DENSITY ESTIMATION FOR CONSERVATIVE Q-LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Batch Reinforcement Learning algorithms aim at learning the best policy from a batch of data without interacting with the environment. Within this setting, one difficulty is to correctly assess the value of state-action pairs that are far from the dataset. Indeed, the lack of information may provoke an overestimation of the value function, leading to non-desirable behaviours. A compromise between enhancing the behaviour policy's performance and staying close to it must be found. To alleviate this issue, most existing approaches introduce a regularization term to favor state-action pairs from the dataset. In this paper, we refine this idea by estimating the density of these state-action pairs to distinguish neighbourhoods. The resulting regularization guides the policy toward meaningful unseen regions, improving the learning process. We hence introduce Density Conservative Q-Learning (D-CQL), a batch-RL algorithm with strong theoretical guarantees that carefully penalizes the value function based on the amount of information collected in the state-action space. The performance of our approach is outlined on many classical benchmark in batch-RL.

1 INTRODUCTION

Transposing the recent successes of Reinforcement Learning (RL) such as recommendation systems (Rojanavasu et al., 2005; Zheng et al., 2018), video games (Mnih et al., 2013), go (Silver et al., 2017) to real-world systems is not possible without facing many challenges (Dulac-Arnold et al., 2021). One of those challenges is that in many real-world applications, direct access to the system can be limited, and sometimes even forbidden. This can be due to various reasons, an important one being that a learning controller may incorrectly assess the implications of its actions and damage the system. Batch, or offline, Reinforcement Learning (Lange et al., 2012; Levine et al., 2020) provides a framework to address those RL problems when no interaction with the system is allowed. In place, the learner is given a dataset collected under a (possibly unknown) *behavioural* policy and has to derive the most efficient policy out of this dataset.

One of the main problematic encountered within this setting is the value-function over-estimation problem (Fujimoto et al., 2019; Levine et al., 2020; Kumar et al., 2019a; Wu et al., 2019). Indeed, when the dataset only covers a small subset of the state-action space, the agent typically wrongly extrapolates the value functions related to pairs far from the dataset, denoted as *Out-Of-Distribution* (OOD). This error is then used as a target in the learning process, leading to highly increasing estimates which may lead to a disastrous learned policy. This problem has been extensively studied in the traditional online Reinforcement Learning setting (Sutton & Barto, 1998). However in this case, this issue is quite naturally alleviated since when a value function becomes over-optimistic, it will drive the agent to visit the related state-action pair. Therefore, the agent will have the chance to directly check the consequences of such pairs and its estimation can be corrected. Some modifications, mostly relying on reducing these errors with *Ensemble Learning* (van Hasselt, 2010; van Hasselt et al., 2016; Anschel et al., 2017; Lee et al., 2021b), can also be used to enhance learning.

In Batch RL, since no inspection can be done to investigate the accuracy of the estimates, those methods cannot be used and hence classical deep *Off-Policy* methods dramatically fail in this setting (Fujimoto et al., 2019; Levine et al., 2020). Additional parts must be introduced to build robust and efficient agents. An important family of state-of-the art algorithms addressing this problem focus on constraining the learned policy to stay close to the dataset either by minimizing its distance to the behavioral policy (Siegel et al., 2020a; Wu et al., 2019; Kumar et al., 2019b) or by penalizing unseen state-action pairs (Luo et al., 2019; Kumar et al., 2020; Yu et al., 2021). Especially, Kumar et al. (2020) propose to optimize a lower bound on the value functions and use this lower bound as a proxy in the policy optimization process. This lower bound should be tight when state-action pairs are contained in the dataset and loosened otherwise. Following this purpose, they proposed Conservative Q-Learning (CQL) that introduces a penalization to under-estimate value functions associated to OOD actions. On top of driving the agent to stay close to the dataset transitions, this method gives a chance to standard *Off-Policy* learning as estimates will have a reasonable scale during learning. Nevertheless, given the lack of information regarding the *behavioural policy*, their regularization remains abrupt and might be problematic in practice. First, the resulting lower bound is loosened with the recommended distributions. Second, their approach does not consider actions' neighbourhoods and could therefore over-penalize interesting action space's regions.

In this paper, we tackle these problems by introducing a novel penalization based on an estimation of the dataset's probability density function instead of the behavioural policy. Besides being easier to learn that the behavioural policy, this density is able to provide information on which regions are near the dataset and thus safe to learn from, and which are the ones far from it and prone to over-estimation errors. On top of instigating those relevant information, we show that this new regularization leads to a more appropriate lower bound on the value function. We show how to integrate this penalization in a CQL-like algorithm and derive a novel algorithm called *Density Conservative Q-Learning* (D-CQL). We finally investigate empirically the relevance of our approach.

2 Preliminaries and Motivations

2.1 Notations and Policy Iteration

The agent-environment framework is modeled as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, r, P, \gamma)$ defined by a state space $\mathcal{S}$, an action space $\mathcal{A}$, a transition kernel $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, a reward function $r : \mathcal{S} \times \mathcal{A} \to [R_{min}, R_{max}]$ and a discount factor $\gamma \in]0, 1[$. A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ is a decision rule mapping a state over a distribution of actions. The value of a policy is measured through the value function $V^\pi(s) = \mathbb{E}_P\left[\sum_{t=0} \gamma^t r(s_t, a_t) \mid s_0 = s,\ a_t \sim \pi(\cdot|s_t)\right]$ and its associated Q-value function $Q^\pi(s, a) = \mathbb{E}_P\left[\sum_{t=0} \gamma^t r(s_t, a_t) \mid s_0 = s,\ a_0 = a,\ a_t \sim \pi(\cdot|s_t)\ \forall t \geq 1\right]$. The goal is to find the policy maximizing these value functions.

Let r, Q be matrices associated to all state-action pairs and P^π be the transition matrix induced by the policy π: $P^\pi(s, s') = \mathbb{E}_{a\sim\pi(\cdot|s)} P(s'|s, a)$. Value functions can be learned by iterating the Bellman operator defined for the Q-function as $\mathcal{B}^\pi Q = r + \gamma P^\pi Q$ where $P^\pi Q(s, a) = \mathbb{E}_{s'\sim P(\cdot|s,a), a'\sim\pi(\cdot|s)}\left[Q(s', a')\right]$. This operator is a γ-contraction (Puterman, 1994), hence having a unique fixed point Q^π. Many classic recent algorithms rely on the *Policy Iteration* scheme (Lillicrap et al., 2016; Fujimoto et al., 2018; Haarnoja et al., 2018), where the agent alternates between *Policy Evaluation* (PE) with the computation of Q^π and *Policy Improvement* (PI) by maximizing the learned Q-values

$$Q^{k+1} \leftarrow \arg\min_{Q} \mathcal{B}^{\pi^k} Q^k - Q, \qquad \text{(policy evaluation)}$$

$$\pi^{k+1} \leftarrow \arg\max_{\pi} \mathbb{E}_{a\sim\pi}\left[Q^{k+1}(\cdot, a)\right]. \qquad \text{(policy improvement)}$$

This iterative process converges towards the optimal policy (Sutton & Barto, 1998; Santos & Rust, 2004). Since no knowledge is assumed on the environment, these steps are commonly solved using samples from a dataset $\mathcal{D} = \{(s_i, a_i, r_i)_{i=1}^N\}$. The expectation under $P(\cdot|s, a)$ is now estimated using sample and leads to the *empirical* Bellman operator $\hat{\mathcal{B}}$. We slightly abuse notations and consider $\mathcal{D}$ is also a distribution on $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$. The procedure is now under the dataset expectation

$$Q^{k+1} \leftarrow \arg\min_{Q} \mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(\hat{\mathcal{B}}^{\pi^k} Q^k(s, a) - Q(s, a)\right)^2\right], \qquad \text{(approximate policy evaluation)}$$

$$\pi^{k+1} \leftarrow \arg\max_{\pi} \mathbb{E}_{s\sim\mathcal{D}, a\sim\pi}\left[Q^{k+1}(s, a)\right]. \qquad \text{(approximate policy improvement)}$$

The Q-values and the policy π are commonly estimated using Neural Networks that are trained with gradient optimization methods. In the approximate policy evaluation step, an important aspect is the expectation of actions from $\pi(\cdot|s')$ in the bootstrapped target. This is where OOD actions may appear and become a decisive topic: extrapolation errors emerge, are then back-propagated to eventually lead to highly over-optimistic estimates. In the *Online* setting, the dataset is constantly updated with samples gathered with the learned policy, coping with this problem. Note there are no OOD states in the Bellman update as the empirical operator only depends on a seen state s'. They might appear on model-based algorithms (Yu et al., 2021) but are for now out of the scope of model-free algorithms. An interesting future direction would be to consider them as well.

In this work, we focus on the Batch setting where the agent cannot interact with the environment. The agent dataset is now fixed and has been gathered with an unknown *behavioural policy* π_β. No knowledge is assumed on π_β as the dataset can come from different sources: optimal control, human or a mixture of them (Fu et al., 2020; Gülçehre et al., 2020). Actions that belong to the dataset distribution given the state s are denoted as In-Distribution (ID) and the ones far from it are Out-Of-Distribution (OOD). A major challenge is to find a good trade-off between staying close to the dataset to avoid extrapolation errors and taking some liberty to overcome the suboptimality of the behaviour policy.

2.2 Conservative Q-Learning

This approach addresses the Q-values over-estimation problem exacerbated in Batch RL. As a reminder, the *Policy Evaluation* step relies on a bootstrapped target that back-propagates any extrapolation error during learning. This error may be of high importance when the consequences of a given state-action pair are unknown, and could lead to highly optimistic estimates. This problem is intensified in Batch RL. First, the dataset often describes a small subset of the state-action space $\mathcal{S} \times \mathcal{A}$ thus OOD actions are very likely to appear. Second, the agent never gets the chance the visit the state-action pairs related to high Q-values and cannot correct its possibly wrong estimations.

At this step, Kumar et al. (2020) propose to penalize the actions not described by the dataset while keeping intact actions from the dataset with proper information. This greatly stabilizes learning: the agent does not have to deal with drastically high estimates and therefore gives the agent a chance to learn accurate Q-values thanks to the empirical Bellman operator. It also implicitly drives the agent to favor state-action pairs described in the dataset. This penalization translates into minimizing the Q-values on an arbitrary distribution μ (*e.g.* uniform) and maximizing them on π_β formally expressed as follows

$$\begin{aligned}\hat{Q}^{k+1} \leftarrow \arg\min_{Q} \alpha\, \mathbb{E}_{s\sim\mathcal{D}} \left[\left(\mathbb{E}_{a\sim\mu(\cdot|s)}\left[Q(s,a)\right] - \mathbb{E}_{a\sim\pi_\beta(\cdot|s)}\left[Q(s,a)\right]\right)\right] \\ + \mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(\hat{\mathcal{B}}^{\pi_k}\hat{Q}^k(s,a) - Q(s,a)\right)^2\right].\end{aligned} \tag{1}$$

Assuming the absence of sampling errors and $\operatorname{supp}\mu \subset \operatorname{supp}\pi_\beta$, this update provides a lower-bound on the expected Q-values on the distribution μ: $\mathbb{E}_{a\sim\mu}\left[Q^{CQL}(s,a)\right] \leq \mathbb{E}_{a\sim\mu}\left[Q^{\pi}(s,a)\right]$ for all $s \in \mathcal{S}$ with $\alpha > 0$. When μ is the current policy π, the lower bound is in the value function, that is $V^{CQL}(s) \leq V^{\pi}(s)$. This property may be highly desirable on applications where the agent must check if the current value function is above a certain threshold such as Constrained Policy Optimization (Achiam et al., 2017) and Conservative Exploration (Garcelon et al., 2020). However, for the common setting where μ is a uniform distribution over the action space, the lower bound is loosened. While a certain control in the Q-values is kept, both the value and point-wise inequalities are lost.

Another point of attention is the maximization term relying on the knowledge of the behavioral policy. In the general setting, it is out of reach as the dataset may be gathered in various ways. Kumar et al. (2020) propose to use the empirical dataset distribution $\hat{\pi}_\beta(a|s) = \frac{\sum_{s',a'\in\mathcal{D}} \mathbb{1}[s'=s,a'=a]}{\sum_{s'\in\mathcal{D}} \mathbb{1}[s'=s]}$, that may become a Dirac over the dataset actions especially when the action space $\mathcal{A}$ is continuous. Not only does this breaks the assumption $\operatorname{supp}\mu \subset \operatorname{supp}\hat{\pi}_\beta$ (as in practice $\operatorname{supp}\pi_\beta \not\subset \operatorname{supp}\hat{\pi}_\beta$) , it also might be problematic on the penalization itself: the distance between the selected actions and the dataset is no longer considered. In other words, actions that are not appearing in the dataset will be equally penalized no matter their closeness with the dataset $\mathcal{D}$. Thus, the regularization pushes the agent to reproduce the *behavioural policy*, not enhance it.

3 Density Conservative Q-Learning

In this section, we introduce Density Conservative Q-Learning (D-CQL), our algorithm circumventing these issues.

3.1 A new weighting scheme on the Out-Of-Distribution actions

The CQL's penalization pushes the agent's learned policy to stay too close to the behavioural policy as it penalizes any action that does not belong to the dataset and thereby prevents the agent from investigating on potential relevant areas. This problem is of high interest, *e.g.* when the dataset is gathered with a distinctly sub-optimal policy but more efficient actions exist in the neighborhood of the sub-optimal ones. The chances of finding a better policy would be reduced as any action that does not belong to the dataset will be equally penalized. Nevertheless, in view of the fixed nature of the batch setting and the high sensibility of the Bellman update, the agent should still be cautious with respect to which regions it should focus on.

Bearing these complications in mind, we propose to refine the original regularization by appropriately weight the actions according to their *uncertainty*. To do so, we quantify the distance between the dataset and (sampled) penalized actions to soften the regularization. It emancipates the agent from a hard penalization that constrains its policy to be extremely close to the behavioural policy. It also carefully guides the agent towards potential meaningful dataset neighbourhoods giving it the opportunity to learn better policies.

A first idea is to introduce a regularization depending on the OOD actions distribution given a state s denoted $\kappa(\cdot|s)$. This would lead to the following update:

$$\hat{Q}^{k+1} \leftarrow \arg\min_{Q} \alpha\, \mathbb{E}_{s\sim\mathcal{D},\, a\sim\kappa(\cdot|s)}\left[Q(s,a)\right] + \mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(\hat{\mathcal{B}}^{\pi_k} Q^k(s,a) - Q(s,a)\right)^2\right]. \tag{2}$$

With the true distribution κ on the OOD actions, this update would rightfully consider the uncertainty of any OOD action while focusing on ID actions in the Bellman update. However, considering we only have access to samples where the distribution is not supposed to be, estimating such density is nearly intractable. For instance, we observed that maximizing the entropy of the distribution while performing *minimum likelihood* on the dataset samples leads to poor results

Instead, we propose a regularization based on the dataset's state-action pairs density denoted $\rho_\beta : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}_+$. This density describes how *likely* a state-action pair belongs to the dataset and by extension, how *unlikely* and therefore *uncertain* it is. Various techniques can be used to learn such a distribution and we deal with its estimation in the next section. Then, we can construct a value $\zeta(a|s)$[1] that quantify how uncertain an action a, given s, is w.r.t. $\mathcal{D}$. ζ is integrated it as follows:

$$\hat{Q}^{k+1} \leftarrow \arg\min_{Q} \alpha\, \mathbb{E}_{s\sim\mathcal{D}, a\sim\mu(\cdot|s)}\left[\zeta(a|s) Q(s,a)\right] + \mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(\hat{\mathcal{B}}^{\pi_k} Q^k(s,a) - Q(s,a)\right)^2\right]. \tag{3}$$

Here, the resulting penalization is, when μ is uniform, equivalent to equation 2 up to a normalizing term. Thus, ζ is able to control how much an action a chosen under μ should be penalized and therefore satisfies our goal. Note that CQL corresponds to a particular version of D-CQL where $\zeta(a|s) = 1 - \delta_{a_\mathcal{D}}(a)$, where $\delta_{a_\mathcal{D}}$ denotes the Dirac over the dataset action $a_\mathcal{D}$.

Theorem 1 *For any μ satisfying $\operatorname{supp}\mu \subset \operatorname{supp}\pi_\beta$ and assuming the absence of sampling errors, then the Q-values obtained by iterating equation 3 are*

$$\forall s \in \mathcal{D}, a \in \operatorname{supp}\pi_\beta(\cdot|s),\ Q^{\text{D-CQL}}(s,a) = Q^\pi(s,a) - \alpha\left[(I - \gamma P^\pi)^{-1}\frac{\mu\,\zeta}{\pi_\beta}\right](a|s), \tag{4}$$

and $\alpha > 0$ leads to a point-wise lower bound on the true Q-values that becomes tight whenever $\zeta(a|s) = 0$.

[1] ζ is a weighting scheme, not a density. However, we slightly abuse notations by taking a conditional notation of ζ on the state s to highlight that is should only depends on the action, not on the state.

On top of its intuitive advantages, the new regularization leads to a strong point-wise lower bound on the Q-values that is tight on ID actions with a well-chosen ζ.

3.2 Choice of the weighting scheme

First, the weighting scheme should capture the uncertainty of an action a given a state s. Second, since we focus on OOD actions given a state, ζ must not yield any information on how likely is a state s to appear in the dataset. Third, we argue having a weighting scheme bounded by 0 and 1 allows a better analysis in practice. In principle, any metric satisfying these characteristics can be used. One that provided good results in practice is

$$\zeta_\nu(a|s) = \max\left(\nu, 1 - \frac{\rho_\beta(s,a)}{\rho_\beta(s,a_\mathcal{D})}\right), \tag{5}$$

with $a_\mathcal{D}$ the action present in the dataset related to the state s.

The ratio $\frac{\rho_\beta(s,a)}{\rho_\beta(s,a_\mathcal{D})}$ frees ζ_ν from the dependency on the state s, and the linear decay lead to good results but practitioners can choose another type of decay (polynomial, exponential, ...). We refer to Appendix B.2 for a validation of this metric.

The hyper-parameter ν On the one hand, we need $\nu = 0$ for the lower bound of Theorem 1 to hold when $\zeta(a|s) = 0$. Indeed, when $\nu = 0$ the Q-values associated to ID actions are not penalized which follows the global intuition. Furthermore, since OOD actions do not appear in the Bellman loss term of Equation (3), the minimum of their Q-values is unbounded which eventually affects the performances of the algorithm as verified empirically in Section 6. On the other hand, when $\nu > 0$ the regularization term will also focus on ID actions and this minimization will be counterbalanced by the Bellman loss that tends to maximize their Q-values. In other words, the lower ν, the greater the difference between OOD and ID Q-values to balance the objective. In practice, it is common to have knowledge on the quality of the behavioural policy and on the considered system, which can help setting the right ν. Indeed, if we know that an expert has gathered the dataset, we will want to stay close to it set $\nu \approx 0$. Otherwise, if we know that the exploration has not been good and that there is no risk of harming the system, we can choose a bigger ν to allow the controller to consider uncertain actions. Another intuition behind setting $\nu \neq 0$ is to penalize in the same way ID actions and unseen actions that are relatively close to the dataset. In fact, the greater ν the larger the region exploitable by the agent[2]. While the tightness of the lower bound is loosened, the rank of the penalization is kept between OOD and ID actions. In Section 6, we conduct a detailed analysis of the effects of the OOD penalization with varying ν.

4 Practical algorithm

4.1 Density Estimation

Density Conservative Q-Learning regularization relies on density estimation. In this work, we propose to use a highly expressive technique: *Normalizing Flows* (Rezende & Mohamed, 2015; Kobyzev et al., 2020). This choice was fueled with their successes on high dimensional distributions (Papamakarios et al., 2019) and their elegance. Indeed, they are known to generalize and scale well from complex distributions (Winkler et al., 2019) - including images (Kobyzev et al., 2020) - while keeping a tractable learning.

Normalizing flow are based on the change of variable for a diffeomorphism $f : X \to Y$ and two densities defined on X and Y

$$\rho_X(x) = \rho_Y\left(f(x)\right)\det\left(\mathrm{Jac}\, f(x)\right). \tag{6}$$

We then learn a diffeomorphism f that maps the unknown distribution on X to a known distribution on Y, for example a multivariate normal distribution so that the factor $\rho_Y f(X)$ is efficiently calculable. The main difficult is remaining factor: the computation of the determinant of the jacobian. In order to leverage this issue, most approach rely on finding a series of mappings $f_1, \ldots, f_n$ such that

[2] $\forall s \in \mathcal{D}, \forall \nu_0 < \nu_1, a_\mathcal{D} \subset \{a \in \mathcal{A} : \zeta_{\nu_0}(a|s) = \nu_0\} \subset \{a \in \mathcal{A} : \zeta_{\nu_1}(a|s) = \nu_1\}$

$f = f_1 \circ \ldots \circ f_n$ is complex enough to handle arbitrarily complex distribution. As long as each f_i determinant of the jacobian is tractable, so if f by chain-rule.

In this work, we use a direct implementation of Real-NVP (Dinh et al., 2017). Each layer randomly selects half of the features on which it applies a neural network. The computation is made in a triangular manner in respect of the variable so that the jacobian is lower triangular, and its diagonal is easily accessible. See Appendix B.1 for further details about Real-NVP. The whole network is then trained using gradient ascent on the log-likelihood.

This algorithm allows us to estimate the densities of the trajectory datasets in a efficient and accurate manner. Note that f^{-1} is also tractable and may act as a generative model.

For this work, we tried other approach such as fitting a multivariate normal distribution and the more refined autoregressive (Uria et al., 2013). We present a study of the impact of these different estimators for Density Conservative Q-Learning in Appendix B.1.

4.2 Density Conservative Q-Learning

Following the ideas expressed in (Kumar et al., 2020, Appendix A), we add an entropic regularization to Equation (3) that leads to the following update for Density Conservative Q-Learning

$$\begin{aligned}\hat{Q}^{k+1} \leftarrow \arg\min_{Q} \alpha\, \mathbb{E}_{s\sim\mathcal{D}} &\left[\log \sum_{a\in\mathcal{A}} \exp\left(\zeta_\nu(a|s) Q(s,a)\right)\right] \\ &+ \frac{1}{2}\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(\mathcal{B}^{\pi_k}\hat{Q}^k(s,a) - Q(s,a)\right)^2\right]. \end{aligned} \tag{7}$$

The `log-sum-exp` operator can be seen as a soft differential maximum operator. Intuitively, Equation (7) will focus on the highest Q-values at a threshold defined by the weighting scheme ζ_ν. This expression exacerbate D-CQL's behaviour described in Section 3.2 when ν is close to zero. Same conclusion holds, the lower ν, the greater the difference between OOD and ID Q-values to balance the objective.

However, the `log-sum-exp` operator cannot be computed as such when the action set is too large or continuous. We resort to *Importance Sampling* (IS) on two different distributions. At each update and for each state s, we sample two sets of M actions: the first one from a uniform distribution over the action space and the second one following the current policy $\pi(\cdot|s)$. Sampling from the uniform allows a certain control over the action space. The learned policy will get close the π_β, so sampling through this distribution will produce ID actions. Thus, such ID actions will eventually be penalized preventing to the unbounded solution discussed in 3.2 from happening.

Finally, we describe Density Conservative Q-Learning or D-CQL in Algorithm 1.

Algorithm 1 D-CQL

```
Learn $\rho_\beta$ with behavioural cloning
Initialize $Q_\theta$ and $\pi_\Phi$
**for** $k \in (1, \ldots)$ **do**
    Sample a batch $\mathcal{B}$ from $\mathcal{D}$
    For each state $s \in \mathcal{B}$ sample $M$ actions from $\mathcal{U}(\mathcal{A})$
    For each state $s \in \mathcal{B}$ sample $M$ actions from $\pi(\cdot|s)$
    Update $Q_\theta$ with gradient descent on Equation 7 using Importance Sampling
    Update $\pi_\Phi$ with gradient ascent on Equation policy improvement
**end for**
```

4.3 Link with KL regularization

We further fuel the relevance of D-CQL penalization with a KL-penalization point of vue (Kumar et al., 2019a; Wu et al., 2019; Kostrikov et al., 2021) that underlines the advantages of adding the weighting scheme ζ_ν. Let π_{DB} be the *Density-regularized Boltzmann* policy

$$\pi_{DB}(a|s) = \frac{\exp\left(\zeta_\nu(a|s)\ Q(s,a)\right)}{\sum_a \exp\left(\zeta(a|s)\ Q(s,a)\right)}\,. \tag{8}$$

This policy is derived from the Boltzmann policy, commonly used to reduce the over-estimation problem (Pan et al., 2020) and provides a good trade-off in the exploration-exploitation dilemma (Cesa-Bianchi et al., 2017). The introduction of ζ_ν keeps these benefits while adding the uncertainties with respect to the fixed dataset that could lead to high exploration capacities. Our regularization can be recovered with a KL-divergence D_{KL} between the behavioural and this policy

$$\begin{aligned} &D_{KL}(\pi_\beta(\cdot|s) \parallel \pi_{DB}(\cdot|s)) = \\ &\qquad \mathbb{E}_{a\sim\pi_\beta(\cdot|s)}\left[\log \sum_a \exp\left(\zeta_\nu(a|s)\ Q(s,a)\right) - \zeta_\nu(a|s)\ Q(s,a) + \log \pi_\beta(a|s)\right] \end{aligned} \tag{9}$$

assuming $\zeta_\nu(a|s) \to 0$ when $a \sim \pi_\beta(\cdot|s)$ and considering that the last term is constant with respect to Q.

When Equation (9) is minimized, π_{DB} is steered to get as close as possible to π_β. As a consequence, the Q-values must be high when $\pi_\beta(a|s)$ is high. The role of ζ_ν is also highlighted: a high $\pi_\beta(a|s)$ implies a low $\zeta_\nu(a|s)$ that in turns further augments the Q-values in this area.

5 Related works

Learning a policy from a fixed dataset has been of high interest in Reinforcement Learning. A first way to address this problem is using Behavioural Cloning. When the dataset is complete and comes from an expert policy, it may achieve great successes as shown in special cases of *autonomous driving* (Pomerleau, 1988) and *flying* (Sammut et al., 1992). However, when the agent finds itself in a situation not described by the dataset, it may choose catastrophic actions and its performance can quickly degrade (Codevilla et al., 2019). Imitation Learning (Hussein et al., 2017; Ho & Ermon, 2016) aims to counter this problematic but still requires an expert policy. In real-world applications, the dataset may come from sub-optimal policies and instead of reproducing the behavioural policy, the goal is to extract a *better* one. This is the promise of Batch RL. Recently, many batch algorithms have been introduced, all relying on making sure the learned policy stays close to the behavioural one, avoiding the presence of Out-Of-Distribution (OOD) actions (Levine et al., 2020). It can take several forms.

Many recent algorithms cope with the distributional shift by directly avoiding the selection of OOD actions by the policy, preventing the agent to learn from highly extrapolated values. BCQ (Fujimoto et al., 2019) explicitly parametrize the policy to stay within a parametric ball around the behavioural policy. Similarly, SPIBB (Laroche et al., 2019) restricts the support of the learned policy to remain in the support of the behavioural policy. Other methods - such as BRAC (Kumar et al., 2019a) and BEAR (Wu et al., 2019) - modify the RL objective with a measure of closeness between the learned and the behavioral policy. AWR (Peng et al., 2019) or ABM (Siegel et al., 2020b) build a Trust Region around the behavioural policy. Rather than a constraint between the policies, AlgaeDICE (Nachum et al., 2019) and OptiDICE (Lee et al., 2021a) focused on the *state-action stationary distributions* which provided competitive results. While successful on a variety of tasks, they do not explicitly counter the over-estimation problem that may dominate and prevent learning (Kostrikov et al., 2021). Another line of work is to enhance the robustness of the agent while leaving the objective unchanged. REM (Agarwal et al., 2020) chooses to not modify the objectives, but uses Ensemble Learning to build a random convex mixtures of targets stabilizing the updates. In practice, it rarely competes with the above methods.

More related to our work is (Dadashi et al., 2021; Rezaeifar et al., 2021) where a clever pessimistic bonus is introduced in the rewards acting as anti-exploration. It represents the distance of the selected action with the one who would be selected by the behavioural policy. The authors showed it acted similarly than a KL-penalty between the learned and behavioural policy, linking their work with Wu et al. (2019) and Kumar et al. (2019a). However, playing with the rewards might lead to a completely modified goal (Ng et al., 1999) and finding an appropriate reward modification might be complicated (Harutyunyan et al., 2015). We argue that modifying the Q-values lead to a better policy control as the whole objective is considered. In this line of search, Kostrikov et al. (2021) augmented the Q-values with the entropy of the behavioural policy and constrained them with as Fisher divergence term. In contrast, rather than modifying the RL objective with an explicit penalty on the dataset distance, we use the density as an uncertainty surrogate in the value function estimation.

Note our dataset density estimation can be combined with most presented methods. Especially, it could replace the ”anti-exploration” bonus or be introduced to improve the Behavioural Cloning step in (Wu et al., 2019) required to build the measures of closeness between policies.

6 Experiments

In this section, we evaluate D-CQL on different environments from OpenAI Gym MuJoCo with D4RL datasets (Fu et al., 2020). For each environment, D4RL provides different datasets that are respectively gathered with `random`, `medium`, `expert`, `mixed` or `medium-expert` behavioural policies. `mixed` corresponds to a mixture between a random and a medium policy, and `medium-expert` a mixture between a medium and an expert policy. All the hyper-parameters used for the experiments can be found in Appendix C. For each experiment, we return the smooth average undiscounted result over 4 seeds.

A first important remark is that, when the quality of the dataset is too good, D-CQL is not designed to beat the state-of-the-art algorithm CQL[3]. Since the datasets already provide all the information required to build expert policies, the agent does not have to move away from the behavioural policy to extract a better policy and doing so would present the risk of overestimating Q-values. This can be seen on figure 2.On both `hopper-medium` and `hopper-medium-expert`, CQL and D-CQL have similar performances.

Another point of interest is the effect on distinguishing ID from OOD actions in expert environments. The ablation study on ν on `hopper-medium-expert` is shown in Figure 1 and on `hopper-expert` in Appendix C. It shows that this distinction is not relevant in when expert data are given and that it can thus prevent learning. We attribute this effect to the unbounded minimization of the Q-values towards minus infinity discussed in Section 3.2. Additional details about this phenomenon can be found in Appendix C.1.

Nevertheless, the weighting scheme ζ_ν has a great impact on `hopper-medium` as outlined in 1, where the trade-off between staying close to the behavioural policy and discovering better actions is culminating. In this setting, D-CQL outperforms or is comparable with CQL. However, we observe a decrease for both methods for `hopper-medium` that remains an open question in off-policy algorithms (Aviral Kumar, 2021; Kumar et al., 2021). Yet, the refined penalization benefits is clearly seen on `hopper-medium`, where a near-expert policy was extracted.

Finally, we highlight the improved representations of the Q-values. We can find on figure 2 the difference between the mean Q-values associated to 10 random actions and the ones associated to 10 behavioural actions during learning. D-CQL consistently has better represented Q-values than CQL. Besides, we observe that a slight difference of 0.1 on ν doubles the differences on the Q-values on most environments once again demonstrating the impact of this weighting factor.

[3]The results for CQL are taken from `https://github.com/aviralkumar2907/CQL`

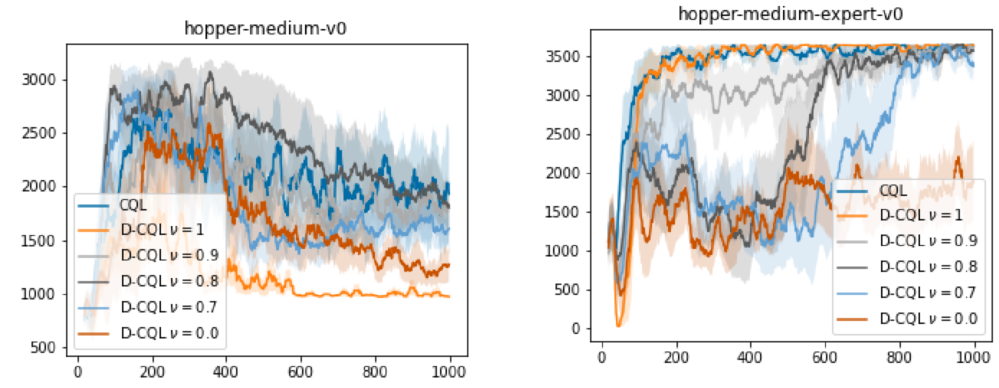

Figure 1: Performances of the algorithm w.r.t. to the hyper-parameter ν of the weighting scheme.

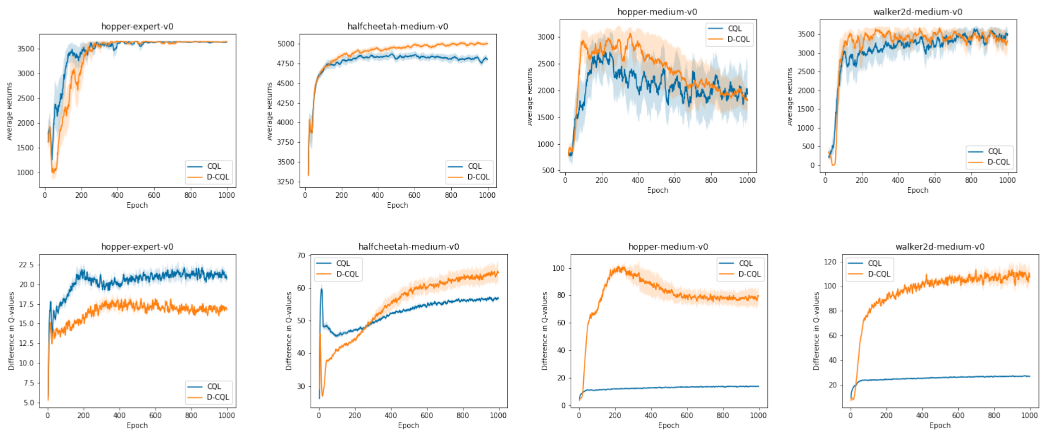

Figure 2: Comparison between D-CQL and CQL
The top four figures show the average returns of the algorithms. The bottom four show the average Q-value difference between ID and OOD actions (ID - OOD).

7 Conclusion and Future Work

We successfully proposed a Batch-RL algorithm with a strong lower-bound property on the value functions. The density ρ_β indeed provides insights to guide the agent towards meaningful areas. This induces better results in practice. In addition, the new lower bound property allows a clear distinction between ID state-action pairs and OOD ones that is of great interest in orthogonal applications such as *Safe Reinforcement Learning* or *Pessimistic Off-Policy Evaluation* where the agent must check the consequences of its decisions *before* playing them. An interesting future work would to apply our method on such problems.

This work focused on model-free agents that do not consider OOD states. However, ρ_β also yields information regarding to states that would be interesting to study in model-based algorithms. Our future works will focus in this direction.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31. PMLR, 2017.

Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 104–114. PMLR, 2020.

Oron Anschel, Nir Baram, and Nahum Shimkin. Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 176–185. PMLR, 2017.

Aaron Courville Tengyu Ma George Tucker Sergey Levine Aviral Kumar, Rishabh Agarwal. Value-based deep reinforcement learning requires explicit regularization. 2021.

Nicolò Cesa-Bianchi, Claudio Gentile, Gergely Neu, and Gábor Lugosi. Boltzmann exploration done right. In *NIPS*, pp. 6284–6293, 2017.

Felipe Codevilla, Eder Santana, Antonio M. López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *ICCV*, pp. 9328–9337. IEEE, 2019.

Robert Dadashi, Shideh Rezaeifar, Nino Vieillard, Léonard Hussenot, Olivier Pietquin, and Matthieu Geist. Offline reinforcement learning with pseudometric learning. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2307–2318. PMLR, 2021.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *ICLR (Poster)*. OpenReview.net, 2017.

Gabriel Dulac-Arnold, Nir Levine, Daniel J. Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Mach. Learn.*, 110(9):2419–2468, 2021.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2020.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1582–1591. PMLR, 2018.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 2019.

Evrard Garcelon, Mohammad Ghavamzadeh, Alessandro Lazaric, and Matteo Pirotta. Conservative exploration in reinforcement learning. In *AISTATS*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1431–1441. PMLR, 2020.

Çaglar Gülçehre, Ziyu Wang, Alexander Novikov, Thomas Paine, Sergio Gómez Colmenarejo, Konrad Zolna, Rishabh Agarwal, Josh Merel, Daniel J. Mankowitz, Cosmin Paduraru, Gabriel Dulac-Arnold, Jerry Li, Mohammad Norouzi, Matthew Hoffman, Nicolas Heess, and Nando de Freitas. RL unplugged: A collection of benchmarks for offline reinforcement learning. In *NeurIPS*, 2020.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Anna Harutyunyan, Sam Devlin, Peter Vrancx, and Ann Nowé. Expressing arbitrary reward functions as potential-based advice. In *AAAI*, pp. 2652–2658. AAAI Press, 2015.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *NIPS*, pp. 4565–4573, 2016.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Comput. Surv.*, 50(2):21:1–21:35, 2017.

Ivan Kobyzev, Simon Prince, and Marcus Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Ilya Kostrikov, Rob Fergus, Jonathan Tompson, and Ofir Nachum. Offline reinforcement learning with fisher divergence critic regularization. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5774–5783. PMLR, 2021.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *NeurIPS*, pp. 11761–11771, 2019a.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32:11784–11794, 2019b.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *NeurIPS*, 2020.

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In *ICLR*. OpenReview.net, 2021.

Sascha Lange, Thomas Gabel, and Martin A. Riedmiller. Batch reinforcement learning. In *Reinforcement Learning*, volume 12 of *Adaptation, Learning, and Optimization*, pp. 45–73. Springer, 2012.

Romain Laroche, Paul Trichelair, and Remi Tachet des Combes. Safe policy improvement with baseline bootstrapping. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3652–3661. PMLR, 2019.

Jongmin Lee, Wonseok Jeon, Byung-Jun Lee, Joelle Pineau, and Kee-Eung Kim. Optidice: Offline policy optimization via stationary distribution correction estimation. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6120–6130. PMLR, 2021a.

Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. SUNRISE: A simple unified framework for ensemble learning in deep reinforcement learning. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6131–6141. PMLR, 2021b.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *ICLR (Poster)*, 2016.

Yuping Luo, Huazhe Xu, and Tengyu Ma. Learning self-correctable policies and value functions from demonstrations with negative sampling. In *International Conference on Learning Representations*, 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. Algaedice: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*, 2019.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, pp. 278–287. Morgan Kaufmann, 1999.

Ling Pan, Qingpeng Cai, Qi Meng, Wei Chen, and Longbo Huang. Reinforcement learning with dynamic boltzmann softmax updates. In *IJCAI*, pp. 1992–1998. ijcai.org, 2020.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Dean Pomerleau. ALVINN: an autonomous land vehicle in a neural network. In *NIPS*, pp. 305–313. Morgan Kaufmann, 1988.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994.

Shideh Rezaeifar, Robert Dadashi, Nino Vieillard, Léonard Hussenot, Olivier Bachem, Olivier Pietquin, and Matthieu Geist. Offline reinforcement learning as anti-exploration. *arXiv preprint arXiv:2106.06431*, 2021.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1530–1538. JMLR.org, 2015.

Pornthep Rojanavasu, Phaitoon Srinil, and Ouen Pinngern. New recommendation system using reinforcement learning. *Special Issue of the Intl. J. Computer, the Internet and Management*, 13 (SP 3), 2005.

Claude Sammut, Scott Hurst, Dana Kedzier, and Donald Michie. Learning to fly. In *ML*, pp. 385–393. Morgan Kaufmann, 1992.

Manuel S. Santos and John Rust. Convergence properties of policy iteration. *SIAM J. Control. Optim.*, 42(6):2094–2115, 2004.

Noah Y Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, Nicolas Heess, and Martin Riedmiller. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *ICLR*, 2020a.

Noah Y. Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, Nicolas Heess, and Martin A. Riedmiller. Keep doing what worked: Behavior modelling priors for offline reinforcement learning. In *ICLR*. OpenReview.net, 2020b.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy P. Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nat.*, 550(7676):354–359, 2017.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998.

B. Uria, I. Murray, and H. Larochelle. Rnade: The real-valued neural autoregressive density-estimator. In *NIPS*, 2013.

Hado van Hasselt. Double q-learning. In *NIPS*, pp. 2613–2621. Curran Associates, Inc., 2010.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *AAAI*, pp. 2094–2100. AAAI Press, 2016.

Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. In *Self-Supervision for Reinforcement Learning Workshop-ICLR 2021*, 2021.

Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. DRN: A deep reinforcement learning framework for news recommendation. In *WWW*, pp. 167–176. ACM, 2018.

A Lower bound details

In this section, we re-state our main theorem and prove it. An additional insight is provided to handle the sampling errors.

Theorem 2 *For any μ satisfying $\operatorname{supp}\mu \subset \operatorname{supp}\pi_\beta$ and assuming there are no sampling errors, then the Q-values obtained by iterating equation 3 are*

$$\forall s \in \mathcal{D}, a \in \operatorname{supp}\pi_\beta(\cdot|s),\ Q^{\textit{D-CQL}}(s,a) = Q^\pi(s,a) - \alpha \left[(I - \gamma P^\pi)^{-1} \frac{\mu\, \zeta}{\pi_\beta} \right] (a|s)\ . \tag{10}$$

$\alpha > 0$ leads to a point-wise lower bound on the true Q-values that becomes tight whenever $\zeta(a|s) \approx 0$ with $a \sim \pi_\beta(\cdot|s)$.

Proof 1 *This proof is done assuming the true Bellman operator $\mathcal{B}$ has been used instead of $\hat{\mathcal{B}}$. Setting the integrande of the derivative of 3 leads to*

$$Q^{k+1}(s,a) = \mathcal{B}^\pi Q^k(s,a) - \alpha \frac{\mu(a|s)\, \zeta(a|s)}{\pi_\beta(a|s)}\ . \tag{11}$$

The interest of ζ is clearly exposed in this equation: the learned Q-values will be highly conservative when associated to OOD actions. On the other hand, when $\zeta(a|s) \approx 0$, the update leads to an application of the classic Bellman operator and drive ID Q-values to stay close to their real associated Q-value.

Let μ, π_β and ζ the matrices associated to $\{\mu(a|s)\}$, $\{\pi_\beta(a|s)\}$ and $\{\zeta(a|s)\}$ for all $s \in \mathcal{D}$ and $a \in \operatorname{supp}\pi_\beta(\cdot|s)$. The operator associated to this update: $\mathcal{T}_\zeta : Q \to \mathcal{B}^\pi Q^k - \alpha \frac{\mu\,\zeta}{\pi_\beta}$ defined for any function Q on defined on states belonging to $\mathcal{D}$ and actions belonging to $\operatorname{supp}\pi_\beta(\cdot|s)$ is a γ-contraction. Its fixed point is

$$Q^{\textit{D-CQL}} = Q^\pi - \alpha \left[(I - \gamma P^\pi)^{-1} \frac{\mu\, \zeta}{\pi_\beta} \right]\ . \tag{12}$$

Remark 1 *The sampling error can be handled using concentration inequalities and would lead to the following lower bound, that holds with probability $1 - \delta$:*

$$Q^{\textit{D-CQL}} \leq Q^\pi - \alpha \left[(I - \gamma P^\pi)^{-1} \frac{\mu\, \zeta}{\pi_\beta} \right] + (I - \gamma P^\pi)^{-1} \frac{C_{r,P,\delta}}{(1-\gamma)\sqrt{|\mathcal{D})|}}\ . \tag{13}$$

We refer to Appendix C of Kumar et al. (2020) for additional details.

B Density Estimation procedure

B.1 Normalizing Flows

We now describe the used architecture to estimate ρ_β. For simplicity, let $y = (s,a)$ and x its associated sample belonging to the X distribution and let D their dimension. Let's further assume there is only one differential mapping: $x = f(y)$.

Let $d < D$ and following RealNVP from Dinh et al. (2017), f maps y to x as follows:

$$\begin{aligned} x_{1:d} &= y_{1:d} \\ x_{d+1:D} &= y_{d+1:D} \odot \exp\left(s(y_{1:d})\right) + t(y_{1:d}) \end{aligned} \tag{14}$$

where s and t are functions defined in $\mathbb{R}^d \to \mathbb{R}^{m-d}$ and $\odot$ is the Hadamard product. This transformation results in a simple formula for the Jacobian determinant: $|\det(\operatorname{Jac} f)(y)| = \exp\left(\sum_j s(x_{1:d})_j\right)$.

This mapping is then repeated N times where each input is randomly permuted before applying equation 14 to ensure each dimension is rightfully processed.

In all datasets, s and t are parametrized by a Neural Network with 1 hidden layer composed by 256 neurons. $N = 10$ mappings f were used, d was set to be the integer mean of $\dim((s,a))$ and X was chosen as a Multivariate Normal distribution $\mathcal{N}(0, \mathcal{I})$.

B.2 Experimental validation

We first validate our Normalizing Flows density models and the capacity of ζ to be accurate on all tested environments. The objective is to ensure the learned density $\hat{\rho}_\beta$ is able to differentiate informative neighbourhoods from highly uncertain ones. In order to do so, we create three different action families: *close* (C), *medium* (M) and *far* (F). Far actions have been sampled uniformly from the action space. To ensure actions close to $\rho_\beta(\cdot|s)$ have not been sampled, we removed actions that belong to the ball centered in the dataset action $a_\mathcal{D}$ with radius 1 using the euclidean distance. Close and medium actions have been created with actions from the dataset perturbed with a Gaussian noise $a_{\{C,M\}} = a_\mathcal{D} + \lambda_{\{C,M\}}\mathcal{N}(0, \mathbb{I})$ with $\lambda_C = \frac{0.4}{\dim(\mathcal{A})}$ and $\lambda_M = \frac{0.8}{\dim(\mathcal{A})}$.

We provide the box-plot of the ratio $1 - \zeta = \frac{\hat{\rho}_\beta(a,s)}{\hat{\rho}_\beta(a_\mathcal{D},s)}$ for 100 actions sampled from each family for 2000 states. We should see a clear distinction between these families, and the ratio related to *close* actions should be near 1.

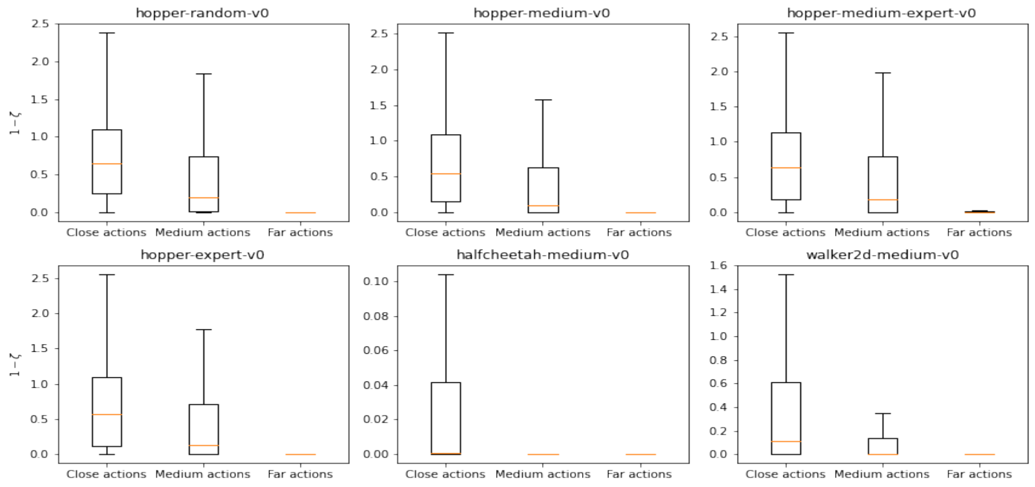

Figure 3: NF-validation

We can see the NF density estimation clearly distinguishes the different neighbourhoods according to their uncertainty. Even better, the fact that the ratio $\frac{\hat{\rho}_\beta(a,s)}{\hat{\rho}_\beta(a_\mathcal{D},s)}$ is often higher than 1 demonstrates $\hat{\rho}_\beta$ has not over-fitted: it is very likely than many actions close to $a_\mathcal{D}$ are closer to ρ_β than the actions from the dataset.

Second, we compare the different density estimation techniques proposed for Behavioural Cloning. We focused on the Multivariate Gaussian (MG) estimate proposed in (Wu et al., 2019) and the Mixture of Gaussian (MoG) introduced in (Kostrikov et al., 2021). Especially, we focused on the `medium-expert` datasets as they have been gathered with two different distributions.

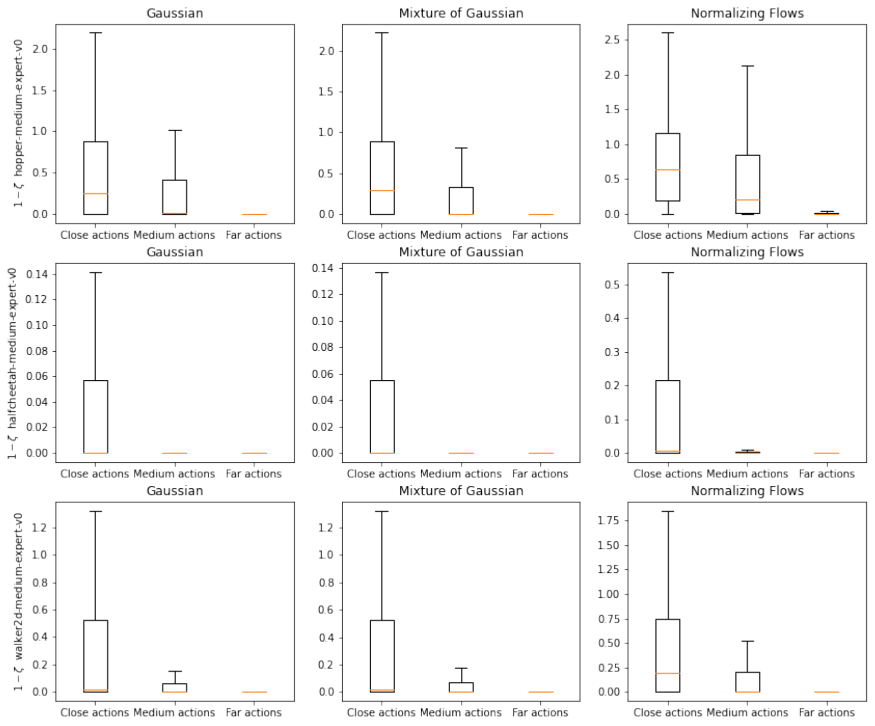

Figure 4: Density comparison techniques

We can that observe the advantages of focusing on Normalizing Flows: they are slightly less prone to over-fitting and represent well the different targeted neighborhoods.

C D-CQL ABLATION STUDY

C.1 HYPER-PARAMETERS

Here, we briefly present the hyper-parameter used in the experiments. CQL's code was extracted from `https://github.com/aviralkumar2907/CQL`. Most of the hyper-parameters - learning rates, size of the networks, optimizers, …- were unchainged to have a proper comparison with CQL. The only parameter we changed for figure 2 were α and ν, recapitulated in the following table:

C.2 EXPERIMENTS

Here, we provide an ablation study on the tested environments.

We confirm expert's datasets do not need to a refined penalization. In fact, choosing a ν different than 1 in `hopper-expert` leads to poor results as shown in 5. Another highly interesting point is the role of ν in the Q-values: any increase on ν leads to a highly increased distinction between ID

Table 1: Hyperparameters

	cheetah-m	hopper-r	hopper-m	hopper-e	walker-m
CQL α	5	1	5	10	5
D-CQL α	5	1	5	5	5
D-CQL ν	0.7	1	0.8	1	0.7

and OOD actions. It also helps us understand why the performance of the agent degrades wheb ν is too low: the difference becomes too high leading to errors in the representation.

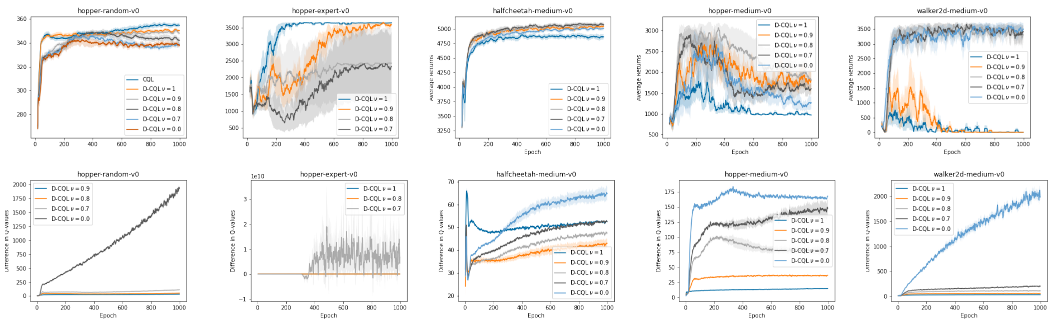

Figure 5: Ablation study on all of our environments