(\pi_0)}{k+1} \right| + (I - \gamma P_{\pi_{k+1}})^{-1} \left| \frac{q_{k+1} - q_0 + \gamma\lambda P\mathcal{H}(\pi_{k+1})}{k+1} \right|.
\end{aligned}$$

We assumed that $\|q_{k+1}\|_\infty \leq v_{\text{max}} \leq v^\lambda_{\text{max}}$ (see also Rk. 1). When introducing the algorithm, we assumed that $\|q_0\|_\infty \leq v_{\text{max}}$. Therefore, $\|q_0 - \gamma\lambda P\mathcal{H}(\pi_0)\|_\infty \leq v^\lambda_{\text{max}}$. Writing $\mathbf{1}$ the vector whose components are all 1, we get $|q_{k+1} - q_0 + \gamma\lambda P\mathcal{H}(\pi_0)| \leq 2v^\lambda_{\text{max}}\mathbf{1}$. Notice that for any policy π, we have that $P_\pi \mathbf{1} = \mathbf{1}$. Therefore, we have

$$(I - \gamma P_{\pi_*})^{-1} \left| \frac{q_{k+1} - q_0 + \gamma\lambda P\mathcal{H}(\pi_0)}{k+1} \right| \leq \frac{2}{1-\gamma} \frac{v^\lambda_{\text{max}}}{k+1} \mathbf{1}.$$

With the same arguments, we have that

$$(I-\gamma P_{\pi_{k+1}})^{-1}\left|\frac{q_{k+1}-q_0+\gamma\lambda P\mathcal{H}(\pi_{k+1})}{k+1}\right| \leq \frac{2}{1-\gamma}\frac{v^\lambda_{\text{max}}}{k+1}\mathbf{1}.$$

We finally have

$$q_* - q_{\pi_{k+1}} \leq \left|\left((I-\gamma P_{\pi_*})^{-1} - (I-\gamma P_{\pi_{k+1}})^{-1}\right)\frac{E_{k+1}}{k+1}\right| + \frac{4}{1-\gamma}\frac{v^\lambda_{\text{max}}}{k+1}\mathbf{1},$$

which is the stated result.

C.3. About Remark 1

We stated in Rk. 1, in the context of DA-VI$_1$(λ,0), that the assumption $\|q_k\|_\infty \leq v_{\text{max}}$ holds without approximation and is not strong with approximation, as this just requires clipping the q-values. We justify these statements here.

No approximation. We will proceed by induction. Assume that $\|q_k\|_\infty \leq v_{\text{max}}$. We assumed generally that $\|q_0\|_\infty \leq v_{\text{max}}$. Without error, the considered scheme is

$$\begin{cases}\pi_{k+1} = \mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k) \\ q_{k+1} = T^{\lambda,0}_{\pi_{k+1}|\pi_k} q_k \\ h_{k+1} = \frac{k+1}{k+2} h_k + \frac{1}{k+2} q_{k+1}\end{cases} \Leftrightarrow \begin{cases}\pi_{k+1} = \mathcal{G}^{\lambda,0}(q_k) \\ q_{k+1} = T^{\lambda,0}_{\pi_{k+1}|\pi_k} q_k\end{cases}.$$

As $\pi_{k+1} = \mathcal{G}^{\lambda,0}(q_k)$, we have that

$$q_{k+1} = T^{\lambda,0}_{\pi_{k+1}|\pi_k} q_k \geq T^{\lambda,0}_{\pi_k|\pi_k} q_k = T_{\pi_k} q_k \geq -v_{\text{max}}\mathbf{1},$$

The inequality making use of the induction argument. On the other hand, making use of the positiveness of the KL divergence, we have that

$$q_{k+1} = T^{\lambda,0}_{\pi_{k+1}|\pi_k} q_k \leq T_{\pi_{k+1}} q_k \leq v_{\text{max}}\mathbf{1},$$

where again the inequality comes from the induction argument. This allows concluding, $\|q_{k+1}\|_\infty \leq v_{\text{max}}$.

No approximation. Knowing a bound of the q-values without approximation, we can clip q_k such that it satisfies the bound, the effect of the clipping being part of the error. For example, assume that the evaluation step is approximated with a least-squares problems, a paramterized q-function, the target being a sampling of $T^{\lambda,0}_{\pi_{k+1}|\pi_k} q_k$, q_k being the previous approximation (for example the target network). We can clip the result of the least-squares in $[-v^\lambda_{\text{max}}, +v_{\text{max}}]$ and call the resulting function q_{k+1}. The resulting error is defined as $\epsilon_{k+1} = q_{k+1} - T^{\lambda,0}_{\pi_{k+1}|\pi_k} q_k$.

C.4. Proof of Theorem 2

In this section, we provide a bound for DA-VI$_1$(λ,τ). First, we recall the scheme:

$$\begin{cases}\pi_{k+1} = \mathcal{G}^{0,\tau}(h_k) \\ q_{k+1} = T^{\lambda,\tau}_{\pi_{k+1}|\pi_k} q_k + \epsilon_{k+1} \\ h_{k+1} = \beta h_k + (1-\beta) q_{k+1} \text{ with } \beta = \frac{\lambda}{\lambda+\tau}\end{cases}.$$

We recall that due to the entropy term, this scheme cannot converge to the unregularized optimal q_* function. Yet, without errors and with $\lambda = 0$, it would converge to the solution of the MDP regularized by the scaled entropy (Geist et al., 2019) (optimizing for the reward augmented by the scaled entropy). Our bound will show that adding a KL penalty does not change this. We recall the notations introduced in the main paper. We already have defined the operator $T^{0,\tau}_\pi$. It has a unique fixed point, which we write q^τ_π. The unique optimal q-function is $q^\tau_* = \max_\pi q^\tau_\pi$. We write $\pi^\tau_* = \mathcal{G}^{0,\tau}(q^\tau_*)$ the associated unique optimal policy, and $q^\tau_{\pi^\tau_*} = q^\tau_*$.

The following lemma, generalizing Lemma 1 to the regularized Bellman operator, will be useful:

Lemma 4. *Let $\tau \geq 0$. For any $q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ and $\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}$, we have*

$$q_\pi^\tau - q = (I - \gamma P_\pi)^{-1}(T_\pi^{0,\tau} q - q).$$

Proof. The proof is the same as the one of Lemma 1, relying on the fact that the regularized Bellman operator has the same properies as the Bellman operator (Geist et al., 2019):

$$q_\pi^\tau - q = T_\pi^{0,\tau} q_\pi^\tau - T_\pi^{0,\tau} q + T_\pi^{0,\tau} q - q = \gamma P_\pi (q_\pi^\tau - q) + T_\pi^{0,\tau} q - q \Leftrightarrow q_\pi^\tau - q = (I - \gamma P_\pi)^{-1}(T_\pi^{0,\tau} q - q).$$

□

We will bound the quantity $q_*^\tau - q_{\pi_{k+1}}^\tau$, using the following decomposition, based on Lemma 4:

$$\begin{aligned} q_*^\tau - q_{\pi_{k+1}}^\tau &= q_*^\tau - h_k + h_k - q_{\pi_{k+1}}^\tau \\ &= (q_*^\tau - h_k) - (I - \gamma P_{\pi_{k+1}})^{-1}(T_{\pi_{k+1}}^{0,\tau} h_k - h_k). \end{aligned} \tag{29}$$

To do so, we will upper-bound $q_*^\tau - h_k$ and lower-bound $T_{\pi_{k+1}}^{0,\tau} h_k - h_k$ (we recall that the matrix $(I - \gamma P_{\pi_{k+1}})^{-1}$ is non-negative). This requires a Bellman-like induction on h_k. For this, the following intermediate lemma, similar to Lemma 2, will be useful.

Lemma 5. *For any $k \geq 0$, we have that*

$$T_{\pi_{k+1}|\pi_k}^{\lambda,\tau} q_k = \frac{1}{1-\beta}\left(T_{\pi_{k+1}}^{0,\tau} h_k - \beta T_{\pi_k}^{0,\tau} h_{k-1}\right).$$

Proof. We have that, for any π,

$$\begin{aligned} \langle \pi, q_k \rangle - \lambda \operatorname{KL}(\pi || \pi_k) + \tau \mathcal{H}(\pi) &= \langle \pi, q_k \rangle - \lambda \langle \pi, \ln \pi - \ln \pi_k \rangle - \tau \langle \pi, \ln \pi \rangle \\ &= \langle \pi, q_k + \lambda \ln \pi_k \rangle - (\lambda + \tau) \langle \pi, \ln \pi \rangle. \end{aligned}$$

As $\pi_{k+1} \propto \exp \frac{h_k}{\tau}$, using also the fact that $\beta = \frac{\lambda}{\lambda+\tau}$ and $1 - \beta = \frac{\tau}{\lambda+\tau}$, as well as the definition of h_k (25), we have

$$\begin{aligned} q_k + \lambda \ln \pi_k &= q_k + \lambda \left(\frac{h_{k-1}}{\tau} - \ln\langle \mathbf{1}, \exp \frac{h_{k-1}}{\tau}\rangle\right) \\ &= \frac{1}{1-\beta}\left((1-\beta) q_k + \beta h_{k-1} - \beta\tau \ln\langle \mathbf{1}, \exp \frac{h_{k-1}}{\tau}\rangle\right) \\ &= \frac{1}{1-\beta}\left(h_k - \beta\tau \ln\langle \mathbf{1}, \exp \frac{h_{k-1}}{\tau}\rangle\right). \end{aligned}$$

Hence, injecting this in the previous result, we get

$$\begin{aligned} \langle \pi, q_k + \lambda \ln \pi_k \rangle - (\lambda + \tau)\langle \pi, \ln \pi \rangle &= \langle \pi, q_k + \lambda \ln \pi_k \rangle - \frac{\tau}{1-\beta}\langle \pi, \ln \pi \rangle \\ &= \frac{1}{1-\beta}\left(\langle \pi, h_k \rangle - \tau \langle \pi, \ln \pi \rangle - \beta\tau \ln\langle \mathbf{1}, \exp \frac{h_{k-1}}{\tau}\rangle\right). \end{aligned}$$

Now, as $\pi_{k+1} \propto \exp \frac{h_k}{\tau}$, we have that $\langle \pi_{k+1}, h_k \rangle + \tau \mathcal{H}(\pi_{k+1}) = \tau \ln\langle \mathbf{1}, \exp \frac{h_k}{\tau}\rangle$ (again from Eq. (9)), therefore

$$\langle \pi_{k+1}, q_k \rangle - \lambda \operatorname{KL}(\pi_{k+1} || \pi_k) + \tau \mathcal{H}(\pi_{k+1}) = \frac{1}{1-\beta}\left(\langle \pi_{k+1}, h_k \rangle + \tau \mathcal{H}(\pi_{k+1}) - \beta(\langle \pi_k, h_{k-1} \rangle + \tau \mathcal{H}(\pi_k))\right).$$

The result follows by the definition of $T_{\pi_{k+1}|\pi_k}^{\lambda,\tau} q_k = r + \gamma P(\langle \pi_{k+1}, q_k \rangle - \lambda \operatorname{KL}(\pi_{k+1} || \pi_k) + \tau \mathcal{H}(\pi_{k+1}))$, and noticing that $r = \frac{1}{1-\beta}(r - \beta r)$. □

This result allows to build the lemma stating a Bellman-like induction for h_k.

Lemma 6. *Define $E^\beta_{k+1} = -(1-\beta)\sum_{j=1}^{k+1}\beta^{k+1-j}\epsilon_j = \beta E^\beta_k + (1-\beta)\epsilon_{k+1}$ (with $E^\beta_0 = 0$). For any $k \geq 0$, we have that*

$$h_{k+1} = T^{0,\tau}_{\pi_{k+1}} h_k - E_{k+1} - \beta^{k+1}(T^{0,\tau}_{\pi_0} h_{-1} - h_0).$$

Proof. Using the definition of h_k, Eq. (25), the relationship between q_{k+1} and q_k, and Lemma 5, we have

$$\begin{aligned}
h_{k+1} &= (1-\beta)\sum_{j=0}^{k+1}\beta^{k+1-j} q_k \\
&= (1-\beta)\beta^{k+1} q_0 + (1-\beta)\sum_{j=1}^{k+1}\beta^{k+1-j} q_j \\
&= (1-\beta)\beta^{k+1} q_0 + (1-\beta)\sum_{j=0}^{k}\beta^{k-j} q_{j+1} \\
&= (1-\beta)\beta^{k+1} q_0 + (1-\beta)\sum_{j=0}^{k}\beta^{k-j}\left(T^{\lambda,\tau}_{\pi_{j+1}|\pi_j} q_j + \epsilon_{j+1}\right) \\
&= (1-\beta)\beta^{k+1} q_0 + (1-\beta)\sum_{j=0}^{k}\beta^{k-j}\left(\frac{1}{1-\beta}\left(T^{0,\tau}_{\pi_{j+1}} h_j - \beta T^{0,\tau}_{\pi_j} h_{j-1}\right) + \epsilon_{j+1}\right).
\end{aligned}$$

Let define E^β_{k+1} as

$$\begin{aligned}
E_{k+1} &= -(1-\beta)\sum_{j=0}^{k}\beta^{k-j}\epsilon_{j+1} \\
&= -(1-\beta)\sum_{j=1}^{k+1}\beta^{k+1-j}\epsilon_j \\
&= \beta E^\beta_k + (1-\beta)\epsilon_{k+1} \text{ with } E_0 = 0.
\end{aligned}$$

We also have

$$\begin{aligned}
(1-\beta)\sum_{j=0}^{k}\beta^{k-j}\left(\frac{1}{1-\beta}\left(T^{0,\tau}_{\pi_{j+1}} h_j - \beta T^{0,\tau}_{\pi_j} h_{j-1}\right)\right) &= \sum_{j=0}^{k}\beta^{k-j}\left(T^{0,\tau}_{\pi_{j+1}} h_j - \beta T^{0,\tau}_{\pi_j} h_{j-1}\right) \\
&= \sum_{j=1}^{k+1}\beta^{k+1-j} T^{0,\tau}_{\pi_j} h_{j-1} - \sum_{j=0}^{k}\beta^{k+1-j} T^{0,\tau}_{\pi_j} h_{j-1} \\
&= T^{0,\tau}_{\pi_{k+1}} h_k - \beta^{k+1} T^{0,\tau}_{\pi_0} h_{-1}.
\end{aligned}$$

Notice also that $h_0 = (1-\beta) q_0$. Putting all these parts together, we obtain

$$\begin{aligned}
h_{k+1} &= \beta^{k+1} h_0 - E^\beta_{k+1} + T^{0,\tau}_{\pi_{k+1}} h_k - \beta^{k+1} T^{0,\tau}_{\pi_0} h_{-1} \\
&= T^{0,\tau}_{\pi_{k+1}} h_k - E^\beta_{k+1} - \beta^{k+1}(T^{0,\tau}_{\pi_0} h_{-1} - h_0),
\end{aligned}$$

which is the stated result. □

Thanks to this result, we can now bound the terms of interest.

Upper-bounding $q_*^\tau - h_k$. Write $e_k = E_k^\beta + \beta^k (T_{\pi_0}^{0,\tau} h_{-1} - h_0)$, we have from Lemma 6 that $h_{k+1} = T_{\pi_{k+1}}^{0,\tau} h_k - e_{k+1}$. Then, we have :

$$\begin{aligned} q_*^\tau - h_{k+1} &= q_*^\tau - T_{\pi_{k+1}}^{0,\tau} h_k + e_{k+1} \\ &= \underbrace{T_{\pi_*^\tau}^{0,\tau} q_*^\tau - T_{\pi_*^\tau}^{0,\tau} h_k}_{=\gamma P_{\pi_*^\tau}(q_*^\tau - h_k)} + \underbrace{T_{\pi_*^\tau}^{0,\tau} h_k - T_{\pi_{k+1}}^{0,\tau} h_k}_{\leq 0 \text{ as } \pi_{k+1} = \mathcal{G}^{0,\tau}(h_k)} + e_{k+1} \\ &\leq \gamma P_{\pi_*^\tau}(q_*^\tau - h_k) + e_{k+1}. \end{aligned}$$

By direct induction, we obtain

$$\begin{aligned} q_*^\tau - h_{k+1} &\leq (\gamma P_{\pi_*^\tau})^{k+1}(q_*^\tau - h_0) + \sum_{j=1}^{k+1} (\gamma P_{\pi_*^\tau})^{k+1-j} e_j \\ &= (\gamma P_{\pi_*^\tau})^{k+1}(q_*^\tau - h_0) + \sum_{j=1}^{k+1} (\gamma P_{\pi_*^\tau})^{k+1-j} \left(E_j^\beta + \beta^j (T_{\pi_0}^{0,\tau} h_{-1} - h_0) \right). \end{aligned} \tag{30}$$

This is the desired upper-bound.

Lower-bounding $T_{\pi_{k+1}}^{0,\tau} h_k - h_k$. Using the same notation e_k, we have

$$\begin{aligned} T_{\pi_{k+1}}^{0,\tau} h_k - h_k &= \underbrace{T_{\pi_{k+1}}^{0,\tau} h_k - T_{\pi_k}^{0,\tau} h_k}_{\geq 0 \text{ as } \pi_{k+1} = \mathcal{G}^{0,\tau}(h_k)} + T_{\pi_k}^{0,\tau} h_k - h_k \\ &\geq T_{\pi_k}^{0,\tau} h_k - h_k \\ &= T_{\pi_k}^{0,\tau} \left(T_{\pi_k}^{0,\tau} h_{k-1} - e_k \right) - \left(T_{\pi_k}^{0,\tau} h_{k-1} - e_k \right) \text{ by Lemma 6} \\ &= \gamma P_{\pi_k} \left(T_{\pi_k}^{0,\tau} h_{k-1} - h_{k-1} \right) - (I - \gamma P_{\pi_k})^{-1} e_k. \end{aligned}$$

We define $P_{k:j} = P_{\pi_k} P_{\pi_{k-1}} \dots P_{\pi_{j+1}} P_{\pi_j}$ for $j \leq k$, with the convention $P_{k:k+1} = I$. By direct induction, the preceding inequality gives

$$\begin{aligned} T_{\pi_{k+1}}^{0,\tau} h_k - h_k &\geq \gamma^k P_{k:1} (T_{\pi_1}^{0,\tau} h_0 - h_0) - \sum_{j=1}^{k} \gamma^{k-j} P_{k:j+1} (I - \gamma P_{\pi_j}) e_j \\ &= \gamma^k P_{k:1} (T_{\pi_1}^{0,\tau} h_0 - h_0) - \sum_{j=1}^{k} \gamma^{k-j} P_{k:j+1} (I - \gamma P_{\pi_j}) (E_j^\beta + \beta^j (T_{\pi_0}^{0,\tau} h_{-1} - h_0)). \end{aligned} \tag{31}$$

Putting things together. Plugging Eqs. (30) and (31) into Eq. (29), we obtain

$$\begin{aligned} q_*^\tau - q_{\pi_{k+1}}^\tau \leq & (\gamma P_{\pi_*^\tau})^k (q_*^\tau - h_0) + \sum_{j=1}^{k} (\gamma P_{\pi_*^\tau})^{k-j} \left(E_j^\beta + \beta^j (T_{\pi_0}^{0,\tau} h_{-1} - h_0) \right) \\ & + (I - \gamma P_{\pi_{k+1}})^{-1} \left(-\gamma^k P_{k:1} (T_{\pi_1}^{0,\tau} h_0 - h_0) + \sum_{j=1}^{k} \gamma^{k-j} P_{k:j+1} (I - \gamma P_{\pi_j}) (E_j^\beta + \beta^j (T_{\pi_0}^{0,\tau} h_{-1} - h_0)) \right). \end{aligned}$$

Using the fact that $q_*^\tau - q_{\pi_{k+1}}^\tau \geq 0$, rearranging terms, we have

$$\begin{aligned} q_*^\tau - q_{\pi_{k+1}}^\tau \leq & \sum_{j=1}^{k} \left| (\gamma P_{\pi_*^\tau})^{k-j} + (I - \gamma P_{\pi_{k+1}})^{-1} \gamma^{k-j} P_{k:j+1} \left(I - \gamma P_{\pi_j}\right) E_j^\beta \right| \\ & + (\gamma P_{\pi_*^\tau})^k |q_*^\tau - h_0| + \sum_{j=1}^{k} (\gamma P_{\pi_*^\tau})^{k-j} \beta^j |T_{\pi_0}^{0,\tau} h_{-1} - h_0| + (I - \gamma P_{\pi_{k+1}})^{-1} \gamma^k P_{k:1} |T_{\pi_1}^{0,\tau} h_0 - h_0| \\ & + (I - \gamma P_{\pi_{k+1}})^{-1} \sum_{j=1}^{k} \gamma^{k-j} P_{k:j+1} (I + \gamma P_{\pi_j}) \beta^j |T_{\pi_0}^{0,\tau} h_{-1} - h_0|. \end{aligned} \tag{32}$$

The first term is related to the error, the others to the initialisation. We'll work on each of these other terms.

Recall that we assumed that $\|q_0\|_\infty \leq v_{\text{max}} = \frac{r_{\text{max}}}{1-\gamma}$. Therefore, $\|q_0\|_\infty \leq v_{\text{max}}^\tau = \frac{r_{\text{max}} + \tau \ln |\mathcal{A}|}{1-\gamma}$. As $h_0 = (1-\beta) q_0$, we have $\|h_0\|_\infty \leq (1-\beta) v_{\text{max}}^\tau$. From obvious properties of regularized MDPs (Geist et al., 2019), we have $\|q_*^\tau\|_\infty \leq v_{\text{max}}^\tau$. Therefore, writing $\mathbf{1} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ the vector with all components equal to 1, we have $|q_*^\tau - h_0| \leq (2-\beta) v_{\text{max}}^\tau \mathbf{1}$. Notice that for any policy π, we have $P_\pi \mathbf{1} = \mathbf{1}$, thus

$$(\gamma P_{\pi_*^\tau})^k |q_*^\tau - h_0| \leq \gamma^k (2-\beta) v_{\text{max}}^\tau \mathbf{1}.$$

We also have that $\|T_{\pi_1}^{0,\tau} h_0\|_\infty \leq r_{\text{max}} + \tau \ln |\mathcal{A}| + \gamma (1-\beta) v_{\text{max}}^\tau = (1 - \gamma\beta) v_{\text{max}}^\tau$, so

$$(I - \gamma P_{\pi_{k+1}})^{-1} \gamma^k P_{k:1} |T_{\pi_1}^{0,\tau} h_0 - h_0| \leq \gamma^k \frac{2 - (1+\gamma)\beta}{1-\gamma} v_{\text{max}}^\tau \mathbf{1}.$$

By definition $h_{-1} = 0$, so we have $\|T_{\pi_0}^{0,\tau} h_{-1}\|_\infty = \|r + \gamma P \tau \mathcal{H}(\pi_0)\|_\infty \leq r_{\text{max}} + \tau \ln |\mathcal{A}| = (1-\gamma) v_{\text{max}}^\tau$, so $\|T_{\pi_0}^{0,\tau} h_{-1} - h_0\|_\infty \leq (2 - \gamma - \beta) v_{\text{max}}^\tau$. Therefore, we have the following bound:

$$\sum_{j=1}^{k} (\gamma P_{\pi_*^\tau})^{k-j} \beta^j |T_{\pi_0}^{0,\tau} h_{-1} - h_0| \leq \gamma^k \sum_{j=1}^{k} \left(\frac{\beta}{\gamma}\right)^j (2 - \beta - \gamma) v_{\text{max}}^\tau \mathbf{1}.$$

Similarly, for the last term we have

$$(I - \gamma P_{\pi_{k+1}})^{-1} \sum_{j=1}^{k} \gamma^{k-j} P_{k:j+1} (I + \gamma P_{\pi_j}) \beta^j |T_{\pi_0}^{0,\tau} h_{-1} - h_0| \leq \frac{1+\gamma}{1-\gamma} \gamma^k \sum_{j=1}^{k} \left(\frac{\beta}{\gamma}\right)^j (2 - \beta - \gamma) v_{\text{max}}^\tau \mathbf{1}.$$

Summing these four upper bounds, we obtain

$$\gamma^k (2-\beta) v_{\text{max}}^\tau \mathbf{1} + \gamma^k \frac{2 - (1+\gamma)\beta}{1-\gamma} v_{\text{max}}^\tau \mathbf{1} + \gamma^k \sum_{j=1}^{k} \left(\frac{\beta}{\gamma}\right)^j (2 - \beta - \gamma) v_{\text{max}}^\tau \mathbf{1} + \frac{1+\gamma}{1-\gamma} \gamma^k \sum_{j=1}^{k} \left(\frac{\beta}{\gamma}\right)^j (2 - \beta - \gamma) v_{\text{max}}^\tau \mathbf{1}$$

$$= 2\gamma^k \frac{2 - \beta - \gamma}{1-\gamma} \sum_{j=0}^{k} \left(\frac{\beta}{\gamma}\right)^j v_{\text{max}}^\tau \mathbf{1} = 2\gamma^k \left(1 + \frac{1-\beta}{1-\gamma}\right) \sum_{j=0}^{k} \left(\frac{\beta}{\gamma}\right)^j v_{\text{max}}^\tau \mathbf{1}.$$

Plugging this result into Eq. (32), we obtain the stated result:

$$q_*^\tau - q_{\pi_{k+1}}^\tau \leq \sum_{j=1}^{k} \left| (\gamma P_{\pi_*^\tau})^{k-j} + (I - \gamma P_{\pi_{k+1}})^{-1} \gamma^{k-j} P_{k:j+1} \left(I - \gamma P_{\pi_j}\right) E_j^\beta \right| + \gamma^k \left(1 + \frac{1-\beta}{1-\gamma}\right) \sum_{j=0}^{k} \left(\frac{\beta}{\gamma}\right)^j v_{\text{max}}^\tau \mathbf{1}.$$

C.5. Proof of Theorem 3

In this section, we provide a bound for DA-VI$_2$(λ,0), which we recall:

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{0, \frac{\lambda}{k+1}}(h_k) \\ q_{k+1} = T_{\pi_{k+1}} q_k + \epsilon_{k+1} \\ h_{k+1} = \frac{k+1}{k+2} h_k + \frac{1}{k+2} q_{k+1} \end{cases}.$$

We cannot apply the proof technique of DA-VI$_1$(λ,0). Indeed, the core of this proof was to show a Bellman-like iteration on h_k (Lemma 3), this relying heavily on the fact the the evaluation operator was regularized (Lemma 2). As the evaluation operator is no longer regularized, the proof does not apply. The following proof is mainly a generalization of the one of MoVI (Vieillard et al., 2019).

As usual, we will decompose the term $q_* - q_{\pi_{k+1}}$ and bound each of its components. The optimal q-function q_* is the q-function of any optimal policy π_*. There exists a deterministic optimal policy (Puterman, 2014), so we will consider π_* to be deterministic, and use the fact that $\mathcal{H}(\pi_*) = 0$. We consider the following usual decomposition, making use of Lemma 1:

$$\begin{aligned} q_* - q_{\pi_{k+1}} &= q_* - h_k + h_k - q_{\pi_{k+1}} \\ &\leq (I - \gamma P_{\pi_*})^{-1}(T_{\pi_*} h_k - h_k) - (I - \gamma P_{\pi_{k+1}})^{-1}(T_{\pi_{k+1}} h_k - h_k). \end{aligned} \tag{33}$$

We'll upper-bound the term $T_{\pi_*} h_k - h_k$. To do so, the following lemma (a direct consequence of $\pi_{k+1} = \mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k)$) will be useful.

Lemma 7. *For any $k \geq 0$ and any policy π, we have for the couple (h_k, π_{k+1}) computed by DA-VI$_2$(λ,0) that*

$$T_{\pi_{k+1}}^{0,\frac{\lambda}{k+1}} h_k = T_{\pi_{k+1}} h_k + \gamma P \frac{\lambda}{k+1} \mathcal{H}(\pi_{k+1}) \geq T_\pi h_k + \gamma P \frac{\lambda}{k+1} \mathcal{H}(\pi) = T_\pi^{0,\frac{\lambda}{k+1}} h_k.$$

Proof. For any policy π, we have

$$\begin{aligned} \pi_{k+1} = \mathcal{G}^{0,\frac{\lambda}{k+1}}(h_k) &\Rightarrow \langle \pi_{k+1}, h_k \rangle + \frac{\lambda}{k+1}\mathcal{H}(\pi_{k+1}) \geq \langle \pi, h_k \rangle + \frac{\lambda}{k+1}\mathcal{H}(\pi) \\ &\Rightarrow T_{\pi_{k+1}}^{0,\frac{\lambda}{k+1}} h_k \geq T_\pi^{0,\frac{\lambda}{k+1}} h_k \text{ (by def. of the Bellman operators)} \\ &\Leftrightarrow T_{\pi_{k+1}} h_k + \gamma P \frac{\lambda}{k+1}\mathcal{H}(\pi_{k+1}) \geq T_\pi h_k + \gamma P \frac{\lambda}{k+1}\mathcal{H}(\pi). \end{aligned}$$

□

In particular, for $\pi = \pi_*$, using the fact that $\mathcal{H}(\pi_*) = 0$, we have

$$T_{\pi_{k+1}}^{0,\frac{\lambda}{k+1}} h_k \geq T_{\pi_*} h_k$$

Therefore, injecting this into Eq. (33) we get

$$q_* - q_{\pi_{k+1}} \leq (I - \gamma P_{\pi_*})^{-1}(T_{\pi_{k+1}}^{0,\frac{\lambda}{k+1}} h_k - h_k) - (I - \gamma P_{\pi_{k+1}})^{-1}(T_{\pi_{k+1}} h_k - h_k). \tag{34}$$

We will upper-bound $T_{\pi_{k+1}}^{0,\frac{\lambda}{k+1}} h_k - h_k$ and lower-bound $T_{\pi_{k+1}} h_k - h_k$.

Upper-bounding $T_{\pi_{k+1}}^{0,\frac{\lambda}{k+1}} h_k - h_k$. By definition, $h_k = \frac{k}{k+1} h_{k-1} + \frac{1}{k+1} q_k$, so

$$\begin{aligned} (k+1) T_{\pi_{k+1}} h_k &= k T_{\pi_{k+1}} h_{k-1} + T_{\pi_{k+1}} q_k \\ &= k T_{\pi_{k+1}} h_{k-1} + q_{k+1} - \epsilon_{k+1} \text{ (by def. of } q_{k+1}) \\ &\leq k \left(T_{\pi_k} h_{k-1} + \gamma P \frac{\lambda}{k} \mathcal{H}(\pi_k) - \gamma P \frac{\lambda}{k} \mathcal{H}(\pi_{k+1}) \right) + q_{k+1} - \epsilon_{k+1} \text{ (by Lemma 7)} \\ &= k T_{\pi_k} h_{k-1} + \gamma \lambda P \mathcal{H}(\pi_k) - \gamma \lambda P \mathcal{H}(\pi_{k+1}) + q_{k+1} - \epsilon_{k+1}. \end{aligned}$$

By direct induction, we obtain

$$(k+1) T_{\pi_{k+1}} h_k \leq \sum_{j=1}^{k+1} q_j - \sum_{j=1}^{k+1} \epsilon_j + \gamma \lambda P \sum_{j=1}^{k+1} (\mathcal{H}(\pi_{j-1}) - \mathcal{H}(\pi_j)).$$

For the first term, we have from the definition of h_k that $\sum_{j=1}^{k+1} q_j = (k+1)h_k + q_{k+1} - q_0$. For the second term, we defined previously (for the analysis of DA-VI$_1$(λ,0)) $E_k = -\sum_{j=1}^{k} \epsilon_j$. The last term is a telescoping sum, $\sum_{j=1}^{k+1}(\mathcal{H}(\pi_{j-1}) - \mathcal{H}(\pi_j)) = \mathcal{H}(\pi_0) - \mathcal{H}(\pi_{k+1})$. Therefore, we have

$$\begin{aligned}
&(k+1)T_{\pi_{k+1}} h_k \leq (k+1)h_k + q_{k+1} - q_0 + E_{k+1} + \gamma\lambda P(\mathcal{H}(\pi_0) - \mathcal{H}(\pi_{k+1}))\\
\Leftrightarrow\ &(k+1)(T_{\pi_{k+1}} h_k + \gamma \frac{\lambda}{k+1} P\mathcal{H}(\pi_{k+1}) - h_k) \leq q_{k+1} - q_0 + E_{k+1} + \gamma\lambda P\mathcal{H}(\pi_0)\\
\Leftrightarrow\ &T_{\pi_{k+1}}^{0,\frac{\lambda}{k+1}} h_k - h_k \leq \frac{1}{k+1}\left(q_{k+1} - q_0 + E_{k+1} + \gamma\lambda P\mathcal{H}(\pi_0)\right). \qquad (35)
\end{aligned}$$

This is the desired upper-bound.

Lower-bounding $T_{\pi_{k+1}} h_k - h_k$. Using again Lemma 7, as well as the definition of h_k, we have

$$\begin{aligned}
T_{\pi_{k+1}} h_k + \gamma P \frac{\lambda}{k+1}\mathcal{H}(\pi_{k+1}) &\geq T_{\pi_k} h_k + \gamma P \frac{\lambda}{k+1}\mathcal{H}(\pi_k)\\
\Leftrightarrow (k+1)T_{\pi_{k+1}} h_k &\geq (k+1)T_{\pi_k} h_k + \gamma\lambda P(\mathcal{H}(\pi_k) - \mathcal{H}(\pi_{k+1}))\\
&= (k+1)T_{\pi_k}\left(\frac{k}{k+1}h_{k-1} + \frac{1}{k+1}q_k\right) + \gamma\lambda P(\mathcal{H}(\pi_k) - \mathcal{H}(\pi_{k+1}))\\
&= kT_{\pi_k} h_{k-1} + T_{\pi_k} q_k + \gamma\lambda P(\mathcal{H}(\pi_k) - \mathcal{H}(\pi_{k+1})).
\end{aligned}$$

By direct induction, we obtain

$$(k+1)T_{\pi_{k+1}} h_k \geq \sum_{j=1}^{k} T_{\pi_j} q_j + \gamma\lambda P \underbrace{\sum_{j=1}^{k}(\mathcal{H}(\pi_j) - \mathcal{H}(\pi_{j+1}))}_{=\mathcal{H}(\pi_1)-\mathcal{H}(\pi_{k+1})\geq -\mathcal{H}(\pi_{k+1})} + \underbrace{T_{\pi_1} h_0}_{=T_{\pi_1} q_0}.$$

Subtracting $(k+1)h_k = \sum_{j=1}^{k} q_j$ from both sides we obtain

$$(k+1)(T_{\pi_{k+1}} h_k - h_k) \geq \sum_{j=1}^{k}(T_{\pi_j} q_j - q_j) + T_{\pi_1} q_0 - q_0 - \gamma\lambda P\mathcal{H}(\pi_{k+1}). \qquad (36)$$

Next, we will lower-bound the term $T_{\pi_j} q_j - q_j$.

Using the definition of q_j and basic properties of the Bellman operator, we have

$$\begin{aligned}
T_{\pi_j} q_j - q_j &= T_{\pi_j}(T_{\pi_j} q_{j-1} + \epsilon_j) - (T_{\pi_j} q_{j-1} + \epsilon_j)\\
&= \gamma P_{\pi_j}(T_{\pi_j} q_{j-1} - q_{j-1}) - (I - \gamma P_{\pi_j})\epsilon_j. \qquad (37)
\end{aligned}$$

To use an induction argument, we should replace $T_{\pi_j} q_{j-1}$ by $T_{\pi_{j-1}} q_{j-1}$. However, as the greediness is on h_k, not q_k, there is some work to relate these two terms:

$$\begin{aligned}
T_{\pi_{j+1}} q_j - T_{\pi_j} q_j &= T_{\pi_{j+1}}((j+1)h_j - jh_{j-1}) - T_{\pi_j}((j+1)h_j - jh_{j-1})\\
&= (j+1)\left(T_{\pi_{j+1}} h_j - T_{\pi_j} h_j\right) + j\left(T_{\pi_j} h_{j-1} - T_{\pi_{j+1}} h_{j-1}\right).
\end{aligned}$$

By Lemma 7 we have

$$T_{\pi_{j+1}} h_j - T_{\pi_j} h_j \geq \gamma P \frac{\lambda}{j+1}(\mathcal{H}(\pi_j) - \mathcal{H}(\pi_{j+1})) \text{ and } T_{\pi_j} h_{j-1} - T_{\pi_{j+1}} h_{j-1} \geq \gamma P \frac{\lambda}{j}(\mathcal{H}(\pi_{j+1}) - \mathcal{H}(\pi_j)).$$

Therefore,

$$T_{\pi_{j+1}} q_j - T_{\pi_j} q_j \geq (j+1)(\gamma P \frac{\lambda}{j+1}(\mathcal{H}(\pi_j) - \mathcal{H}(\pi_{j+1}))) + j(\gamma P \frac{\lambda}{j}(\mathcal{H}(\pi_{j+1}) - \mathcal{H}(\pi_j))) = 0.$$

Going back to Eq. (37) and using this last result, we have

$$\begin{aligned}T_{\pi_j} q_j - q_j &= \gamma P_{\pi_j}(T_{\pi_j} q_{j-1} - q_{j-1}) - (I - \gamma P_{\pi_j})\epsilon_j \\ &\geq \gamma P_{\pi_j}(T_{\pi_{j-1}} q_{j-1} - q_{j-1}) - (I - \gamma P_{\pi_j})\epsilon_j.\end{aligned}$$

Recall that we defined $P_{j:i} = P_{\pi_j} P_{\pi_{j-1}} \dots P_{\pi_i}$ if $1 \leq i \leq j$, $P_{j:i} = I$ otherwise. By direct induction, we obtain

$$T_{\pi_j} q_j - q_j \geq -\sum_{i=1}^{j} \gamma^{j-i} P_{j:i+1}(I - \gamma P_{\pi_i})\epsilon_i + \gamma^j P_{j:1}(T_{\pi_1} q_0 - q_0).$$

Plugging this in Eq. (36), we get

$$\begin{aligned}(k+1)(T_{\pi_{k+1}} h_k - h_k) &\geq \sum_{j=1}^{k}(T_{\pi_j} q_j - q_j) + T_{\pi_1} q_0 - q_0 - \gamma\lambda P\mathcal{H}(\pi_{k+1}) \\ &\geq \sum_{j=1}^{k}\left(-\sum_{i=1}^{j} \gamma^{j-i} P_{j:i+1}(I - \gamma P_{\pi_i})\epsilon_i + \gamma^j P_{j:1}(T_{\pi_1} q_0 - q_0)\right) + T_{\pi_1} q_0 - q_0 - \gamma\lambda P\mathcal{H}(\pi_{k+1}) \\ &= -\sum_{j=0}^{k-1} \gamma^j \sum_{i=1}^{k-j} P_{i+j:i+1}(I - \gamma P_{\pi_i})\epsilon_i + \sum_{j=0}^{k} \gamma^j P_{j:1}(T_{\pi_1} q_0 - q_0) - \gamma\lambda P\mathcal{H}(\pi_{k+1}) \\ &= -\sum_{j=1}^{k} \gamma^{k-j} \sum_{i=1}^{j} P_{i+j-k:i+1}(I - \gamma P_{\pi_i})\epsilon_i + \sum_{j=0}^{k} \gamma^j P_{j:1}(T_{\pi_1} q_0 - q_0) - \gamma\lambda P\mathcal{H}(\pi_{k+1}).\end{aligned}$$

Defining the weighted error $\mathcal{E}_{j,k} = -\sum_{i=1}^{j} P_{i+k-j:i+1}(I - \gamma P_{\pi_i})\epsilon_i$, we can rewrite the last equation as

$$T_{\pi_{k+1}} h_k - h_k \geq \frac{1}{k+1}\left(\sum_{j=0}^{k} \gamma^j P_{j:1}(T_{\pi_1} q_0 - q_0) + \sum_{j=1}^{k} \gamma^{k-j} \mathcal{E}_{j,k} - \gamma\lambda P\mathcal{H}(\pi_{k+1})\right). \tag{38}$$

This is the desired lower bound.

Putting things together. To get the final result, we just have to plug Eqs. (35) and (38) into Eq. (34):

$$\begin{aligned}q_* - q_{\pi_{k+1}} \leq &(I - \gamma P_{\pi_*})^{-1}\left(\frac{1}{k+1}\left(q_{k+1} - q_0 + E_{k+1} + \gamma\lambda P\mathcal{H}(\pi_0)\right)\right) \\ &- (I - \gamma P_{\pi_{k+1}})^{-1}\left(\frac{1}{k+1}\left(\sum_{j=0}^{k} \gamma^j P_{j:1}(T_{\pi_1} q_0 - q_0) + \sum_{j=1}^{k} \gamma^{k-j} \mathcal{E}_{j,k} - \gamma\lambda P\mathcal{H}(\pi_{k+1})\right)\right).\end{aligned}$$

Using the fact that $q_* - q_{\pi_{k+1}} \geq 0$ and rearranging term, we obtain

$$\begin{aligned}q_* - q_{\pi_{k+1}} \leq &\left|(I - \gamma P_{\pi_*})^{-1} \frac{E_{k+1}}{k+1} - (I - \gamma P_{\pi_{k+1}})^{-1} \frac{1}{k+1} \sum_{j=1}^{k} \gamma^{k-j} \mathcal{E}_{j,k}\right| \\ &+ (I - \gamma P_{\pi_*})^{-1} \frac{|q_{k+1} - q_0 + \gamma\lambda P\mathcal{H}(\pi_0)|}{k+1} \\ &+ (I - \gamma P_{\pi_{k+1}})^{-1} \frac{|\sum_{j=0}^{k} \gamma^j P_{j:1}(T_{\pi_1} q_0 - q_0) - \gamma\lambda P\mathcal{H}(\pi_{k+1})|}{k+1}.\end{aligned}$$

As we assumed $q_0 = 0$ and $\|q_k\|_\infty \leq v_{\text{max}}$, we have, similarly to previous bounds, that

$$(I - \gamma P_{\pi_*})^{-1} \frac{|q_{k+1} - q_0 + \gamma\lambda P\mathcal{H}(\pi_0)|}{k+1} \leq \frac{1}{1-\gamma} \frac{v_{\text{max}}^\lambda}{k+1} \mathbf{1}.$$

We also have that $\|T_{\pi_1} q_0 - q_0\|_\infty = \|T_{\pi_1} 0\|_\infty \leq r_{\max}$, so

$$(I - \gamma P_{\pi_{k+1}})^{-1} \frac{|\sum_{j=0}^k \gamma^j P_{j:1}(T_{\pi_1} q_0 - q_0) - \gamma\lambda P \mathcal{H}(\pi_{k+1})|}{k+1} \leq \frac{1}{1-\gamma} \frac{v^\lambda_{\max}}{k+1} \mathbf{1}.$$

This provides the stated bound:

$$q_* - q_{\pi_{k+1}} \leq \left| (I - \gamma P_{\pi_*})^{-1} \frac{E_{k+1}}{k+1} - (I - \gamma P_{\pi_{k+1}})^{-1} \frac{1}{k+1} \sum_{j=1}^k \gamma^{k-j} \mathcal{E}_{j,k} \right| + \frac{2}{1-\gamma} \frac{v^\lambda_{\max}}{k+1} \mathbf{1}.$$

C.6. The issue with DA-VI$_2$(λ,τ) and a workaround

Here, we provide more details on the issue with DA-VI$_2$(λ,τ), we extend footnote 5 regarding the mellowmax policy of Asadi & Littman (2017), and we propose a workaround for DA-VI$_2$(λ,τ), which consists in introducing a third type of Bellman evaluation.

The issue with DA-VI$_2$(λ,τ). We explained in Sec. 4.4 the issue, which is that DA-VI$_2$(0,τ) amounts to applying repeatedly an operator T_τ which is not contractive and that might have multiple fixed points, precluding convergence. The derivation of the equivalence between DA-VI$_2$(0,τ) and the repeated application of T_τ was indeed done while justifying the link to softmax-DQN in Appx. B.1. We repeat it here shortly for clarity. DA-VI$_2$(0,τ) is the following scheme (without error, written here as the equivalent MD-VI$_2$(0,τ)):

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{0,\tau} q_k \\ q_{k+1} = T_{\pi_{k+1}} q_k \end{cases}$$

We have seen a number of time that $\pi_{k+1} = \mathcal{G}^{0,\tau}(q_k) = \frac{\exp \frac{q_k}{\tau}}{\langle \mathbf{1}, \exp \frac{q_k}{\tau} \rangle}$, that is, π_{k+1} is the softmax policy of q_k, with parameter τ. Thus, DA-VI$_2$(0,τ) simplifies to

$$q_{k+1} = T_\tau q_k = r + \gamma P \left\langle \frac{\exp \frac{q_k}{\tau}}{\langle \mathbf{1}, \exp \frac{q_k}{\tau} \rangle}, q_k \right\rangle,$$

that is, the application of the softmax operator. As discussed by Asadi & Littman (2017), it might not be a contraction and can have multiple fixed points and be unstable.

About the mellowmax policy. Asadi & Littman (2017) introduced a so-called mellowmax policy as a convergent alternative to the softmax operator. This can be seen as a (complicated) way of regularizing the evaluation step, as stated in footnote 5. We explain here why. To do so, we reframe the mellowmax idea with our notations. Asadi & Littman (2017) introduced the mellowmax operator as

$$\operatorname{mm}_\tau(q) = \tau \ln \left\langle \mathbf{1}, \frac{1}{|\mathcal{A}|} \exp \frac{q}{\tau} \right\rangle.$$

One can easily see that it is indeed the convex conjugate of the KL with respect to the uniform policy (that behaves like the entropy). Indeed, from Eq. (7), we have directly that

$$\operatorname{mm}_\tau(q) = \max_{\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}} \left(\langle \pi, q \rangle - \tau \operatorname{KL}(\pi || \pi_U) \right),$$

with π_U the uniform policy. From Geist et al. (2019), we know that the following equivalent schemes,

$$\begin{cases} \pi_{k+1} = \operatorname{argmax}_{\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}} \left(\langle \pi, q \rangle - \tau \operatorname{KL}(\pi || \pi_U) \right) \\ q_{k+1} = T_{\pi_{k+1}} q_k - \gamma P \tau \operatorname{KL}(\pi_{k+1} || \pi_U) \end{cases} \Leftrightarrow q_{k+1} = r + \gamma P \operatorname{mm}_\tau(q_k),$$

are convergent (MDP regularized with $\lambda \operatorname{KL}(\cdot || \pi_U)$, the equivalence being from Eq. (9)). This is not the viewpoint of Asadi & Littman (2017). They try to find a policy π'_{k+1} such that $q_{k+1} = r + \gamma P \operatorname{mm}_\tau(q_k) = r + \gamma P \langle \pi'_{k+1}, q_k \rangle$. To account for the possible existence of multiple policies, they look for the one with maximal entropy and solve (numerically) for

$$\pi'_{k+1} = \max_{\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}} : \langle \pi, q_k \rangle = \operatorname{mm}_\tau(q_k)} \mathcal{H}(\pi).$$

Then, they apply $q_{k+1} = r + \gamma P \langle \pi'_{k+1}, q_k \rangle$. If there is no error when computing π'_{k+1}, this is equivalent to adding the regularization to the evaluation step, and we think a complicated way to do so.

A workaround to DA-VI$_2(\lambda,\tau)$. We explained in Sec. 4.4 that a workaround for the convergence problem would be to consider a third type of evaluation, ignoring the KL term but taking into account the entropy term. We define this third type:

$$T^3_{\pi|\mu} q = T^{0,\tau}_\pi q = T_\pi q + \tau \mathcal{H}(\pi).$$

The associated DA-VI$_3(\lambda,\tau)$ scheme is thus

$$\begin{cases} \pi_{k+1} = \mathcal{G}^{0,\tau}(h_k) \\ q_{k+1} = T^{0,\tau}_{\pi_{k+1}} q_k + \epsilon_{k+1} \\ h_{k+1} = \beta h_k + (1-\beta) q_{k+1} \text{ with } \beta = \frac{\lambda}{\lambda+\tau} \end{cases}.$$

We think that if one consider regularizing the evaluation step, it is worth considering both the KL and the entropy rather than just the entropy (because it is not more costly and leads to a better bound). Yet, for completeness, we provide the propagation of errors for DA-VI$_3(\lambda,\tau)$ in the following theorem. It is close to the bound of Thm 2, but with the weighted error of Thm. 3.

Theorem 4. *Define the moving average of weighted errors as $\mathcal{E}^\beta_{j,k} = -(1-\beta)\sum_{i=1}^j \beta^{j-i} P_{i+k-j:i+1}(I-\gamma P_{\pi_i})\epsilon_i$, with $P_{j:i}$ as in Thm. 3. Define E_k as in Thm. 1. Assume that $q_0 = 0$ (to simplify a bit the bound). We have that*

$$q^\tau_* - q^\tau_{\pi_{k+1}} \leq \left| \sum_{j=1}^k (\gamma P_{\pi^\tau_*})^{k-j} E^\beta_j - (I - \gamma P_{\pi_{k+1}})^{-1} \sum_{j=1}^k \gamma^{k-j} \mathcal{E}^\beta_{j,k} \right| + 2\gamma^k v^\tau_{max} \sum_{j=0}^k \left(\frac{\beta}{\gamma}\right)^j \mathbf{1}.$$

Proof. We start with a decomposition of $q^\tau_* - q^\tau_{\pi_{k+1}}$:

$$\begin{aligned} q^\tau_* - q^\tau_{\pi_{k+1}} &= q^\tau_* - h_k + h_k - q^\tau_{\pi_{k+1}} \\ &= q^\tau_* - h_k - (I - \gamma P_{\pi_{k+1}})^{-1}(T^{0,\tau}_{\pi_{k+1}} h_k - h_k) \text{ (by Lemma 4).} \end{aligned}$$

So, we'll upper-bound $q^\tau_* - h_k$ and lower-bound the residual $T^{0,\tau}_{\pi_{k+1}} h_k - h_k$.

Upper-bounding $q^\tau_* - h_k$. By definition, $h_k = \beta h_{k-1} + (1-\beta) q_k$, so

$$\begin{aligned} T^{0,\tau}_{\pi_{k+1}} h_k &= \beta \underbrace{T^{0,\tau}_{\pi_{k+1}} h_{k-1}}_{\leq T^{0,\tau}_{\pi_k} h_{k-1}} + (1-\beta) \underbrace{T^{0,\tau}_{\pi_{k+1}} q_k}_{= q_{k+1} - \epsilon_{k+1}} \\ &\leq \beta T^{0,\tau}_{\pi_k} h_{k-1} + (1-\beta)(q_{k+1} - \epsilon_{k+1}) \\ &\leq \beta^{k+1} T^{0,\tau}_{\pi_0} h_{-1} + \underbrace{(1-\beta)\sum_{j=1}^{k+1} \beta^{k+1-j} q_j}_{= h_{k+1} - (1-\beta)\beta^{k+1} q_0} - \underbrace{(1-\beta)\sum_{j=1}^{k+1} \beta^{k+1-j} \epsilon_j}_{= -E^\beta_{k+1}} \text{ (by direct induction)} \\ &= \beta^{k+1}(T^{0,\tau}_{\pi_0} h_{-1} - \underbrace{(1-\beta) q_0}_{= h_0}) + h_{k+1} + E^\beta_{k+1} \end{aligned}$$

Thus, we have

$$-h_{k+1} \leq -T^{0,\tau}_{\pi_{k+1}} h_k + E_{k+1} + \beta^{k+1}(T^{0,\tau}_{\pi_0} h_{-1} - h_0).$$

Using this, we can now derive the upper bound. We have

$$\begin{aligned} q^\tau_* - h_{k+1} &\leq q^\tau_* - T^{0,\tau}_{\pi_{k+1}} h_k + E_{k+1} + \beta^{k+1}(T^{0,\tau}_{\pi_0} h_{-1} - h_0) \\ &= T^{0,\tau}_{\pi^\tau_*} q^\tau_* - T^{0,\tau}_{\pi^\tau_*} h_k + \underbrace{T^{0,\tau}_{\pi^\tau_*} h_k - T^{0,\tau}_{\pi_{k+1}} h_k}_{\leq 0} + E_{k+1} + \beta^{k+1}(T^{0,\tau}_{\pi_0} h_{-1} - h_0) \\ &\leq \gamma P_{\pi^\tau_*}(q^\tau_* - h_k) + E_{k+1} + \beta^{k+1}(T^{0,\tau}_{\pi_0} h_{-1} - h_0). \end{aligned}$$

By direct induction, we get

$$q^\tau_* - h_{k+1} \leq (\gamma P_{\pi^\tau_*})^{k+1}(q^\tau_* - h_0) + \sum_{j=1}^{k+1} (\gamma P_{\pi^\tau_*})^{k+1-j} \left(E^\beta_j + \beta^j (T^{0,\tau}_{\pi_0} h_{-1} - h_0) \right).$$

This is the desired upper bound.

Lower-bounding $T^{0,\tau}_{\pi_{k+1}} h_k - h_k$. We have that:

$$\begin{aligned}
T^{0,\tau}_{\pi_{k+1}} h_k &\geq T^{0,\tau}_{\pi_k} h_k \text{ (as } \pi_{k+1} = \mathcal{G}^{0,\tau}(h_k)) \\
&= \beta T^{0,\tau}_{\pi_k} h_{k-1} + (1-\beta) T^{0,\tau}_{\pi_k} q_k \text{ (by def. of } h_k) \\
&\geq \beta^{k+1} T^{0,\tau}_{\pi_0} h_{-1} + (1-\beta) \sum_{j=0}^{k} \beta^{k-j} T^{0,\tau}_{\pi_j} q_j \text{ (by induction)} \\
\Leftrightarrow T^{0,\tau}_{\pi_{k+1}} h_k - h_k &\geq \beta^{k+1} T^{0,\tau}_{\pi_0} h_{-1} + (1-\beta) \sum_{j=0}^{k} \beta^{k-j} (T^{0,\tau}_{\pi_j} q_j - q_j).
\end{aligned}$$

Now, we'll bound the residuals $T^{0,\tau}_{\pi_j} q_j - q_j$:

$$\begin{aligned}
T^{0,\tau}_{\pi_j} q_j - q_j &= T^{0,\tau}_{\pi_j} \left(T^{0,\tau}_{\pi_j} q_{j-1} - \epsilon_{j-1} \right) - \left(T^{0,\tau}_{\pi_j} q_{j-1} - \epsilon_{j-1} \right) \text{ (by def. of } q_j) \\
&= \gamma P_{\pi_j} (T^{0,\tau}_{\pi_j} q_{j-1} - q_{j-1}) - (I - \gamma P_{\pi_j}) \epsilon_j
\end{aligned}$$

By definition of $h_k = \beta h_{k-1} + (1-\beta) q_k$, we have $q_k = \frac{h_k - \beta h_{k-1}}{1-\beta}$, so

$$\begin{aligned}
T^{0,\tau}_{\pi_{j+1}} q_j - T^{0,\tau}_{\pi_j} q_j &= T^{0,\tau}_{\pi_{j+1}} \left(\frac{h_j - \beta h_{j-1}}{1-\beta} \right) - T^{0,\tau}_{\pi_j} \left(\frac{h_j - \beta h_{j-1}}{1-\beta} \right) \\
&= \frac{1}{1-\beta} \left(\underbrace{T^{0,\tau}_{\pi_{j+1}} h_j - T^{0,\tau}_{\pi_j} h_j}_{\geq 0} + \beta \underbrace{(T^{0,\tau}_{\pi_j} h_{j-1} - T^{0,\tau}_{\pi_{j+1}} h_{j-1})}_{\geq 0} \right) \\
&\geq 0.
\end{aligned}$$

Therefore,

$$T^{0,\tau}_{\pi_j} q_j - q_j \geq \gamma P_{\pi_j} (T^{0,\tau}_{\pi_{j-1}} q_{j-1} - q_{j-1}) - (I - \gamma P_{\pi_j}) \epsilon_j.$$

By direct induction, and using the fact that the entropy is non-negative,

$$\begin{aligned}
T^{0,\tau}_{\pi_j} q_j - q_j &\geq - \sum_{i=1}^{j} \gamma^{j-i} P_{j:i+1} (I - \gamma P_{\pi_i}) \epsilon_i + \gamma^j P_{j:1} (T^{0,\tau}_{\pi_0} q_0 - q_0) \\
&\geq - \sum_{i=1}^{j} \gamma^{j-i} P_{j:i+1} (I - \gamma P_{\pi_i}) \epsilon_i + \gamma^j P_{j:1} (T_{\pi_0} q_0 - q_0).
\end{aligned}$$

Plugging this into the lower bound of the residual of interest, we obtain

$$\begin{aligned}
T_{\pi_{k+1}} h_k - h_k &\geq \beta^{k+1} T^{0,\tau}_{\pi_0} h_{-1} + (1-\beta) \sum_{j=0}^{k} \beta^{k-j} \gamma^j P_{j:1} (T_{\pi_0} q_0 - q_0) - (1-\beta) \sum_{j=0}^{k} \beta^{k-j} \sum_{i=1}^{j} \gamma^{j-i} P_{j:i+1} (I - \gamma P_{\pi_i}) \epsilon_i \\
&= \beta^{k+1} T^{0,\tau}_{\pi_0} h_{-1} + (1-\beta) \sum_{j=0}^{k} \beta^{k-j} \gamma^j P_{j:1} (T_{\pi_0} q_0 - q_0) - \sum_{j=1}^{k} \gamma^{k-j} (1-\beta) \sum_{i=1}^{j} \beta^{j-i} P_{i+k-j:i+1} (I - \gamma P_{\pi_i}) \epsilon_i
\end{aligned}$$

Defining the moving average of the weighted error $\mathcal{E}^{\beta}_{j,k} = -(1-\beta) \sum_{i=1}^{j} \beta^{j-i} P_{i+k-j:i+1} (I - \gamma P_{\pi_i}) \epsilon_i$, we get

$$T_{\pi_{k+1}} h_k - h_k \geq \beta^{k+1} T^{0,\tau}_{\pi_0} h_{-1} + (1-\beta) \sum_{j=0}^{k} \beta^{k-j} \gamma^j P_{j:1} (T_{\pi_0} q_0 - q_0) + \sum_{j=1}^{k} \gamma^{k-j} \mathcal{E}^{\beta}_{j,k}.$$

This is the desired lower bound.

Putting things together. Injecting the upper-bound and the lower bound into the decomposition of errors, we obtain

$$\begin{aligned}
q_*^\tau - q_{\pi_{k+1}}^\tau &\leq q_*^\tau - h_k - (I - \gamma P_{\pi_{k+1}})^{-1}(T_{\pi_{k+1}}^{0,\tau} h_k - h_k) \\
&\leq (\gamma P_{\pi_*^\tau})^k (q_*^\tau - h_0) + \sum_{j=1}^k (\gamma P_{\pi_*^\tau})^{k-j} \left(E_j^\beta + \beta^j (T_{\pi_0}^{0,\tau} h_{-1} - h_0)\right) \\
&\quad - (I - \gamma P_{\pi_{k+1}})^{-1} \left(\beta^{k+1} T_{\pi_0}^{0,\tau} h_{-1} + (1-\beta) \sum_{j=0}^k \beta^{k-j} \gamma^j P_{j:1}(T_{\pi_0} q_0 - q_0) + \sum_{j=1}^k \gamma^{k-j} \mathcal{E}_{j,k}^\beta \right).
\end{aligned}$$

We can bound a few terms (assuming that $q_0 = 0$), as was done in the previous proofs:

$$\begin{aligned}
\left|(\gamma P_{\pi_*^\tau})^k (q_*^\tau - h_0)\right| &\leq \gamma^k v_{\text{max}}^\tau \mathbf{1}, \\
\left|\sum_{j=1}^k (\gamma P_{\pi_*^\tau})^{k-j} \beta^j (T_{\pi_0}^{0,\tau} h_{-1} - h_0)\right| &\leq \gamma^k \sum_{j=1}^k \left(\frac{\beta}{\gamma}\right)^j v_{\text{max}}^\tau \mathbf{1}, \\
\left|(I - \gamma P_{\pi_{k+1}})^{-1} \beta^{k+1} T_{\pi_0}^{0,\tau} h_{-1}\right| &\leq \beta^{k+1} v_{\text{max}}^\tau \mathbf{1}, \\
\left|(I - \gamma P_{\pi_{k+1}})^{-1} (1-\beta) \sum_{j=0}^k \beta^{k-j} \gamma^j P_{j:1}(T_{\pi_0} q_0 - q_0)\right| &\leq (1-\beta)\beta^k \sum_{j=0}^k \left(\frac{\gamma}{\beta}\right)^j v_{\text{max}}^\tau \mathbf{1}.
\end{aligned}$$

Injecting this in the original bound and simplifying, we obtain

$$q_*^\tau - q_{\pi_{k+1}}^\tau \leq \left|\sum_{j=1}^k (\gamma P_{\pi_*^\tau})^{k-j} E_j^\beta - (I - \gamma P_{\pi_{k+1}})^{-1} \sum_{j=1}^k \gamma^{k-j} \mathcal{E}_{j,k}^\beta \right| + v_{\text{max}}^\tau \left(\gamma^k \sum_{j=0}^k \left(\frac{\beta}{\gamma}\right)^j + \beta^k \sum_{j=0}^k \left(\frac{\gamma}{\beta}\right)^j \right) \mathbf{1}.$$

Using the fact that $\gamma^k \sum_{j=0}^k \left(\frac{\beta}{\gamma}\right)^j = \beta^k \sum_{j=0}^k \left(\frac{\gamma}{\beta}\right)^j$, we obtain the stated result and concludes the proof. □

D. Empirical illustration of the bounds

To better understand the role played by β and τ, we implement a tabular version with a generative model of MD-VI$_1$(λ,τ). Recall that in this case MD-VI and DA-VI are equivalent. We call it Sampled MD-VI, it is described in Alg. 1. The error term comes from the sampling error made during the evaluation step. As such, the sequence of estimation errors is a martingale difference with respect to the natural filtration, and the average of errors vanishes asymptotically (Azar et al., 2011). We run this algorithm on randomly generated MDPs (garnets, described thereafter). As we have the model, we can compute q_*, and thus track the error made with respect to this optimal q-value. At each interaction $k \leq K$, we compute a policy π_k. For each random MDP, we compute the value

$$1 - \frac{1}{K\|q_*\|_1} \sum_{k=1}^K \|q_{\pi_k} - q_*\|_1,$$

that is the average normalized error between the q-value of the current policy and the optimal q-value. We then average this value over 10 random MDPs. We present the results in Figure 6, with the same visualisation method as in Section 6. In Figure 7, we present the same results, but comparing the regularized q-values, $q_{\pi_k}^\tau$ and q_*^τ. Actually, whenever $\tau > 0$, it is toward q_*^τ that the algorithmic scheme converges (at least without errors).

Definition of a Garnet. A Garnet (Archibald et al., 1995) is an abstract MDP, built from three parameters (N_S, N_A, N_B), with N_S and N_A respectively the number of states and actions. The principle is to directly build the transition kernel P that represnts the MDP. For each $(s, a) \in \mathcal{S} \times \mathcal{A}$, N_B states $(s_1, \dots s_{N_B})$ are drawn uniformly from $\mathcal{S}$ without replacement. Then, $N_B - 1$ numbers are drawn uniformly in $(0, 1)$ and sorted as $(p_0 = 0, p_1, \dots p_{N_B-1}, p_{N_B} = 1)$. The transition kernel is then defined as $P(s_k|s, a) = p_k - p_{k-1}$ for each $1 \leq k \leq N_B$. The reward function is drawn uniformly in $(0, 1)$ for 10% of the states, these states being drawn uniformly without replacement. We used $N_S = 30$, $N_A = 4$ and $N_B = 4$ in all our experiments.

Algorithm 1 Sampled MD-VI(λ, τ)

```
Require: $K$ number of iterations. $P$ the transition kernel from which we can only sample new states.
set $\beta = \frac{\lambda}{\lambda+\tau}$
set $q_0$ to the null vector
set $\pi_0$ to be the uniform policy
  for $1 \leq k \leq K$ do
    for $(s,a) \in \mathcal{S} \times \mathcal{A}$ do
      $\pi_k(a|s) = \frac{\pi_{k-1}(a|s)^\beta \exp \frac{q_{k-1}(s,a)}{\lambda+\tau}}{\sum_{b\in\mathcal{A}} \pi_{k-1}(b|s)^\beta \exp \frac{q_{k-1}(s,b)}{\lambda+\tau}}$
    end for
    for $(s,a) \in \mathcal{S} \times \mathcal{A}$ do
      $s' \sim P(\cdot|s,a)$
      $q_k(s,a) = r(s,a) + \gamma \sum_{b\in\mathcal{A}} q_{k-1}(s',b)\pi_k(b|s')$
    end for
  end for
output $\pi_K$
```

Discussion (w.r.t. q_*). The results in Fig 6 are consistent with the analysis. With $\tau > 0$, we observe that the lower τ is, the better the results are, as introducing an entropy term biases the solution. We also observe that the closer β is to 1, the better the results are: higher values of β further reduce the variance (without cancelling it asymptotically). We do not observe a clear difference between type 1 and type 2 here. We think that type 2 is especially harmful in the regime of high entropy, and we considered relatively small scales of entropy here ($\tau \leq 0.3$). This could explain this observation. For the case $\tau = 0$, we observe that the lower λ is, the better the results are. Here again, there is no clear difference between type 1 and type 2. This suggests that the additional weighting of the error terms by the transition kernels is not necessarily harmful, an observation already made by Vieillard et al. (2019) for MoVI.

Discussion (w.r.t. q_*^τ). In Fig. 7, we show the relative performance with respect to q_*^τ, the quantity to which the algorithm would converge without error. In this case, we would except the algorithm to behave similarly for the different values of τ, as only β has an influence in the bound. This is what we observe for the smaller values of τ. However, for larger values of the entropy, the results seem to improve. This might seem surprising, but this can be easily explained. The larger τ is, the closer to uniform is the optimal policy. The q-value being initialized as the null vector, the initial policy is uniform. Therefore, for large values of τ, we initialize closer to the optimal solution, which explains the different behavior.

E. Experimental details

Here, we give details on how we conducted the experiments on Deep RL algorithms presented in Section 5.

E.1. More on practical algorithms

In this section, we detail the losses presented in Section 5, giving equations that are closer to implementation, and providing a detailed pseudo-code in Algorithm 2. Firts, let us introduce some notations.The q-value is represented by a neural network Q_θ of parameters θ, and the policy is represented by a network Π_ϕ of parameters ϕ. During training, the algorithms interact with an environment, and collect transitions (s,a,r,s') that are stored in a FIFO replay buffer $\mathcal{B}$. The parameters of the networks are copied regularly into old versions of themselves, with target weights $\bar\theta$ and $\bar\phi$. The weights θ are optimized during the evaluation step, and ϕ during the greedy step.

E.1.1. Evaluation step

All the actor-critics we consider have the same update rule of their critic – the Q-network. We consider two regressions targets, corresponding to types 1 or 2. For type 2, we define a regression target as

$$\hat{Q}_2(r,s') = r + \gamma \sum_{b\in\mathcal{A}} Q_{\bar\theta}(s',b)\Pi_\phi(b|s'),$$

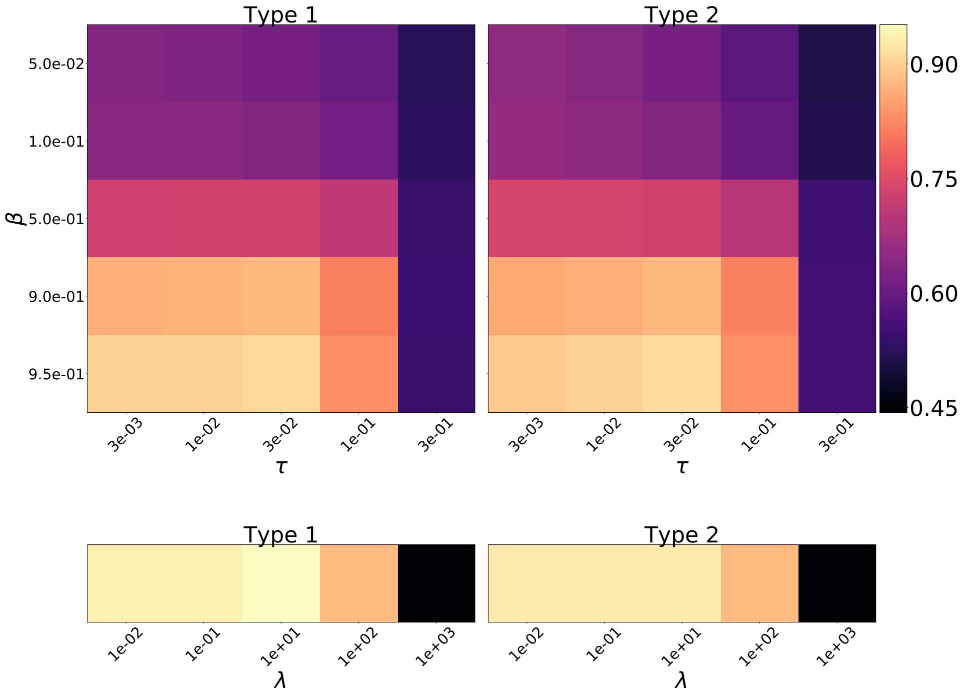

Figure 6. Relative performance of sampled MD-VI according to τ and β. We use a similar visualiation as in Section 6. The value we plot is $1 - \frac{1}{K\|q_*\|_1}\sum_k[\|q_{\pi_k} - q_*\|_1]$. The **top** grids correspond to values of $\tau > 0$, while the **bottom** ones correspond to $\tau = 0$.

and for type 1

$$\hat{Q}_1(r, s') = \hat{Q}_2(r, s') - \lambda \operatorname{KL}\left(\Pi_\phi \| \Pi_{\bar{\phi}}\right)(s') + \tau\mathcal{H}\left(\Pi_\phi\right)(s').$$

The weights θ are then updated by minimizing the following regression loss with a variant of SGD

$$\mathcal{L}_{1-2}(\theta) = \hat{E}_{\mathcal{B}}\left[\left(Q_\theta(s, a) - \hat{Q}_{1-2}(r, s')\right)^2\right]. \tag{39}$$

Note that if Π_ϕ was greedy with respect to $Q_{\bar{\theta}}$, using $\mathcal{L}_2$ would reduce to Deep Q-Networks (DQN) (Mnih et al., 2015).

E.1.2. Greedy step

Let us re-write in detail the three equations from Section 5 that define three ways of performing the greedy step.

MD-dir. The Direct MD update tackles directly the optimization problem derived from the greedy step. For convenience, we define a loss (the opposite of what we would like to maximize) that we minimize with SGD

$$\mathcal{L}_{\text{dir}}(\phi) = \hat{E}_{\mathcal{B}}\left[-\sum_{b\in\mathcal{A}} Q_{\bar{\theta}}(s, b)\Pi_\phi(b|s) + \lambda \operatorname{KL}\left(\Pi_\phi \| \Pi_{\bar{\phi}}\right)(s') - \tau\mathcal{H}\left(\Pi_\phi\right)(s')\right]. \tag{40}$$

MD-ind. The indirect version is based on the analytical result of the optimization problem corresponding to the greedy step. We show in Appendix B.1 that , at iteration k of MD-VI(λ, τ), we have $\pi_{k+1} = \mathcal{G}^{\lambda,\tau}_{\pi_k}(q_k) \propto \pi_k^\beta \exp \frac{q_k}{\tau+\lambda}$. Hence, we would need to fit a target that approximates this maximizer, by defining $\hat{\Pi}(a|s)$ as

$$\hat{\Pi}(a|s) = \Pi_{\bar{\phi}}(a|s)^\beta \exp \frac{Q_{\bar{\theta}}(s, a)}{\lambda + \tau}\left(\sum_{b\in\mathcal{A}} \Pi_{\bar{\phi}}(b|s)^\beta \exp \frac{Q_{\bar{\theta}}(s, b)}{\lambda + \tau}\right)^{-1}.$$

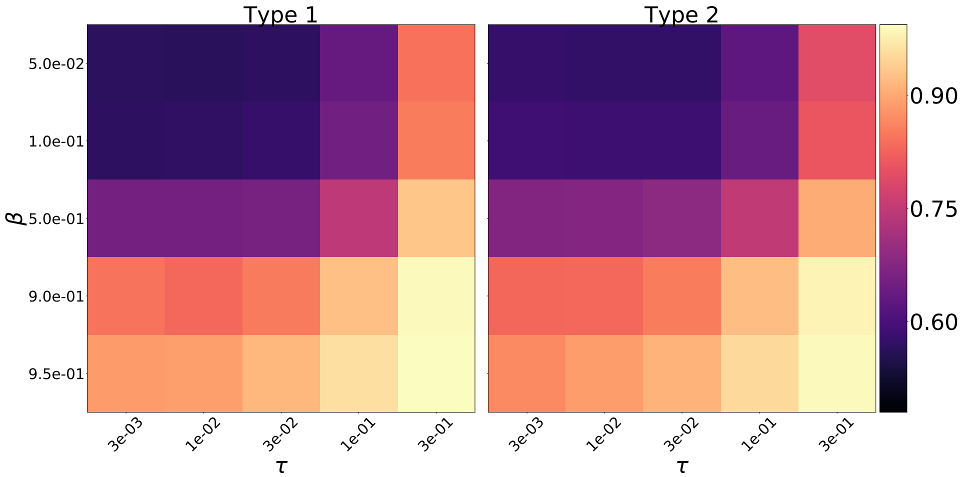

Figure 7. Relative performance of sampled MD-VI according to τ and β. The value we plot is $1 - \frac{1}{K\|q_*\|_1}\sum_k[\|q^\tau_{\pi_k} - q^\tau_*\|_1]$.

However, the exponential term can cause numerical problems, so what we optimize during the evaluation step is actually the logarithm of the policy. To work around this, we define a network L_ϕ that represents the log-probabilities of a policy, and we define a regression target

$$\hat{L}(s,a) = \frac{\lambda L_{\bar\phi}(a|s) + Q_{\bar\theta}(s,a)}{\lambda+\tau} - \ln \sum_{b\in\mathcal{A}} \frac{\lambda L_{\bar\phi}(b|s) + Q_{\bar\theta}(s,b)}{\lambda+\tau},$$

and then we have $\hat\Pi(a|s) = \exp\left(\hat{L}(s,a)\right)$ and $\Pi_\phi(a|s) = \exp\left(L_\phi(a|s)\right)$. We then define a loss on the parameters ϕ,

$$\mathcal{L}_{\text{ind}}(\phi) = \hat{E}_{\mathcal{B}}\left[\text{KL}\left(\hat\Pi\|\Pi_\phi\right)(s)\right]. \tag{41}$$

DA. The dual averaging version is inspired by the DA-VI formulation. Instead of representing directly the policy, we estimate a moving average of the q-values, and then compute its soft-maximum. The moving average is estimated via a network H_ϕ, which fits a regression target

$$\hat{H}(s,a) = \beta H_{\bar\phi}(s,a) + (1-\beta)Q_{\bar\theta}(s,a),$$

and the policy is defined as softmax over $H_\phi(s,\cdot)$,

$$\Pi_\phi(a|s) = \exp\frac{H_\phi(s,a)}{\tau}\left(\sum_b \exp\frac{H_\phi(s,b)}{\tau}\right)^{-1}.$$

The weights ϕ are optimized by mlinimizng the loss

$$\mathcal{L}_{\text{da}}(\phi) = \hat{E}_{\mathcal{B}}\left[\left(H_\phi(s,a) - \hat{H}(s,a)\right)^2\right]. \tag{42}$$

E.1.3. Pseudo code

We give a general pseudo-code of the deep RL algorithms we used in Alg. 2. Notice that for a policy π, we define the e-greedy policy with respect to π as the policy that takes a random action (uniformly on $\mathcal{A}$) with probability e, and follows π with probability $1-e$.

Algorithm 2 (MD-dir | MD-ind | DA)

```
Require: $L_q(\theta)$ and $L_\pi(\phi)$, two losses, respectively for the evaluation and the greediness. The choice of these losses determines the algorithm, see Table 1.
Require: $K \in \mathbb{N}^*$ the number of steps, $C \in \mathbb{N}^*$ the update period, $F \in \mathbb{N}^*$ the interaction period.
set $\theta$, $\phi$ at random
set $Q_\theta$ the Q-value network, $\Pi_\phi$ the policy network, as defined in Sec. E.1.
set $\mathcal{B} = \{\}$
set $\Pi_{\phi,e_k}$ the policy $e_k$-greedy w.r.t. $\Pi_\phi$
  $\bar{\theta} = \theta, \bar{\phi} = \phi$
  for $1 \leq k \leq K$ do
    Collect a transition $t = (s, a, r, s')$ from $\Pi_{\phi,e_k}$
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{t\}$
    if $k \mod F == 0$ then
      On a random batch of transitions $B_{q,k} \subset \mathcal{B}$, update $\theta$ with one step of SGD on $L_q$
      On a random batch of transitions $B_{h,k} \subset \mathcal{B}$, update $\phi$ with one step of SGD on $L_\pi$
    end if
    if $k \mod C == 0$ then
      $\bar{\theta} \leftarrow \theta, \bar{\phi} \leftarrow \phi$
    end if
  end for
output $\Pi_\phi$
```

Table 1. Resulting algorithms given the choice of losses in Algorithm 2

	L_π		
L_q	$\mathcal{L}_{\text{dir}}$ (Eq.(41))	$\mathcal{L}_{\text{ind}}$ (Eq. (40))	$\mathcal{L}_{\text{da}}$ (Eq. (42))
$\mathcal{L}_1$ (Eq. (39))	MD-dir Type 1	MD-ind Type 1	DA Type 1
$\mathcal{L}_2$ (Eq. (39))	MD-dir Type 2	MD-ind Type 2	DA Type 2

E.2. Hyperparameters

We provide the hyperparameters used on the Atari environments in Table 2, and on the Gym environments in Table 3. We use the following notations to describe neural networks: FC n is a fully connected layer with n neurons; $\text{Conv}^d_{a,b}\, c$ is a 2d convolutional layer with c filters of size $a \times b$ and a stride of d. All hyperparameters are the one found in the Dopamine code base. We only tuned the learning rate and the update period of DQN on Lunar Lander (not provided in Dopamine).

E.3. Additional results

Full tables Here, we provide the full results of the experiments from Section 6. The same plots are reported, expect that we add the exact value of each grid cell for completeness. Results for Carpole and Lunarlander are provided in Figs. 8 and 9, while results for the considered Atari games (Asterix, Breakout and Seaquest) are reported in Figs. 10, 11 and 12.

Training curves We also report training curves on Atari. We report training curves of DA, MD-dir and MD-ind in Fig. 13 for Asterix, on Fig. 14 for Breakout, and on Fig. 15 for Seaquest. We report the training curves of the limit cases on these three games on Figs. 16, 17 and 18. In these figures, an *iteration* corresponds to 250000 training steps, and we report every iteration the undiscounted reward averaged over the last 100 episodes (the *averaged score*). The training curves are averaged over 3 random seeds.

The training curves gives more hindsight on the performance of the algorithms. Indeed, the metric we used in the tables (the averaged score over all iteration) is partly flawed, because it could give a high score to an algorithm with a performance drop at the end of training. For example, the MD-dir method on Atari seems to benefit from type 1 compared to type 2 (as type 2 suffers from a performance drop), which is not obvious from the score tables. In almost all the cases, we do not observe such behaviour, which validates the use of our metric.

Table 2. Parameters used on Atari. Both the Q-network and policy-network have the same structure. n_A is the number of actions available in a given game.

Parameter	Value
K (number of steps)	$5 * 10^7$
C (update period)	8000
F (interaction period)	4
γ (discount)	0.99
$\vert\mathcal{B}\vert$ (replay buffer size)	10^6
$\vert B_{\pi,k}\vert$ and $\vert B_{q,k}\vert$ (batch size)	32
e_k (random actions rate)	$e_0 = 0.01$, linear decay of period $2.5 \cdot 10^5$ steps
networks structure	$\text{Conv}^4_{8,8}\,32 - \text{Conv}^2_{4,4}\,64 - \text{Conv}^1_{3,3}\,64 - \text{FC}\,512 - \text{FC}\,n_A$
activations	Relu
optimizers	RMSprop ($lr = 0.00025$)

Table 3. Parameters used on CartPole and Lunar Lander . Both the Q-network and policy-network have the same structure. We have $n_A = 2$ on CartPole, and $n_A = 8$ on Lunar Lander.

Parameter	Value
K (number of steps)	$5 * 10^5$
C (update period)	100 (Cartpole), 2500 (Lunar Lander)
F (interaction period)	4
γ (discount)	0.99
$\vert\mathcal{B}\vert$ (replay buffer size)	$5 * 10^4$
$\vert B_{\pi,k}\vert$ and $\vert B_{q,k}\vert$ (batch size)	128
e_k (random actions rate)	0.01 (constant with k)
networks structure	$\text{FC}\,512 - \text{FC}\,512 - \text{FC}\,n_A$
activations	Relu
optimizers	Adam ($lr = 0.001$)

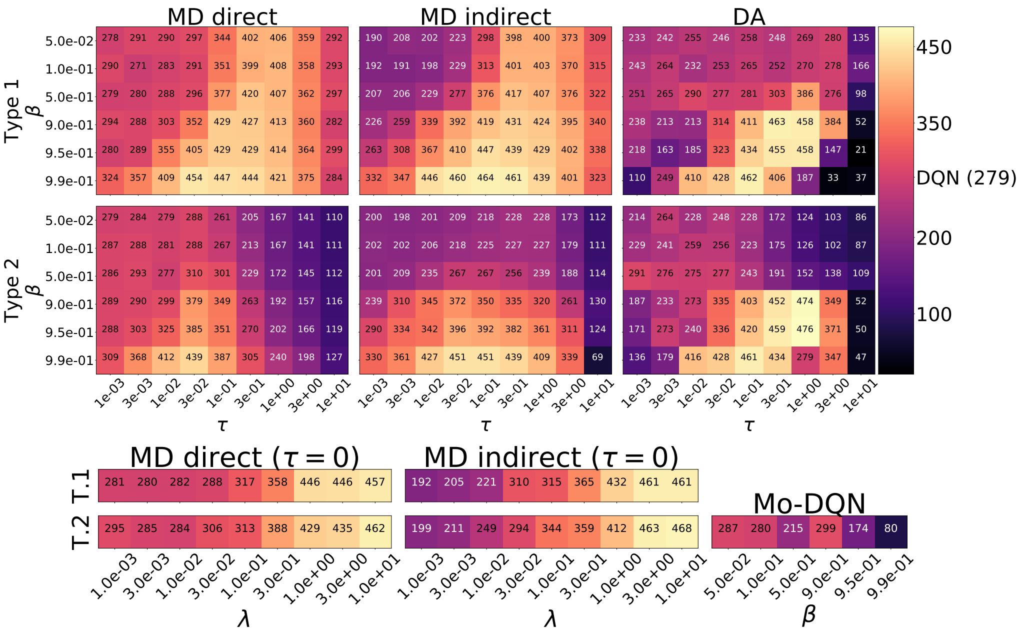

Figure 8. Cartpole with complete values.

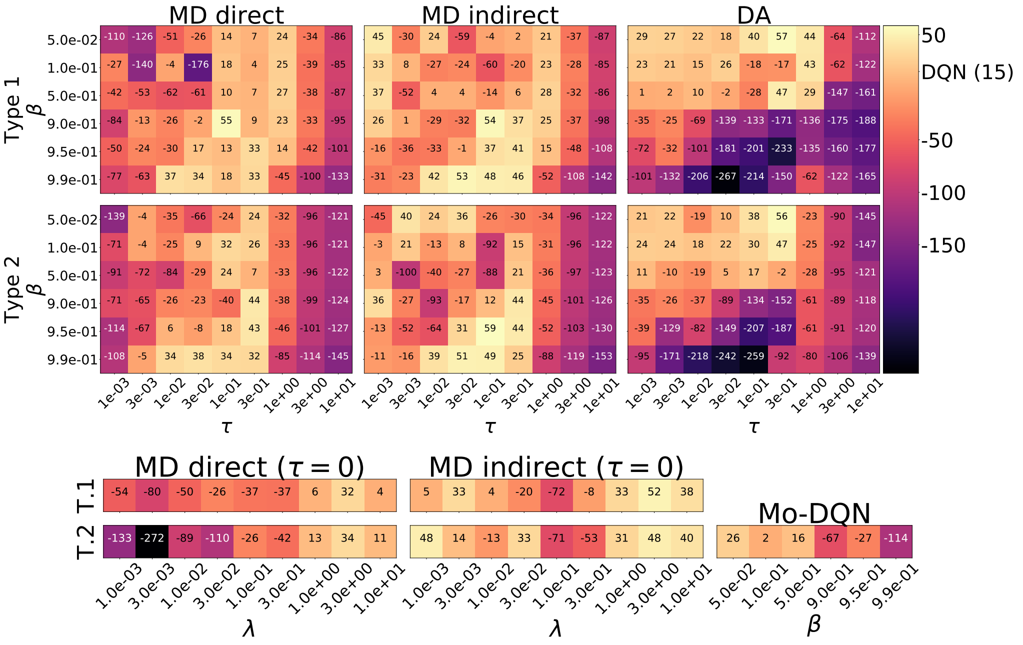

Figure 9. Lunar Lander with complete values.

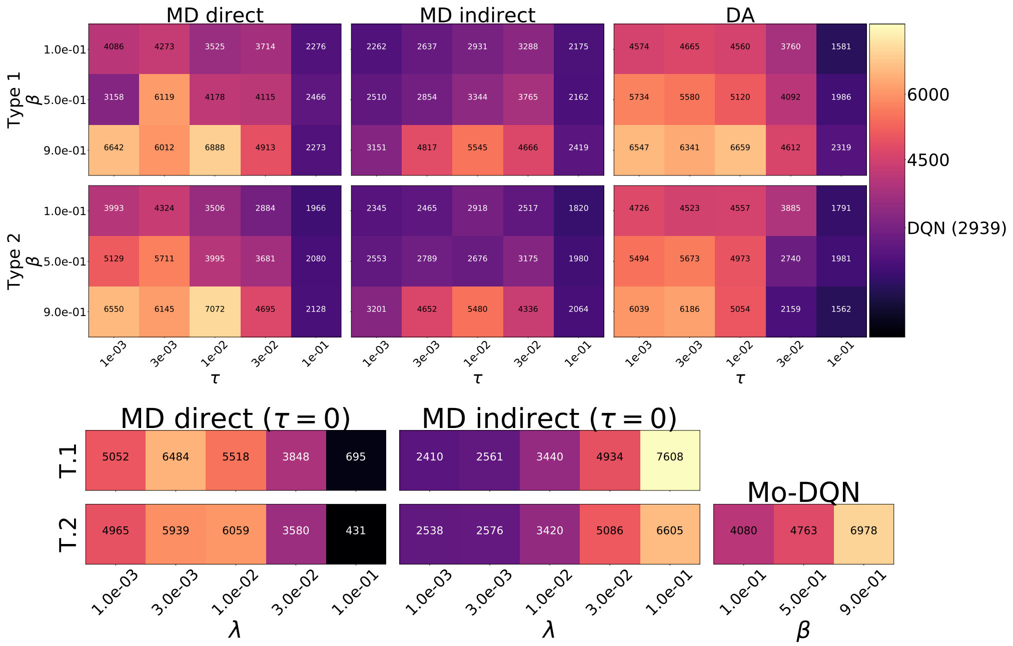

Figure 10. Asterix with complete values.

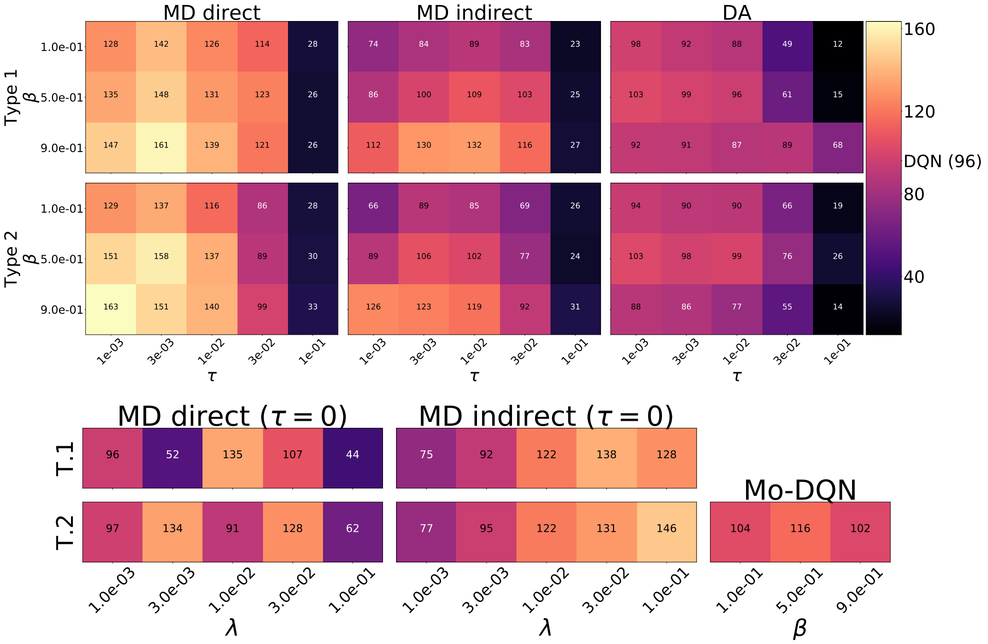

Figure 11. Breakout with complete values.

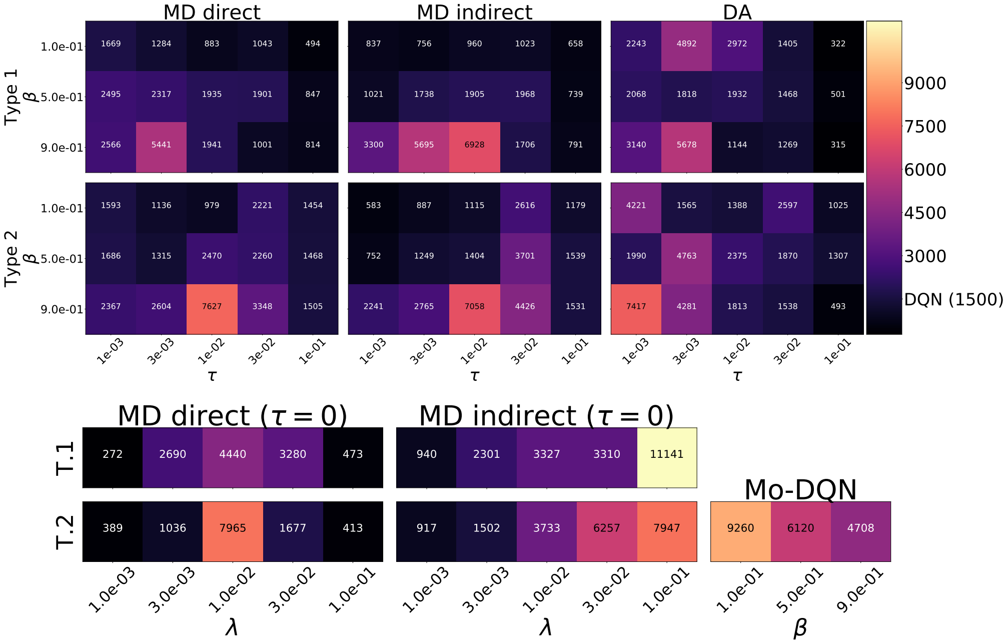

Figure 12. Seaquest with complete values.

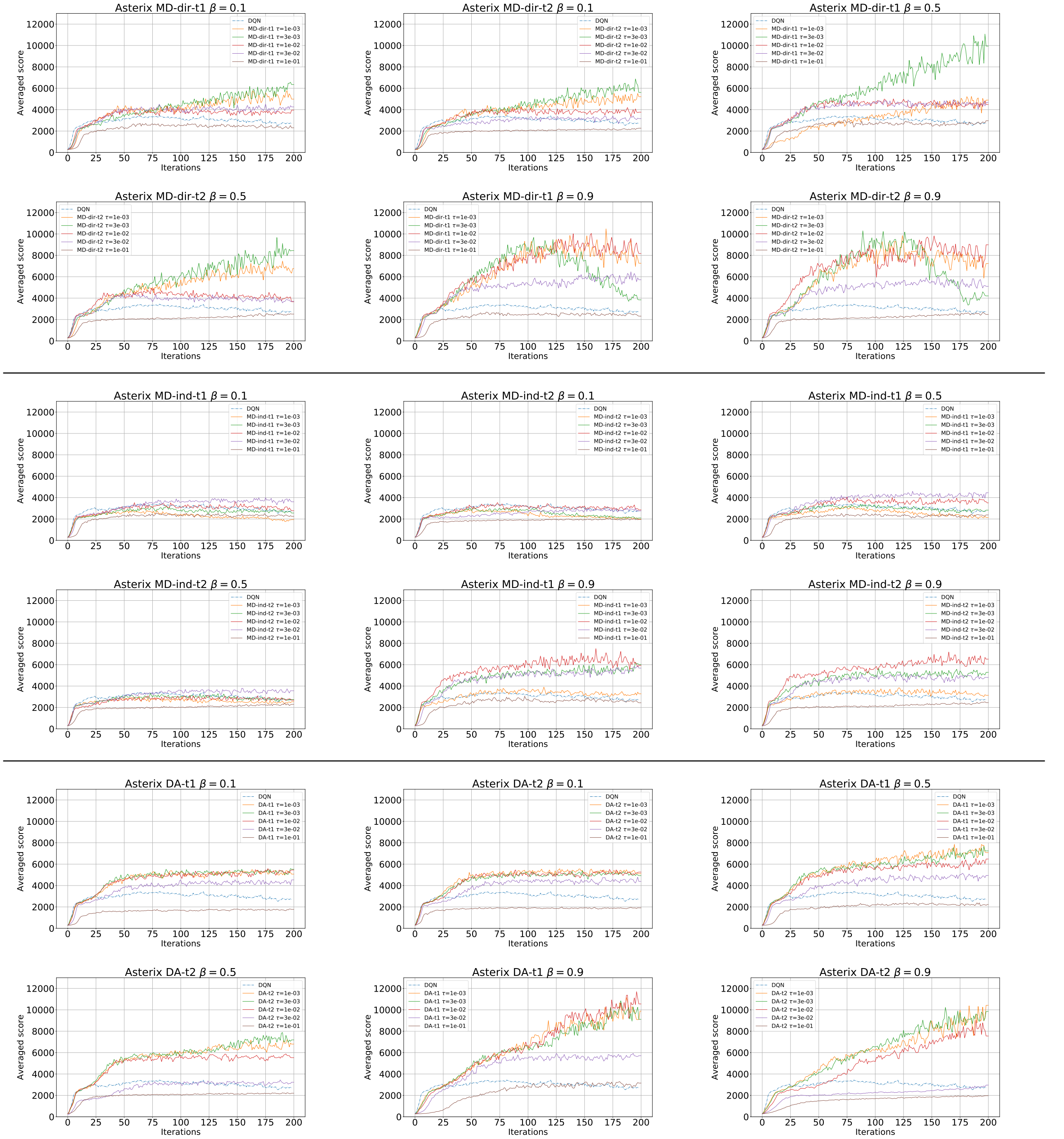

Figure 13. All averaged training scores of MD-dir (top), MD-ind (middle) and DA (bottom), types 1 and 2, on Asterix, for several values of β and τ. Each plot corresponds to one value of β (in the titles). In each plot, a curve corresponds to a value of τ: $1e-3$ (orange), $3e-3$ (green), $1e-02$ (red), $3e-2$ (blue), $1e-1$ (brown). The blue dotted line is DQN.

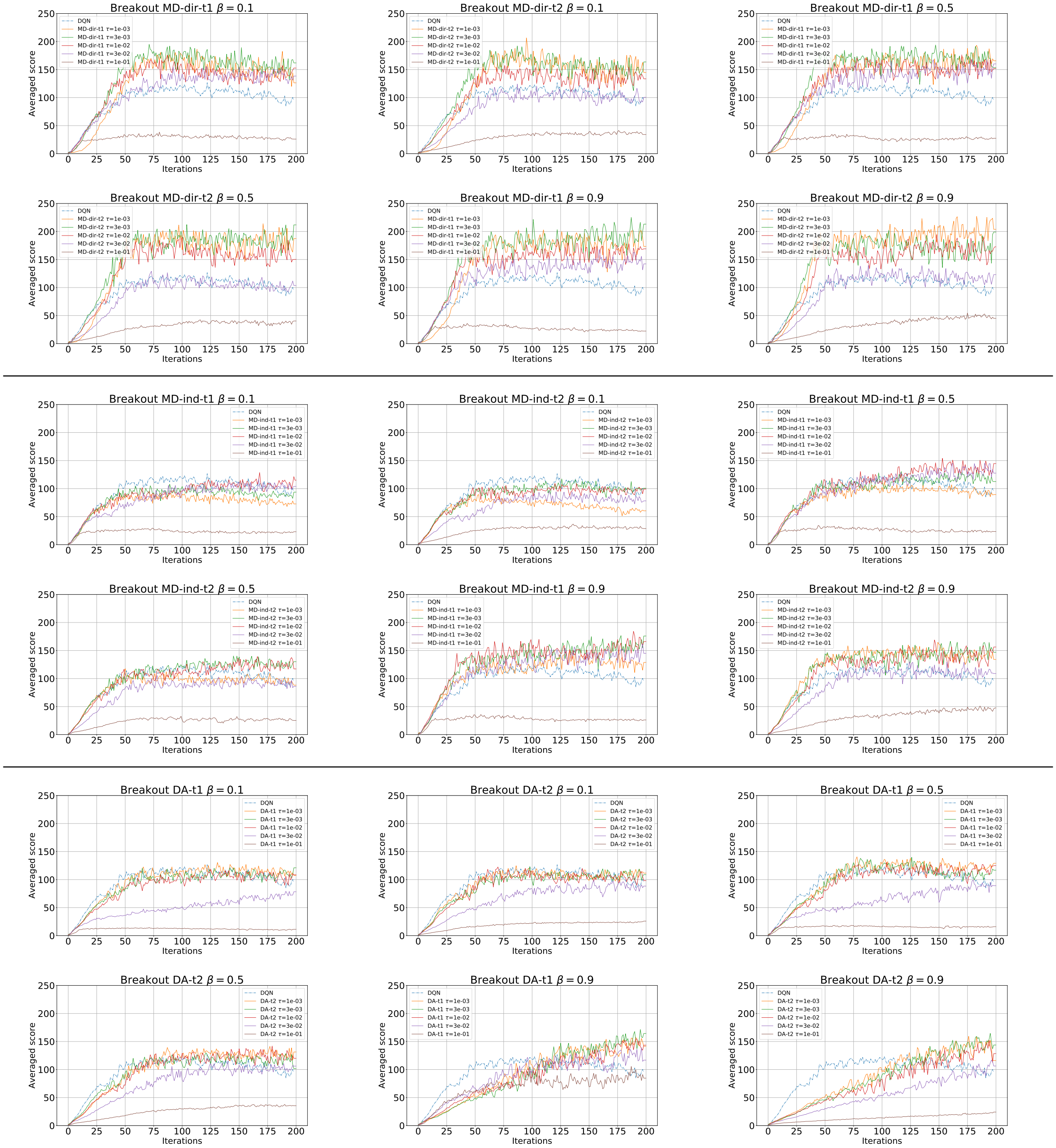

Figure 14. All averaged training scores of MD-dir (top), MD-ind (middle) and DA (bottom), types 1 and 2, on Breakout, for several values of β and τ. Each plot corresponds to one value of β (in the titles). In each plot, a curve corresponds to a value of τ: $1e-3$ (orange), $3e-3$ (green), $1e-02$ (red), $3e-2$ (blue), $1e-1$ (brown). The blue dotted line is DQN.

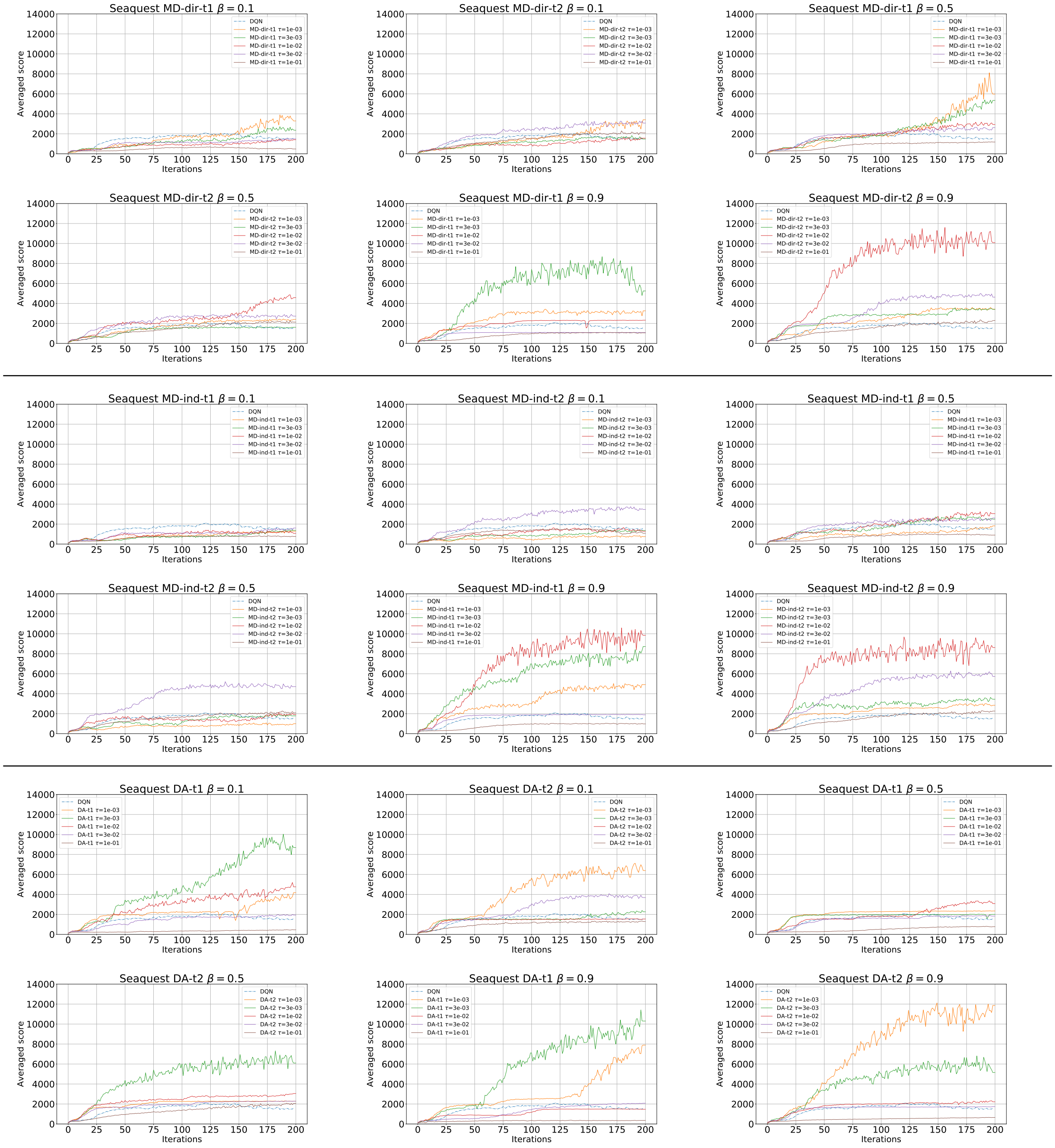

Figure 15. All averaged training scores of MD-dir (top), MD-ind (middle) and DA (bottom), types 1 and 2, on Seaquest, for several values of β and τ. Each plot corresponds to one value of β (in the titles). In each plot, a curve corresponds to a value of τ: $1e-3$ (orange), $3e-3$ (green), $1e-02$ (red), $3e-2$ (blue), $1e-1$ (brown). The blue dotted line is DQN.

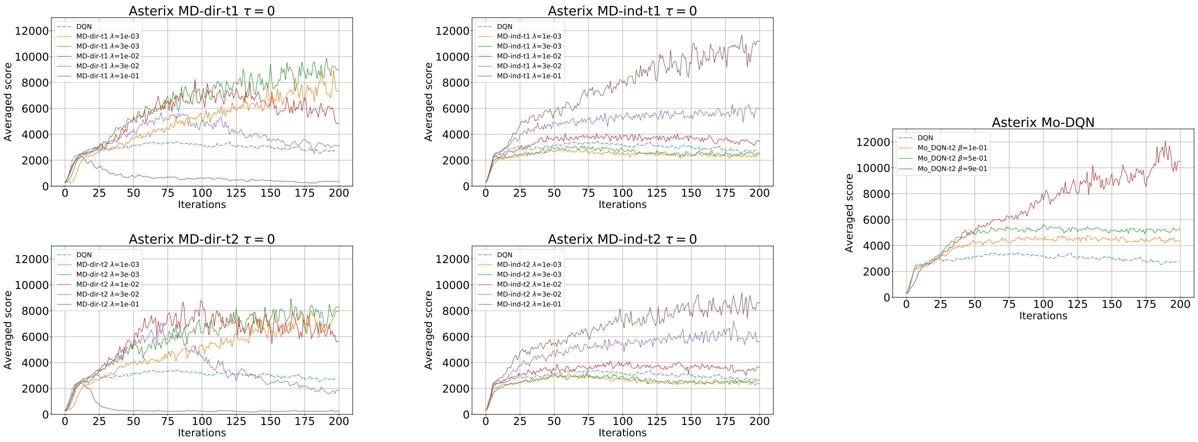

Figure 16. All averaged training scores of limit cases on Asterix, for several values of β and λ. In each plot, a curve corresponds to a value of λ for MD-ind and MD-dir, and to a value of β for Mo-DQN. The blue dotted line is DQN.

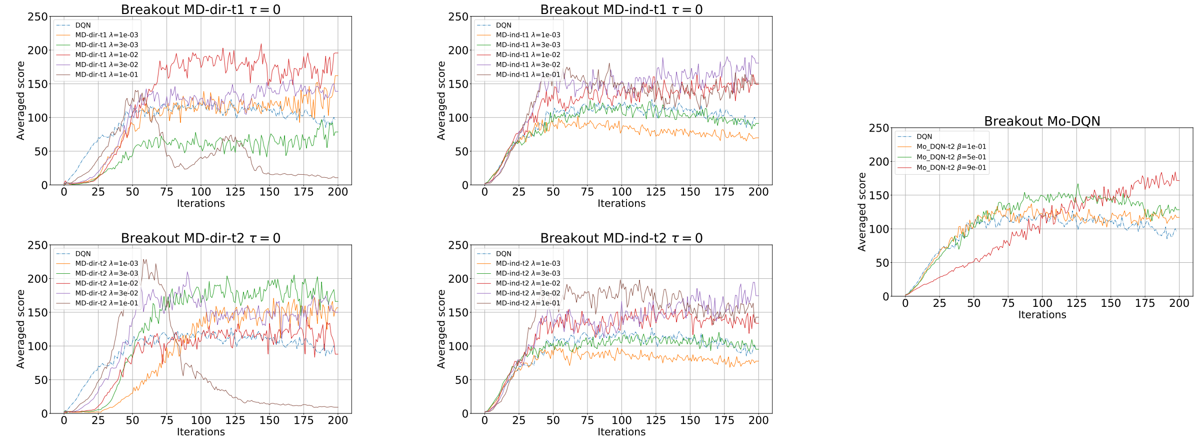

Figure 17. All averaged training scores of limit cases on Breakout, for several values of β and λ. In each plot, a curve corresponds to a value of λ for MD-ind and MD-dir, and to a value of β for Mo-DQN. The blue dotted line is DQN.

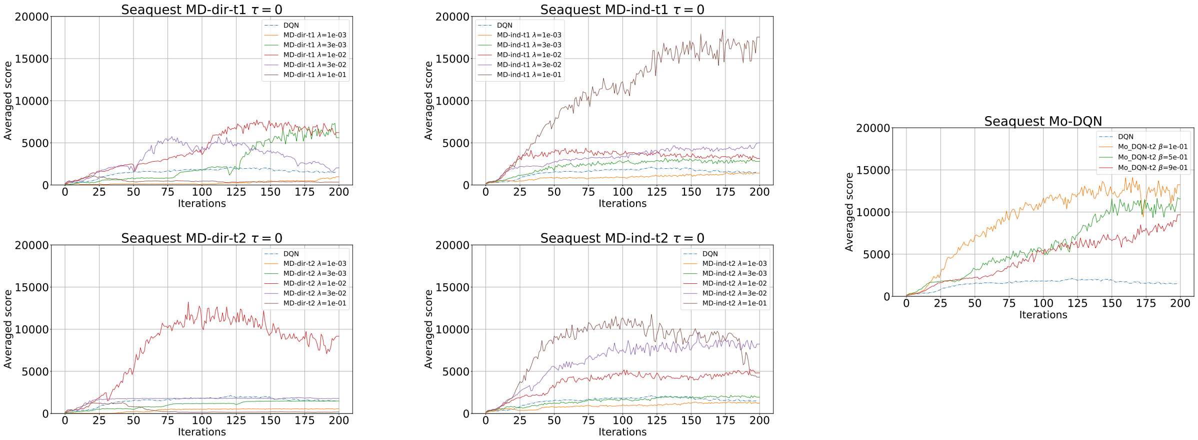

Figure 18. All averaged training scores of limit cases on Seaquest, for several values of β and λ. In each plot, a curve corresponds to a value of λ for MD-ind and MD-dir, and to a value of β for Mo-DQN. The blue dotted line is DQN.