ON VALUE OPTIMIZATION IN CONSERVATIVE Q LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Conservatism, the act of underestimating an agent's expected value estimates, has demonstrated profound success in model-free, model-based, multi-task, safe and other realms of offline Reinforcement Learning (RL). Recent work, on the other hand, has noted that conservatism often hinders learning of behaviors. To that end, the paper asks the question *how does conservatism affect offline learning?* The proposed answer studies Conservative Q Learning in light of value function optimization, approximate objectives that upper bound underestimations and behavior cloning as auxilary regularization objective. Conservative agents implicitly steer estimates away from the true value function, resulting in optimization objectives with high condition numbers. Mitigating these issues requires an upper bounding objective. These approximate upper bounds, however, impose a strong assumption on a scaling matrix, a result which is only sparsely fulfilled. Driven by theoretical observations, provision of an auxilary behavior cloning objective as variational regularization to estimates results in accurate value estimation, well-conditioned search spaces and expressive parameterizations. In an empirical study of discrete and continuous control tasks, we validate our theoretical insights and demonstrate the practical effects of learning underestimated value functions.

1 INTRODUCTION

Consider the scenario wherein an agent learns to drive a car. The agent observes a human driving the car. This involves paying attention to crucial insights of controlling the vehicle such as steering while making a turn, accelerating during a green light and monitoring mirrors during brakes. Increasing amount of observations made available to the agent result in learning finer details of the task which are not available *apriori*. For instance, the agent may never learn to make a U-turn if the human never encountered a U-turn crossing.

Offline Reinforcement Learning (RL) (Kalashnikov et al., 2018; Levine et al., 2020) addresses this intuitive gap in learning by equipping the driver (the *agent* in above example) with the ability to stitch together portions of observations by making use of a dataset of transitions. For instance, the driver may learn to make a U-turn on its own if it observes the teacher (*human*) making sharp turns and slowing down the vehicle at intersections. Adoption of transitions in the offline setting allows the agent to tackle distributional shift (Kumar et al., 2019) between current and behavior policies.

Recent interests in offline RL (Kumar et al., 2020b; Ajay et al., 2021) textcolorrespare motivated by conservatively learning value estimates. Conservative Q values, in expectation, yield a lower bound on the true Q values which facilitates in constraining an agent's policy to be close to the dataset's behavior policy. Additionally, a conservative lower bound generalizes towards a family of objectives which regularize the agent away from the path of Out-Of Distribution (OOD) actions. This combination of data-driven learning with behavioral constraints demonstrates efficacious learning with extensions to safe policy improvement. However, a concrete theory of lower bounded Q values remains unexplored.

Conservative agents often face increased gaps between optimal values and lower-bounded estimates (Kumar et al., 2021b). This leads the policy to overfit the behavior policy, hence hindering provision of diverse behaviors. The nature of problem arises twofold. On one hand, conservatism bakes in a policy distribution which is challenging to optimize in practical settings (Ajay et al., 2021). On the other hand, conservatism imposes prior conditions on dataset design, a scenario effectively addressed by alternate paradigms (Sinha et al., 2021).

Before asking the question how can conservatism be improved?, one must begin by asking *how does conservatism affect offline learning?* The proposed answers realize conservatism in light of value function optimization, upper bounding approximations and behavior cloning as regularization within the Conservative Q Learning (CQL) (Kumar et al., 2020b) framework. Our novel contributions are threefold; (1) We show that optimizing offline policies conservatively steers estimates away from true Q functions. This gives rise to ill-conditioned objectives driven by high condition numbers. The challenge is further pronounced with underparameterized function approximators. (2) Upper bounding approximations, on the other hand, mitigate the above issues but hurt convergence. These upper bounds impose a challenging assumption on a scaling matrix during policy optimization. (3) We further show that provision of a behavior cloning objective as variational regularization empirically results in accurate value estimates, well-conditioned search spaces and expressive parameterizations. We validate our theoretical insights and their practical effects in discrete and continuous control simulations.

2 Related Work

Offline Reinforcement Learning: A myriad of offline RL methods (Levine et al., 2020) provision learning of policies from a predefined set of behaviors (Wang et al., 2021; Agarwal et al., 2020). Data-driven learning has demonstrated strides resulting from safe exploration (Bharadhwaj et al., 2021; Eysenbach et al., 2018) to connection of prior skills with new experience (Singh et al., 2020) in robotics. Additionally, provision of prior data collection facilitates scalability (Kalashnikov et al., 2018) to high-dimensional manipulation scenarios. Recent theoretical advances aim at tackling the covariance shift (Gelada & Bellemare, 2019) arising from erroneous bootstrapping of values. Towards this direction, prior efforts highlight the construction of robust off-policy optimization techniques (Huang & Jiang, 2020). While off-policy methods (Liu et al., 2020a) highlight their effectiveness for behavior primitives (Nair et al., 2020; Lange et al., 2012), regularization techniques (Liu et al., 2020b) depict promise in the model-based (Kidambi et al., 2020) setting.

Pessimistic Learning: Various works develop penalty-based optimization methods for offline RL (Ghasemipour et al., 2021; Fujimoto et al., 2019; Kumar et al., 2021b; Yu et al., 2021a). Although centered towards the regularization of policy iteration scheme (massoud Farahmand et al., 2016), penalties utilized to address distributional shift are instances of pessimistic updates (Jin et al., 2021; Buckman et al., 2021). Distributional shift may be addressed using policy constraints in the policy space (Lee et al., 2021) or action sequences in the action space (Zhou et al., 2020). Between these two extremes is value function regularization (Kumar et al., 2020b; Yu et al., 2021b) which learns in the policy space and regularizes in the value space. Penalization of Q values is closely related to uncertainty estimation which does away with erroneous bootstrapping of estimates (Osband et al., 2016; O'Donoghue et al., 2018). Our work explains conservatism based on the above techniques.

Optimization and Approximation Theory: Prior works in offline RL borrow from statistical optimization (Wang et al., 2021). Analysis of value function regularization is often extended to primal-dual spaces (Bharadhwaj et al., 2021). This provisions surrogate objective designs which provide the construction of safe policy guarantees bounding sub-optimality and policy convergence. Alternatively, tractable probabilistic approximations (Wu et al., 2021) are employed to explicitly account for epistemic uncertainty in OOD detection of state-action pairs. Additionally, convergence evaluation (Yin et al., 2021b) and optimality (Xiao et al., 2021) analysis of off-policy optimization relies on uncertainty in metric spaces. Our theoretical analysis is parallel to these prior efforts.

3 Problem Formulation

The problem utilizes the offline RL setup wherein an agent adheres to an environment in order to transition to new states and observe rewards by following a sequence of actions. The problem is modeled as a finite-horizon Markov Decision Process (MDP) (Sutton & Barto, 2018) defined by the tuple $(\mathcal{S}, \mathcal{A}, r, P, \gamma)$ where the state space is denoted by $\mathcal{S}$ and action space by $\mathcal{A}$, r presents the reward observed by agent such that $r : \mathcal{S} \times \mathcal{A} \to [0, r_{\max}]$, $P : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, \infty)$ presents the unknown transition model consisting of the transition probability to the next state s_{t+1} (or simply $s') \in \mathcal{S}$ given the current state s_t (or simply $s) \in \mathcal{S}$ and action a_t (or simply $a) \in \mathcal{A}$ at time step t and γ is the discount factor. The agent's policy $\pi_\theta(a_t|s_t)$ is a function of its parameters θ and a behavior policy $\hat{\pi}_\beta(a_t|s_t)$ with the discounted marginal state distribution $d^{\hat{\pi}_\beta}(s_t)$. A dataset $\mathcal{D} \sim d^{\hat{\pi}_\beta}(s_t)\hat{\pi}_\beta(a_t|s_t)$ describes (s, a) pairs. Offline RL defines the agent's objective to maximize the expected discounted reward $\mathbb{E}_{s_t, a_t \sim \mathcal{D}}[\sum_{t=1}^{T} \gamma^{t-1} r(s_t, a_t)]$.

CQL (Kumar et al., 2020b) is an offline RL algorithm which addresses distributional shift by obtaining an expected lower bound on the Q-values. CQL regularizes the Bellman objective utilizing expected Q values $\mathbb{E}_{s\sim\mathcal{D},a\sim\hat{\pi}_\beta}[Q(s,a)]$ under the behavior policy $\hat{\pi}_\beta$. To explicitly highlight the per-sample dependence of Q values on states and actions, we use $Q(s,a)$ and otherwise denote the value function as simply Q (clear from context). The empirical Q-values of agent's policy, $\hat{Q}^k$, are underestimated at each policy iteration k utilizing the empirical Bellman operator $\hat{\mathcal{B}}^\pi Q = r(s,a) + \gamma\mathbb{E}_{s'\sim P(s'|s,a)}[Q(s',a')]$ and a tradeoff constant α. The CQL objective, for a regularization $\mathcal{R}(\mu) = -\mathrm{D}_{\mathrm{KL}}(\mu,\rho)$ as the KL-divergence with a prior $\rho(a|s)$ and $\mu(s,a)$ as the state-action distribution, reduces to CQL($\mathcal{H}$) expressed in Eq. 1.

$$\mathrm{CQL}(\mathcal{H}) = \min_Q \alpha\,\mathbb{E}_{s\sim\mathcal{D}} \underbrace{[\mathrm{lse}(Q)}_{\text{soft-maximum}} \underbrace{- \mathbb{E}_{a\sim\hat{\pi}_\beta}[Q(s,a)]]}_{\text{expected value}} + \underbrace{\frac{1}{2}\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(Q(s,a) - \hat{\mathcal{B}}^\pi\hat{Q}^k\right)^2\right]}_{\text{empirical Bellman error}} \tag{1}$$

Eq. 1 uses $\mathrm{lse}(Q)$ to denote $\log\sum_a \exp Q(s,a)$ with Q as shorthand for $Q(s,a)$. The second term indicates a pessimistic expected value corresponding to in-disrtibution actions arising from $\hat{\pi}_\beta$ and the third term constitutes the empirical Bellman error. We overload notation by denoting $\mathbb{H}$ as the Hessian of CQL($\mathcal{H}$) and other (explicitly stated) quantities in general.

4 Conservatism in Offline Learning

The proposed answer studies two facets of CQL($\mathcal{H}$) objective (see Appendix D and Appendix G for conjugate and dual space analyses respectively). (1) Conservatism results in ill-conditioned objectives. (2) Tractable approximations to improve the search process impose a strong assumption during policy optmization. While the analysis is centered to the model-free setting, our results readily transfer to model-based setting (see Appendix E). To further simplify intuition, suitable theoretical results are denoted in blue, potential hindrances in red and key insights in boxes.

4.1 Conservative Value Function Optimization

Estimation errors arising from value estimates may hurt training. Prior works study this phenomenon in light of value and policy-based methods (Van Hasselt et al., 2016; Fujimoto et al., 2018). However, the effect is often omitted during evaluation of agent's optimization process. For instance, a natural question to ask is *how do estimation errors affect the space of value functions?* The answer to the above lies in the assessment of value function space curvature. Consider the Hessian $\mathbb{H}$ of CQL($\mathcal{H}$) in Eq. 2. Here, $\mathrm{softmax}(Q) = \frac{\exp(Q(s_t,a_t))}{\sum_{a\in\mathcal{A}}\exp(Q(s_t,a))}$ denotes the $|\mathcal{A}|$ dimensional softmax over Q and $\hat{\pi}_\beta$ the $|\mathcal{A}|$ dimensional action probabilities from dataset. The first term, being a symmetric matrix obtained from the outer product of $\mathrm{softmax}(Q)$, highlights the dependence of curvature on value function. The second term denotes action likelihoods from dataset.

$$\mathbb{H} = \nabla^2_Q\mathrm{CQL}(\mathcal{H}) = \underbrace{\alpha\,\mathrm{softmax}(Q)^{\mathrm{T}}(\mathbf{1} - \mathrm{softmax}(Q))}_{\substack{\text{curvature depends on}\\ \text{estimations}}} + \underbrace{\hat{\pi}_\beta}_{\substack{\text{curvature depends}\\ \text{on dataset}}} \tag{2}$$

To assess the curvature of $\mathbb{H}$, one considers the condition number $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$ which is the ratio of largest to smallest eigenvalues of $\mathbb{H}$. κ signifies the smoothness of $\mathbb{H}$ curvature by informing how close to singular $\mathbb{H}$ may be or how efficiently search methods will perform in the value space. High κ values denote ill-conditioned curvatures with the lowest value of $\kappa = 1$ being desirable (Nocedal & Wright, 2006). Upon evaluating the convergence of $\mathbb{H}$ as a convex quadratic (see Appendix A), we obtain a lower bound of Eq. 3 on κ with estimation errors exponentiated around true Q function.

Explanation 1 (Ill-Conditioned Value Space) *The conservative framework, for all Q values, traverses an ill-conditioned value space governed by high condition numbers κ,*

$$\left|\frac{\|\mathrm{softmax}(\hat{Q}^k) - \mathrm{softmax}(Q)\|_2 + \|\mathrm{softmax}(\hat{Q}^{k+1}) - \mathrm{softmax}(Q)\|_2}{\|\mathrm{softmax}(\hat{Q}^k) - \mathrm{softmax}(Q)\|_2 - \|\mathrm{softmax}(\hat{Q}^{k+1}) - \mathrm{softmax}(Q)\|_2}\right| \leq \kappa \tag{3}$$

Eq. 3 describes geometry of value function space. As per the design of conservative policy evaluation (Kumar et al., 2020b), Q values $\hat{Q}^{k+1}$ at each subsequent iteration $k+1$ are underestimated. This results in the following case of Eq. 3 $\|\mathrm{softmax}(\hat{Q}^k) - \mathrm{softmax}(Q)\|_2 <$

$\|\text{softmax}(\hat{Q}^{k+1}) - \text{softmax}(Q)\|_2$. The expression indicates increasing values of the lower bound which shift κ towards higher values (see Appendix A). Intuitively, an increase in the estimation error lead to ill-conditioned value function space which is found challenging to traverse.

4.2 Value Functions & Underparameterization

We further study the optimization process using parameterization. Recent work notes the phenomenon of *underparameterization*, wherein value-based methods face a rapid decrease in the rank of parameters (Kumar et al., 2021a). This decrease is studied in the function approximation setting. A degradation in parameter rank corresponds to reduced expressivity of the value network. This is because the effective number of bits required to express the Q function reduce with each iteration. Other works connect this drop in expressivity to successive iterations of collapsing self-distillation (Mobahi et al., 2020). While underparameterization is a well-known phenomenon in the deep RL setting, its link to value optimization process remains unexplored.

We bridge this gap by connecting underparameterization with optimization. Denote $\underline{t}$ as the number of rounds for which the rank of value function does not drop, prior work (Mobahi et al., 2020) notes the relation of Eq. 4 (where $\|\hat{\pi}_\beta\| > \sqrt{|\mathcal{D}|\epsilon}$ for constant ϵ).

$$\underline{t} = \frac{K-1}{\kappa}, \text{ where } K = \frac{\|\hat{\pi}_\beta\|}{\sqrt{|\mathcal{D}|\epsilon}} \text{ is a constant} \tag{4}$$

Since $|\mathcal{D}|$ is fixed and $\|\hat{\pi}_\beta\|$ does not change during training, Eq. 4 emphasizes on the inverse relationship $\underline{t} \propto \frac{1}{\kappa}$. Intuitively, an increased value of $\underline{t}$, i.e.- the number of steps for which the rank does not drop, lead to smoothened curvatures of the parameter space. On the other hand, lower values, i.e.- more frequent underparameterization, result in ill-conditioned search spaces. This leads us to Explanation 2.

Explanation 2 (Ill-Conditioned Search Space) *Underparamterization in value-based function approximators inhibits expressivity and induces an ill-conditioned search space characterized by increasing condition numbers κ.*

Rank degradation arises as a direct consequence of value prediction in Temporal Difference (TD) learning (Kumar et al., 2021a). The degradation reduces values of $\underline{t}$ which leads to inaccurate value estimates. This reduction in $\underline{t}$ relates to ill-conditioned search spaces with increasing κ values over the course of training. Our empirical evaluation verifies this trend in Section 5.

4.3 Value Functions & Böhning Approximation

We further address high condition numbers by seeking higher-order alternatives using tractable approximations. This would provide the policy with increased curvature information. As a suitable example, we study the well-known Böhning's quadratic approximation (Böhning, 1992; Murphy, 2012) to $\text{lse}(Q)$. Approximating lse allows use to address high variance and inaccurate estimates during training (Khan, 2012). We seek an approximation which is an upper bound on CQL($\mathcal{H}$) and simultaneously a tight lower bound on the true Q values. Towards this goal, consider an upper bound on the hessian of CQL($\mathcal{H}$) using the second order Taylor's series expansion of $\text{lse}(Q)$ around $\psi \in \mathbb{R}^{|\mathcal{A}|}$,

$$\text{lse}(Q) = \text{lse}(\psi) + (Q-\psi)^{\mathrm{T}} g(\psi) + \frac{1}{2}(Q-\psi)^{\mathrm{T}} \mathbb{H}(\psi)(Q-\psi) \tag{5}$$

where $g(\psi) = \exp(\psi - \text{lse}(\psi))$ denotes the gradient of $\text{lse}(\psi)$ and $\mathbb{H}(\psi) = \text{diag}(g(\psi)) - g(\psi)g(\psi)^{\mathrm{T}}$ its hessian. Forming an upper bound on the hessian $\mathbb{H}(\psi) \leq \hat{\mathbb{H}}$ such that $x^{\mathrm{T}}\mathbb{H}x \leq x^{\mathrm{T}}\hat{\mathbb{H}}x$ for all x,

$$\hat{\mathbb{H}} = \frac{1}{2}\left[I_{|\mathcal{A}|} - \frac{1}{|\mathcal{A}|+1} 1_{|\mathcal{A}|} 1_{|\mathcal{A}|}^{\mathrm{T}}\right] \tag{6}$$

Here $\hat{\mathbb{H}}$ is the *Böhning Approximation*. Utilizing $\hat{\mathbb{H}}$ in place of the true Hessian $\mathbb{H}(\psi)$ and simplifying Eq. 5 leads to the following formulation,

$$\text{lse}(Q) \leq \frac{1}{2} Q^{\mathrm{T}} \hat{\mathbb{H}} Q - bQ + c \tag{7}$$

The above quadratic expression is denoted with $\text{Bohn}(Q)$ where $b = -(\hat{\mathbb{H}}\psi - g(\psi))$ and $c = \frac{1}{2}\psi^{\mathrm{T}}\hat{\mathbb{H}}\psi - g(\psi)^{\mathrm{T}}\psi + \text{lse}(\psi)$. Upon assessing the convergence of $\text{Bohn}(Q)$ (see Appendix B) one obtains Eq. 8. On comparing this to Explanation 1, we observe that estimation errors are not on the exponential scale anymore. This leads to an alternate lower bound on κ which empirically presents improvement in search space.

$$\left| \frac{\|\hat{Q}^k - Q\|_2 + \|\hat{Q}^{k+1} - Q\|_2}{\|\hat{Q}^k - Q\|_2 - \|\hat{Q}^{k+1} - Q\|_2} \right| \leq \kappa \tag{8}$$

Since $\text{Bohn}(Q)$ is an upper bound on $\text{lse}(Q)$, $\text{lse}(Q) \leq \text{Bohn}(Q)$, plugging it into CQL($\mathcal{H}$) in Eq. 1 objective yields a suitable result $\text{CQL}_{\text{Bohn}}(\mathcal{H})$.

$$\text{CQL}_{\text{Bohn}}(\mathcal{H}) = \min_Q \alpha \underbrace{\mathbb{E}_{s\sim\mathcal{D}}\left[\text{Bohn}(Q)\right]}_{\text{approximation}} - \underbrace{\mathbb{E}_{a\sim\hat{\pi}_\beta}[Q]}_{\text{expected value}} + \underbrace{\frac{1}{2}\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(Q - \hat{\mathcal{B}}^\pi\hat{Q}^k\right)^2\right]}_{\text{empirical Bellman error}} \tag{9}$$

Similar to the original CQL($\mathcal{H}$) objective, $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ consists of an expected value (second term in Eq. 9) and the empirical Bellman error (third term in Eq. 9). In contrast to the high-variance soft-maximum $\text{lse}(Q)$, the objective enjoys a tractable $\text{Bohn}(Q)$ approximation. Fig. 1 (left) presents the relationship between the two objectives.

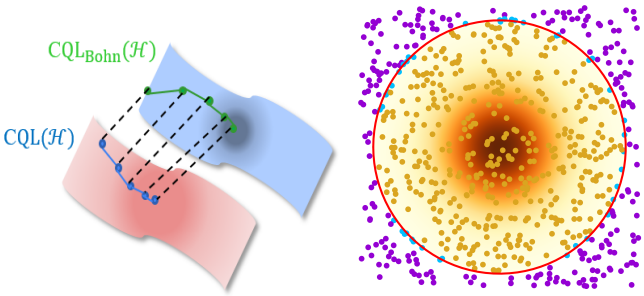

Figure 1: **(left)** Relationship between CQL($\mathcal{H}$) and $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ objectives, **(right)** Diagonal entries of $\tilde{\mathbb{H}}$ as basis points with unit circle in red. Values in violet lie in overestimation regime. Values in golden lie in overconservatism regime. Values (seldomly encountered) in blue are accurate.

With that said, it is only left to show that $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ establishes a tight lower bound on true Q values. However, $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ does not satisfy this result. The Böhning approximation trades off a smoother objective for (1) a lower bound on true Q values and (2) the key property of $\text{lse}(Q)$ to be a contraction (Asadi & Littman, 2017). While we address the latter in Appendix B showing that $\text{Bohn}(Q)$ obeys contraction, the former imposes a strong and challenging assumption which we discuss next.

We follow the process of CQL (Kumar et al., 2020b) and analyze Eq. 9 by optimizing over Q values and setting the derivative to 0,

$$Q = \underbrace{\tilde{\mathbb{H}}}_{\text{scaling matrix}} \underbrace{\hat{\mathcal{B}}^\pi\hat{Q}^k}_{\text{estimate}} - \underbrace{\alpha(\alpha\hat{\mathbb{H}} + \hat{\pi}_\beta(a|s))^{-1}}_{\text{effective tradeoff}} \underbrace{(\hat{\mathbb{H}}\psi - \text{softmax}(\psi) - \hat{\pi}_\beta(a|s))}_{\text{underestimation}} \tag{10}$$

Eq. 10 provides a general result for Q values at each iteration. The expression consists of a scaling rate which is determined by the entries of the matrix $\tilde{\mathbb{H}} = (\frac{\hat{\pi}_\beta(a|s)}{\alpha\hat{\mathbb{H}} + \hat{\pi}_\beta(a|s)})^{-1}$. Note the dependence of iterates on behavior policy distribution $\hat{\pi}_\beta(a|s)$ which suggests that a dataset with good coverage may lead to improved convergence.

Assumption on $\tilde{\mathbb{H}}$: We now consider the role of $\tilde{\mathbb{H}}$ in optimization process. In order for the policy evaluation scheme to provide an accurate estimate, the first two terms when combined should yield the Bellman estimate $\hat{\mathcal{B}}^\pi\hat{Q}^k$. Mathematically, $\tilde{\mathbb{H}}\hat{\mathcal{B}}^\pi\hat{Q}^k = \hat{\mathcal{B}}^\pi\hat{Q}^k$. Note that this result can only be achieved if the matrix $\tilde{\mathbb{H}}$ is exactly equal to the identity $\tilde{\mathbb{H}} = \mathbf{I}$. The expression indicates that all diagonal terms of $\tilde{\mathbb{H}}$ must equal to 1. This leads us to Explanation 3.

Explanation 3 (Strong Assumption on $\tilde{\mathbb{H}}$) *Smoother conservative objectives require the scaling matrix $\tilde{\mathbb{H}}$ to be identity, $\tilde{\mathbb{H}} = \mathbf{I}$.*

Fig. 1 (right) presents an intuition of the above result. If the diagonal entries of the matrix lie inside the unit circle of the basis space ($\tilde{\mathbb{H}}_{ii} < 1$), then the Q values result in collapsing estimates.

At each subsequent policy iteration, estimates $\hat{\mathcal{B}}^{\pi}\hat{Q}^{k}$ decay to minimum values and collapse at the center of unit circle. In case the values lie outside the unit circle ($\tilde{\mathbb{H}}_{ii} > 1$), Q values will result in overestimations and hence, diverge upon subsequent policy iterations. Thus, in order for $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ to underestimate Q values, all diagonal entries of the matrix $\tilde{\mathbb{H}}$ should lie on the unit circle ($\tilde{\mathbb{H}}_{ii} = 1$) with off-diagonal entries as 0.

4.4 Conservatism through the lens of Behavior Cloning

Provision of ill-conditioned search spaces and a strong assumption on $\tilde{\mathbb{H}}$ hurt convergence to the true Q function. On one hand, ill-conditioned curvature hinders the optimization process of value function. While on the other hand, $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ requires $\tilde{\mathbb{H}} = \mathbf{I}$.

Our analysis of the optimization process implicitly hints at a potential solution. To improve the optimization process, one must seek low κ values which would address underparameterization. Additionally, these search spaces must facilitate approximations which address distributional shift.

We address the former by following prior works in regularization. Iterative rank degredation may be evaded utilizing regularization as an auxilary learning signal in the optimization process (Mobahi et al., 2020). Explicit penalization of Q values fulfills two objectives; (1) the scheme presents an auxilary objective which motivates the agent to maximize expected returns while matching behaviors to the dataset. (2) The objective provides a richer learning signal in cases where rewards may not be informative (Matusch et al., 2020). Based on this insight, we incorporate value function regularization as a suitable tool for retaining expressive parameterization.

While a variety of regularizers emerge as suitable candidates, we seek the one which best approximates the dataset. An ideal objective should effectively capture prior transitions with minimum unrealizable requirements. This would make policy's optimization convenient. Towards this goal, we explore the information theoretic alternative of Mutual Information $\mathcal{MI}\{\pi; \hat{\pi}_{\beta}\}$ (Poole et al., 2019) between agent and behavior policies π and $\hat{\pi}_{\beta}$ respectively. While $\mathcal{MI}\{\pi; \hat{\pi}_{\beta}\}$ presents increased expressivity of $\hat{\pi}_{\beta}$ prior, the quantity is itself difficult to estimate. We alleviate this by deriving a variational lower bound objective (Bishop, 2006) (see Appendix C) consisting of the tractable posterior approximation $q(\pi|\hat{\pi}_{\beta})$ and entropy regularization $\mathcal{H}(\pi)$. Such a scheme resembles behavior cloning with noisy dataset demonstrations (Rajeswaran et al., 2018). We theoretically show that the scheme is ϵ-stable and converges to true Q faster than $\text{CQL}(\mathcal{H})$ (see Appendix C).

$$\mathcal{MI}\{\pi; \hat{\pi}_{\beta}\} \geq \underbrace{\mathbb{E}_{p(\pi,\hat{\pi}_{\beta})}[\log q(\pi|\hat{\pi}_{\beta})]}_{\text{approximation}} + \underbrace{\mathcal{H}(\pi)}_{\text{entropy regularization}} \tag{11}$$

Explanation 4 (Convergence with Smooth Objectives) *Conservative offline learning objectives, for all Q values, yield improved empirical convergence with behavior cloning regularization serving as tighter approximations to Q values.*

It is worth noting that naively incorporating the lower bound with returns as Intrinsic Motivation (IM) may lead to instabilities (Xie et al., 2021). In addition, expected returns being estimated by the agent are based on off-policy dataset transitions and must be utilized in a similar fashion. Thus, we propose to use an off-policy variational regularization of Eq. 12 wherein Q values are regularized following sampled trajectories. We direct the curious reader to Appendix C for the full derivation.

$$\hat{Q}^{k} := \hat{Q}^{k} + \mathbb{E}_{p(\pi,\hat{\pi}_{\beta})}\left[\log\left(\prod_{t=1}^{T} q_t(\pi|\hat{\pi}_{\beta})^{\gamma^{t-1}}\right)\right] \tag{12}$$

Eq. 12 comprises of the per-timestep posterior $q_t(\pi|\hat{\pi}_{\beta})$ ranging up to the horizon length T. Note the additional reduction in bootstrapping errors due to the exponentiated γ^{t-1} in likelihood. The posterior is implicitly downweighed over longer temporal spans which prevent compounding of errors in estimates. The formulation, when combined with $\text{CQL}(\mathcal{H})$, leads to the CQL-IM objective. We validate the soundness of variational optimization and approximations in the next section.

5 Experiments

Motivated by theoretical evidence, our evaluation aims to study the practical effects of underestimations when agents demonstrate significant conservatism. More specifically, our experiments aim at addressing the following questions in discrete and control action settings;

(1) How does the conservative framework of offline learning compare to the online setting?

(2) How does underestimation in Q values affect the search space?

(3) Does underparameterization of the value function hamper learning?

(4) How much conservatism does CQL($\mathcal{H}$) demonstrate?

(5) How does the condition on $\tilde{\mathbb{H}}$ imposed by Bohn(Q) explain CQL($\mathcal{H}$)?

(6) How much efficacy does the variational approximation yield in practice?

(7) Are the behaviors learned offline significant from a representational perspective?

5.1 The LineWorld Domain

We briefly visit the toy example of a linefollower agent in the Lineworld domain from Kumar et al. (2019) presented in Fig. 2 (top). The agent starts at the location S and is tasked to reach the goal location G by consistently moving right as its optimal policy $\pi^*(\cdot|s)$. For each step in the right direction, the agent observes a reward of $+1$. For each step of left action the agent observes a -1 reward resulting in the termination of the episode. The total number of states correspond to the number of steps between the start and goal states. The setup consists of two settings, namely *online* and *offline* agents. The *online* agent utilizes Q-learning and enjoys its own data collected from the domain. The *offline* agent utilizes CQL($\mathcal{H}$) and is initialized with a dataset of sub-optimal transitions wherein the behavioral policy $\hat{\pi}_\beta$ *always opposes* the optimal policy $\pi^*(\cdot|s)$ by going left. The *offline* agent is allowed to collect data at selective intervals after the first 100 steps of simulation (see Appendix I for details).

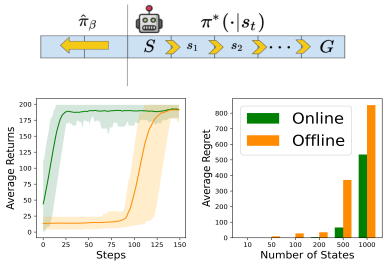

Figure 2: **(top)** The Lineworld domain, **(bottom-left)** Average Returns, **(bottom-right)** Average Regret (results averaged over 100 runs)

The evaluation yields insights towards posed questions; **(1)** Fig. 2 (bottom-left) presents the comparison of average returns achieved by *online* and *offline* agents for 200 states domain. The online agent, by virtue of data collection, learns from rich transitions observed in the environment. This leads to faster and consistent learning. On the other hand, the offline agent learns slowly due to the sub-optimal preferences of behavioral policy arising from the dataset. Once the agent starts collecting data, it updates its estimates based on observed transitions and retrieves optimal behavior.

One can further study the scalability of problem by varying number of states in the environment. Fig. 2 (bottom-right) presents the variation of average regret (calculated with respect to optimal returns after 100 steps) with the number of states. The offline agent observes a faster accumulation in regret in comparison to the online agent. Additionally, the difference between regrets of offline and online agents widens with the former aggregating more regret over larger state spaces. Thus, the offline learning problem additionally restricts the scalability of agents towards larger state spaces.

5.2 Aerial Control

The setting evaluates agents on a suite of aerial control tasks simulating Drones (Panerati et al., 2021; Coumans, 2015) wherein our agent is a CF2X quadcopter controlling the torques applied to the 4 motor fins at 240 Hz (see Appendix F for D4RL experiments). At each step, agent's policy observes the drone's position, velocity and orientation as a state. Policies are evaluated on 4 diverse tasks; (1) *takeoff* requires the policy to gradually ascend upward for flight, (2) *hover* requires the policy to stay mid-air above a given location, (3) *zigzag* requires the policy to manoeuvre in a zig-zag flight pattern and (4) *flythrugate*, being a significantly challenging scenario, requires the policy to escape through a narrow gate opening. Fig. 3 provides an illustration of the tasks with details deferred to Appendix I.

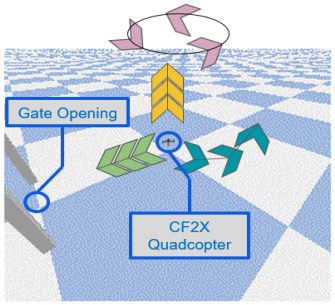

Figure 3: Aerial tasks *takeoff*, *hover*, *zigzag* and *flythrugate* illustrated in their respective colors moving away from the drone

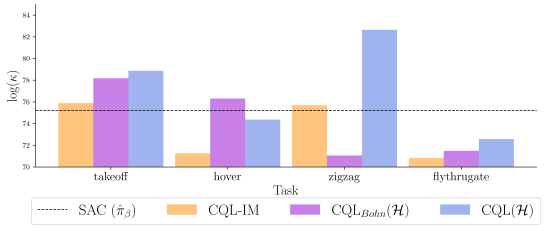

Figure 4: Comparison of average $\log \kappa$

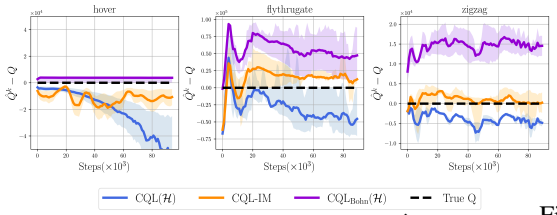

Figure 5: Comparison of the gap $\hat{Q}^k - Q$

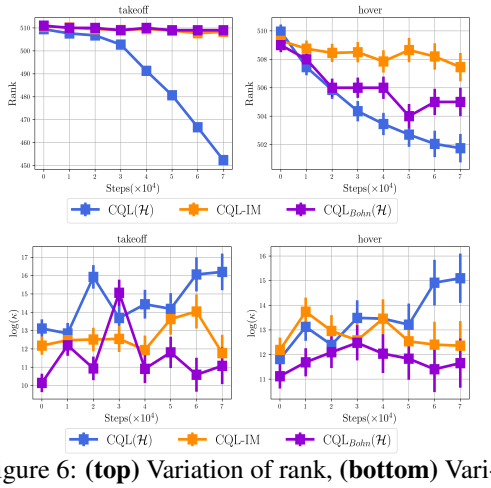

Figure 6: **(top)** Variation of rank, **(bottom)** Variation of $\log(\kappa)$ during training.

We address the theoretical claims of our evaluation by comparing CQL($\mathcal{H}$) to Behavior Cloning (BC) and Soft Actor-Critic (SAC) (Haarnoja et al., 2018). Additionally, CQL$_{\text{Bohn}}(\mathcal{H})$ denotes CQL with Bohn(Q). All offline policies were trained subject to a pretrained SAC as the behavior policy.

Fig. 4 compares average of condition numbers during training. The plot answers the second question, **(2)** Underestimated Q values pose a challenging search space which is confirmed by high κ values in log scale. Compared to CQL($\mathcal{H}$), CQL$_{\text{Bohn}}(\mathcal{H})$ has a moderately suitable objective. CQL-IM, on the other hand, proclaims a well-behaved objective reflective of low $\log(\kappa)$ values.

(3) We now verify our claim on underparameterization by studying its effect on the search space. Following the setting of Kumar et al. (2021a), Fig. 6 (top) presents the evolution in rank of weight matrix corresponding to last layer of the critic (see Appendix F for additional results). One readily notices a steady drop for CQL($\mathcal{H}$) arising from conventional TD updates. CQL-IM, by virtue of variational regularization, is found robust to the rank degradation phenomenon. Fig. 6 (bottom) further connects expressivity to search space by observing the evolution of $\log(\kappa)$. Proggressive degradation of rank drives increasing values of κ for CQL($\mathcal{H}$). CQL-IM remains robust to this trend, hence presenting a well-conditioned search space.

Fig. 5 compares conservatism by virtue of the gap $\hat{Q}^k - Q$ (Kumar et al., 2020b) wherein $\hat{Q}^k$ are the values predicted by offline methods at iteration k and Q is the true Q value as per average discounted returns. The comparison is representative of desired approximations by suggesting that a low $\hat{Q}^k - Q$ gap is indicative of the value estimates closely resembling the true Q values. Observed results answer the remaining questions; **(4)** CQL($\mathcal{H}$) learns significantly conservative Q values in expectation which steer the agent away from optimality. Estimates learned by virtue of excessive underestimations drift away from the true Q values, hence highlighting their theoretical concerns. **(5)** CQL$_{\text{Bohn}}(\mathcal{H})$ represents an empirical upper bound on CQL($\mathcal{H}$). However, requirements raised by the Bohn(Q) approximation for scaling matrix $\tilde{\mathbb{H}}$ hinder convergence of CQL$_{\text{Bohn}}(\mathcal{H})$ towards true Q values. **(6)** Lastly, values learned by CQL-IM closely match the true values. This is a direct consequence of learning smoother information-theoretic objectives which sufficiently represent the true expected returns by virtue of an expressive posterior.

Method	*takeoff*	*hover*	*zigzag*	*flythrugate*
BC	0.86±0.17	**0.97±0.02**	**0.96±0.17**	**0.98±0.13**
SAC ($\hat{\pi}_\beta$)	0.88±0.21	0.97±0.11	0.94±0.33	0.96±0.08
CQL($\mathcal{H}$)	**0.89±0.07**	0.95±0.03	0.89±0.06	0.86±0.15
CQL$_{\text{Bohn}}(\mathcal{H})$	0.51±0.23	0.92±0.05	0.67±0.19	0.77±0.27
CQL-IM	0.86±0.06	0.95±0.01	0.94±0.09	0.97±0.08

Table 1: Normalized average returns on Drone Control tasks over 10 runs. Values in **bold** denote highest returns. Values in green indicate best returns between offline RL policies.

Method	Low κ	Low rank collapse	Less Conservative	Less assumptions
CQL($\mathcal{H}$)	✗	✗	✗	✓
CQL$_{\text{Bohn}}(\mathcal{H})$	✓	✓	✓	✗
CQL-IM	✓	✓	✓	✓

Table 2: Summary of comparisons.

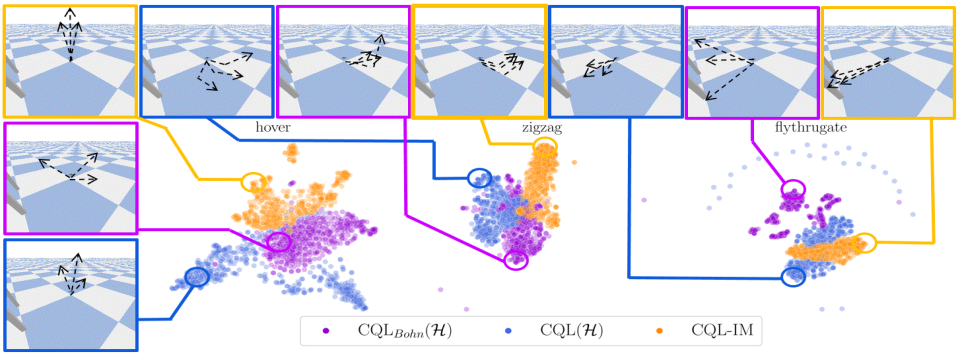

Figure 7: 2D t-SNE embeddings of representations learned by the critic parameters.

To further study the impact of erroneous approximations, one can assess the task performance of methods in Table 1. CQL-IM, by virtue of intrinsic motivation, effectively captures the behavior policy distribution and is competitive to BC and online SAC. $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ presents consistent returns but falls short of optimal performance.

Random runs in RL may hinder fair comparison between algorithms. For instance, a good run of a method when combined with its other moderate runs can push its otherwise on-par performance higher. Following the recommendation of Lones (2021), we validate the statistical significance of results by carrying out the Mann-Whitney U test (Mann & Whitney, 1947). All 40 seeds (10 seeds per task) of each algorithm are compared to those of $\hat{\pi}_\beta$ (online SAC) to yield the $\mathcal{U}$ statistic. $\mathcal{U}$ here denotes the statistical significance of performance with higher values being desirable. We obtain $\mathcal{U}_{\text{BC}} = 123, \mathcal{U}_{\text{CQL}} = 88, \mathcal{U}_{\text{Bohn}} = 13$ and $\mathcal{U}_{\text{IM}} = 97$ with subscripts denoting respective algorithms. Results in Table 1 are validated as BC and CQL-IM have higher statistical significance over $\text{CQL}(\mathcal{H})$ and $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ when compared to $\hat{\pi}_\beta$.

(7) Qualitatively, representations learned by the critic parameters link estimation errors with actor's behaviors. Fig. 7 presents 2D t-SNE (van der Maaten & Hinton, 2008) embeddings of these representations. The critic parameters appropriately capture conservative, optimistic and near-accurate estimations which are well-seperated in the low-dimensional representation space. We further visualize behaviors of actors corresponding to critic's value estimates. In *flythrugate*, $\text{CQL}(\mathcal{H})$ agent underestimates its Q values while moving towards the gate. This results in the drone's trajectories falling short of optimal behavior. $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ overestimates Q values which results in the drone's trajectories overshooting optimal behavior. Contrary to above, CQL-IM presents accurate Q values resulting in near-optimal behaviors.

6 Discussion

Summary: The paper studied conservatism through the lens of value function optimization, upper bounding approximations and behavior cloning as regularization in the CQL framework (Table 2). Optimizing underestimated value functions steers the agent away from true value distributions. The problem is further reflected as a link from underparameterization to ill-conditioned curvatures. A tractable upper bound approximation to $\text{lse}(Q)$ mitigates these challenges but imposes a strong condition on scaling matrix $\tilde{\mathbb{H}}$ to be $\mathbf{I}$. Driven by its theoretical concerns, the framework incorporates behavior cloning as off-policy variational regularization. The presented scheme results in accurate value estimation, well-conditioned search spaces and expressive parameterizations.

Limitations: Data-driven mechanisms and RL reside at the pinnacle of machine learning. A theoretical study of the two opens several new avenues for future work. We briefly highlight two promising directions for further perusal; **(1)** A practical realization of the information-theoretic alternative and its scalability to higher dimensions remain unbolted. An optimistic perspective towards this problem is obtained using variational theory. Future directions could utilize alternate objectives with information rich training signals for offline policies. **(2)** Orthogonally, one can seek alternative offline mechanisms which would utilize tractable approximations to guarantee convergence. A meticulous inspection of the policy structure sheds promise towards this aspect. We hope that an explanation of the conservative learning framework serves as a motivating direction towards practical offline learning problems.

REFERENCES

Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, 2020.

Anurag Ajay, Aviral Kumar, Pulkit Agrawal, Sergey Levine, and Ofir Nachum. Opal: Offline primitive discovery for accelerating offline reinforcement learning. In *International Conference on Learning Representations*, 2021.

Anonymous. Should i run offline reinforcement learning or behavioral cloning? In *Submitted to The Tenth International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=AP1MKT37rJ. under review.

Kavosh Asadi and Michael L Littman. An alternative softmax operator for reinforcement learning. In *International Conference on Machine Learning*, 2017.

Dimitri P Bertsekas. *Abstract dynamic programming*. Athena Scientific Nashua, NH, USA, 2018.

Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-dynamic Programming*, volume 1. Athena Scientific, 1995.

Homanga Bharadhwaj, Aviral Kumar, Nicholas Rhinehart, Sergey Levine, Florian Shkurti, and Animesh Garg. Conservative safety critics for exploration. In *International Conference on Learning Representations*, 2021.

Nam Parshad Bhatia and Giorgio P Szegö. *Stability theory of dynamical systems*. Springer Science & Business Media, 2002.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Jacob Buckman, Carles Gelada, and Marc G Bellemare. The importance of pessimism in fixed-dataset policy optimization. In *International Conference on Learning Representations*, 2021.

Dankmar Böhning. Multinomial logistic regression algorithm. *Annals of the Institute of Statistical Mathematics*, 44(1):197–200, March 1992.

Qi Cai, Mingyi Hong, Yongxin Chen, and Zhaoran Wang. On the global convergence of imitation learning: A case for linear quadratic regulator. *arXiv preprint arXiv:1901.03674*, 2019.

Erwin Coumans. Bullet physics simulation. In *ACM SIGGRAPH*, 2015.

Benjamin Eysenbach, Shixiang Gu, Julian Ibarz, and Sergey Levine. Leave no trace: Learning to reset for safe and autonomous reinforcement learning. In *International Conference on Learning Representations*, 2018.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *arXiv preprint arXiv:2106.06860*, 2021.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, 2018.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, 2019.

Bolin Gao and Lacra Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

Carles Gelada and Marc G Bellemare. Off-policy deep reinforcement learning by bootstrapping the covariate shift. In *AAAI Conference on Artificial Intelligence*, volume 33, 2019.

Seyed Kamyar Seyed Ghasemipour, Dale Schuurmans, and Shixiang Shane Gu. Emaq: Expected-max q-learning operator for simple yet effective offline and online rl. *International Conference on Machine Learning*, 2021.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, volume 80, pp. 1861–1870, 2018.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Jiawei Huang and Nan Jiang. From importance sampling to doubly robust policy gradient. In *International Conference on Machine Learning*, 2020.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, 2021.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, 2018.

Mohammad Khan. *Variational learning for latent Gaussian model of discrete data*. PhD thesis, University of British Columbia, 2012.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. In *Neural Information Processing Systems*, 2020.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Neural Information Processing Systems*, volume 32, 2019.

Aviral Kumar, Abhishek Gupta, and Sergey Levine. Discor: Corrective feedback in reinforcement learning via distribution correction. In *Neural Information Processing Systems*, volume 33, 2020a.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Neural Information Processing Systems*, 2020b.

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In *International Conference on Learning Representations*, 2021a.

Aviral Kumar, Anikait Singh, Stephen Tian, Chelsea Finn, and Sergey Levine. A workflow for offline model-free robotic reinforcement learning. In *5th Annual Conference on Robot Learning*, 2021b.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning*, pp. 45–73. 2012.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Addressing distribution shift in online reinforcement learning with offline datasets, 2021.

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.

Xiao Li, Yao Ma, and Calin Belta. Automata guided reinforcement learning with demonstrations. *arXiv preprint arXiv:1809.06305*, 2018.

Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Off-policy policy gradient with stationary distribution correction. In *Uncertainty in Artificial Intelligence*, 2020a.

Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Provably good batch off-policy reinforcement learning without great exploration. In *Neural Information Processing Systems*, 2020b.

Michael A Lones. How to avoid machine learning pitfalls: a guide for academic researchers. *arXiv preprint arXiv:2108.02497*, 2021.

Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18, 1947.

Amir massoud Farahmand, Mohammad Ghavamzadeh, Csaba Szepesvári, and Shie Mannor. Regularized policy iteration with nonparametric function spaces. *Journal of Machine Learning Research*, 17(139):1–66, 2016.

Brendon Matusch, Jimmy Ba, and Danijar Hafner. Evaluating agents without rewards. *arXiv preprint arXiv:2012.11538*, 2020.

Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. Self-distillation amplifies regularization in hilbert space. In *Neural Information Processing Systems*, volume 33, 2020.

Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

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.

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6292–6299. IEEE, 2018.

Ashvin Nair, Murtaza Dalal, Abhishek Gupta, and Sergey Levine. Accelerating online reinforcement learning with offline datasets, 2020.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2006.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Neural Information Processing Systems*, volume 29, 2016.

Brendan O'Donoghue, Ian Osband, Remi Munos, and Volodymyr Mnih. The uncertainty bellman equation and exploration. In *International Conference on Machine Learning*, pp. 3836–3845, 2018.

Jacopo Panerati, Hehui Zheng, SiQi Zhou, James Xu, Amanda Prorok, and Angela P Schoellig. Learning to fly–a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control. *arXiv preprint arXiv:2103.02142*, 2021.

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.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, 2019.

Nived Rajaraman, Lin F Yang, Jiantao Jiao, and Kannan Ramachandran. Toward the fundamental limits of imitation learning. *arXiv preprint arXiv:2009.05990*, 2020.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. doi: 10.15607/RSS.2018.XIV.049.

Avi Singh, Albert Yu, Jonathan Yang, Jesse Zhang, Aviral Kumar, and Sergey Levine. Chaining behaviors from data with model-free reinforcement learning. In *Conference on Robot Learning*, 2020.

Samarth Sinha, Ajay Mandlekar, and Animesh Garg. S4RL: Surprisingly simple self-supervision for offline reinforcement learning in robotics. In *5th Annual Conference on Robot Learning*, 2021.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9, 2008.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Ruosong Wang, Dean Foster, and Sham M. Kakade. What are the statistical limits of offline RL with linear function approximation? In *International Conference on Learning Representations*, 2021.

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

Yue Wu, Shuangfei Zhai, Nitish Srivastava, Joshua Susskind, Jian Zhang, Ruslan Salakhutdinov, and Hanlin Goh. Uncertainty weighted actor-critic for offline reinforcement learning. *arXiv preprint arXiv:2105.08140*, 2021.

Chenjun Xiao, Yifan Wu, Tor Lattimore, Bo Dai, Jincheng Mei, Lihong Li, Csaba Szepesvari, and Dale Schuurmans. On the optimality of batch policy optimization algorithms. *arXiv preprint arXiv:2104.02293*, 2021.

Kevin Xie, Homanga Bharadhwaj, Danijar Hafner, Animesh Garg, and Florian Shkurti. Latent skill planning for exploration and transfer. In *International Conference on Learning Representations*, 2021.

He Yin, Peter Seiler, Ming Jin, and Murat Arcak. Imitation learning with stability and safety guarantees. *IEEE Control Systems Letters*, 2021a.

Ming Yin, Yu Bai, and Yu-Xiang Wang. Near-optimal provable uniform convergence in offline policy evaluation for reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1567–1575, 2021b.

Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Sergey Levine, and Chelsea Finn. Conservative data sharing for multi-task offline reinforcement learning. *arXiv preprint arXiv:2109.08128*, 2021a.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *arXiv preprint arXiv:2102.08363*, 2021b.

Wenxuan Zhou, Sujay Bajracharya, and David Held. Plas: Latent action space for offline reinforcement learning. In *Conference on Robot Learning*, 2020.

Konrad Zolna, Alexander Novikov, Ksenia Konyushkova, Caglar Gulcehre, Ziyu Wang, Yusuf Aytar, Misha Denil, Nando de Freitas, and Scott Reed. Offline learning from demonstrations and unlabeled experience. *arXiv preprint arXiv:2011.13885*, 2020.

A CONSERVATIVE VALUE FUNCTION OPTIMIZATION

A.1 HESSIAN OF CQL($\mathcal{H}$)

The computation of CQL($\mathcal{H}$) hessian follows the computaion of its gradient,

$$\nabla_Q \text{CQL}(\mathcal{H}) = \alpha \left(\text{softmax}(Q) - \hat{\pi}_\beta\right) + \hat{\pi}_\beta \left(Q - \hat{\mathcal{B}}^\pi \hat{Q}^k\right) \tag{13}$$

$$\nabla_Q^2 \text{CQL}(\mathcal{H}) = \underbrace{\alpha\, \text{softmax}(Q)^{\mathrm{T}} (\mathbf{1} - \text{softmax}(Q))}_{\substack{\text{curvature depends on} \\ \text{estimations}}} + \underbrace{\hat{\pi}_\beta}_{\substack{\text{curvature depends} \\ \text{on dataset}}} \tag{14}$$

It is readily noticeable that CQL($\mathcal{H}$) is convex in the space of real Q values. More formally, $\text{softmax}(Q) \in (0, 1]$ and $\mathbf{1}$ are $|\mathcal{A}|$ dimensional arrays which yields $\text{softmax}(Q)^{\mathrm{T}}(1 - \text{softmax}(Q))$ as a $|\mathcal{A}| \times |\mathcal{A}|$ positive definite matrix. This leaves $\hat{\pi}_\beta$ which is an $|\mathcal{A}|$ dimensional array of action probabilities and is always nonnegative. Thus, the complete sum results in a positive definite matrix with dimensions $|\mathcal{A}| \times |\mathcal{A}|$.

The first term emphasizes the dependence of estimations on a smooth curvature. Presence of $\hat{\pi}_\beta$, on the other hand, highlights the dependence of curvature on dataset with action likelihoods reflecting a fixed contribution.

A.2 OPTIMIZATION OF CQL($\mathcal{H}$) LANDSCAPE

This section studies the space of CQL($\mathcal{H}$) by analyzing its Hessian in the presence of steepest descent method. Restating the Hessian of CQL($\mathcal{H}$),

$$\nabla_Q^2 \text{CQL}(\mathcal{H}) = -\alpha\, \text{softmax}(Q)^{\mathrm{T}} \text{softmax}(Q) + \alpha\, \text{softmax}(Q) + \hat{\pi}_\beta \tag{15}$$

As before, $\text{softmax}(Q)$ here denotes an $|\mathcal{A}|$ dimensional array which makes the first term an $|\mathcal{A}| \times |\mathcal{A}|$ matrix. The second term remains an $|\mathcal{A}|$ dimensional array with α being a scalar. Finally, the third term $\hat{\pi}_\beta$ denotes an $|\mathcal{A}|$ dimensional array of action probabilities. The sum of second and third term when broadcasted to each column of the first term results in an $|\mathcal{A}| \times |\mathcal{A}|$ positive definite matrix.

Since $\hat{\pi}_\beta$ is held fixed during training, Eq. 15 may be realized as a quadratic expression with the variable $\hat{\mathrm{v}} = \text{softmax}(Q)$, A as a matrix with diagonal entries -2α, d as a vector with entries α and e as the fixed $\hat{\pi}_\beta$,

$$\text{Quad}(\hat{\mathrm{v}}) = \frac{1}{2} \hat{\mathrm{v}}^{\mathrm{T}} \mathrm{A} \hat{\mathrm{v}} - d\hat{\mathrm{v}} + e \tag{16}$$

Consider the gradient of $\text{Quad}(\hat{\mathrm{v}})$,

$$\nabla_{\hat{\mathrm{v}}} \text{Quad}(\hat{\mathrm{v}}) = \mathrm{A}\hat{\mathrm{v}} - d \tag{17}$$

The minimizer $\hat{\mathrm{v}} = \mathrm{v}$ is the unique solution to the linear system $\mathrm{A}\hat{\mathrm{v}} = d$. One can compute the optimal step length η_k that minimizes $\text{Quad}(\hat{\mathrm{v}} - \eta \nabla_{\hat{\mathrm{v}}} \text{Quad}(\hat{\mathrm{v}}))$.

$$\begin{aligned} \text{Quad}(\hat{\mathrm{v}} - \eta \nabla_{\hat{\mathrm{v}}} \text{Quad}(\hat{\mathrm{v}})) = \frac{1}{2} \left(\hat{\mathrm{v}} - \eta \nabla_{\hat{\mathrm{v}}} \text{Quad}(\hat{\mathrm{v}})\right)^{\mathrm{T}} \mathrm{A} \left(\hat{\mathrm{v}} - \eta \nabla_{\hat{\mathrm{v}}} \text{Quad}(\hat{\mathrm{v}})\right) \\ - d^{\mathrm{T}} \left(\hat{\mathrm{v}} - \eta \nabla_{\hat{\mathrm{v}}} \text{Quad}(\hat{\mathrm{v}})\right)^{\mathrm{T}} + e \end{aligned} \tag{18}$$

Setting the derivative w.r.t η to 0, we get,

$$\eta_k = \frac{\nabla_{\hat{\mathrm{v}}} \text{Quad}(\hat{\mathrm{v}})^{\mathrm{T}} \nabla_{\hat{\mathrm{v}}} \text{Quad}(\hat{\mathrm{v}})}{\nabla_{\hat{\mathrm{v}}} \text{Quad}(\hat{\mathrm{v}})^{\mathrm{T}} \mathrm{A} \nabla_{\hat{\mathrm{v}}} \text{Quad}(\hat{\mathrm{v}})} \tag{19}$$

Utilizing this in the steepest descent update rule yields the following,

$$\hat{\mathrm{v}}^{k+1} = \hat{\mathrm{v}}^k - \left(\frac{\nabla_{\hat{\mathrm{v}}} \text{Quad}(\hat{\mathrm{v}})^{\mathrm{T}} \nabla_{\hat{\mathrm{v}}} \text{Quad}(\hat{\mathrm{v}})}{\nabla_{\hat{\mathrm{v}}} \text{Quad}(\hat{\mathrm{v}})^{\mathrm{T}} \mathrm{A} \nabla_{\hat{\mathrm{v}}} \text{Quad}(\hat{\mathrm{v}})} \right) \nabla_{\hat{\mathrm{v}}} \text{Quad}(\hat{\mathrm{v}}) \tag{20}$$

We now study the rate of convergence. Using $\mathrm{A}\mathrm{v} = d$, we have,

$$\frac{1}{2} \|\hat{\mathrm{v}} - \mathrm{v}\|_2^2 = \text{Quad}(\hat{\mathrm{v}}) - \text{Quad}(\mathrm{v}) \tag{21}$$

Utilizing Eq. 20 and noting that $\nabla_{\hat{\mathrm{v}}} \operatorname{Quad}(\hat{\mathrm{v}}) = \mathrm{A}(\hat{\mathrm{v}} - \mathrm{v})$, we have the following,

$$\|\hat{\mathrm{v}}^{k+1} - \mathrm{v}\|_2^2 = \chi \|\hat{\mathrm{v}}^{k} - \mathrm{v}\|_2^2 \tag{22}$$

where $\chi = \left(1 - \frac{\left(\nabla_{\hat{\mathrm{v}}} \operatorname{Quad}(\hat{\mathrm{v}})^{\mathrm{T}} \nabla_{\hat{\mathrm{v}}} \operatorname{Quad}(\hat{\mathrm{v}})\right)^2}{(\nabla_{\hat{\mathrm{v}}} \operatorname{Quad}(\hat{\mathrm{v}})^{\mathrm{T}} \mathrm{A} \nabla_{\hat{\mathrm{v}}} \operatorname{Quad}(\hat{\mathrm{v}}))(\nabla_{\hat{\mathrm{v}}} \operatorname{Quad}(\hat{\mathrm{v}})^{\mathrm{T}} \mathrm{A}^{-1} \nabla_{\hat{\mathrm{v}}} \operatorname{Quad}(\hat{\mathrm{v}}))}\right)$.

The above results in the following bound,

$$\|\hat{\mathrm{v}}^{k+1} - \mathrm{v}\|_2^2 \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 \|\hat{\mathrm{v}}^{k} - \mathrm{v}\|_2^2 \tag{23}$$

$$\|\hat{\mathrm{v}}^{k+1} - \mathrm{v}\|_2^2 \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^2 \|\hat{\mathrm{v}}^{k} - \mathrm{v}\|_2^2 \tag{24}$$

where $0 \leq \lambda_1 \leq \lambda_2 \ldots \leq \lambda_n$ are the eigenvalues of A and $\kappa = \frac{\lambda_n}{\lambda_1}$.

Dependence of $\operatorname{Quad}(\hat{\mathrm{v}})$ convergence on κ highlights its necessity for a well-conditioned objective. If κ is large, $\operatorname{Quad}(\hat{\mathrm{v}})$ optimization will be ill-conditioned. On the other hand, if $\kappa \to 1$, $\left(\frac{\kappa-1}{\kappa+1}\right)^2 \to 0$, resulting in sound convergence.

We thus establish a lower bound on κ which captures estimation errors in $\hat{\mathrm{v}}$ values. Simply rearranging Eq. 24 yields the following,

$$\frac{\|\hat{\mathrm{v}}^{k+1} - \mathrm{v}\|_2^2}{\|\hat{\mathrm{v}}^{k} - \mathrm{v}\|_2^2} \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^2 \tag{25}$$

$$\frac{\|\hat{\mathrm{v}}^{k+1} - \mathrm{v}\|_2}{\|\hat{\mathrm{v}}^{k} - \mathrm{v}\|_2} + 1 \leq \kappa - \kappa \left(\frac{\|\hat{\mathrm{v}}^{k+1} - \mathrm{v}\|_2}{\|\hat{\mathrm{v}}^{k} - \mathrm{v}\|_2}\right) \tag{26}$$

$$\left|\frac{1 + \frac{\|\hat{\mathrm{v}}^{k+1} - \mathrm{v}\|_2}{\|\hat{\mathrm{v}}^{k} - \mathrm{v}\|_2}}{1 - \frac{\|\hat{\mathrm{v}}^{k+1} - \mathrm{v}\|_2}{\|\hat{\mathrm{v}}^{k} - \mathrm{v}\|_2}}\right| \leq \kappa \tag{27}$$

$$\left|\frac{\|\hat{\mathrm{v}}^{k} - \mathrm{v}\|_2 + \|\hat{\mathrm{v}}^{k+1} - \mathrm{v}\|_2}{\|\hat{\mathrm{v}}^{k} - \mathrm{v}\|_2 - \|\hat{\mathrm{v}}^{k+1} - \mathrm{v}\|_2}\right| \leq \kappa \tag{28}$$

Finally, plugging in $\hat{\mathrm{v}} = \operatorname{softmax}(\hat{Q})$ and $\mathrm{v} = \operatorname{softmax}(Q)$ provides the lower bound on κ.

$$\left|\frac{\|\operatorname{softmax}(\hat{Q}^{k}) - \operatorname{softmax}(Q)\|_2 + \|\operatorname{softmax}(\hat{Q}^{k+1}) - \operatorname{softmax}(Q)\|_2}{\|\operatorname{softmax}(\hat{Q}^{k}) - \operatorname{softmax}(Q)\|_2 - \|\operatorname{softmax}(\hat{Q}^{k+1}) - \operatorname{softmax}(Q)\|_2}\right| \leq \kappa \tag{29}$$

We further prove by example that increasing underestimations drive the condition number to higher values. Begin by considering the case wherein the error at iteration $k+1$ is twice of error at iteration k, $|\operatorname{softmax}(\hat{Q}^{k+1}) - \operatorname{softmax}(Q)| = 2|\operatorname{softmax}(\hat{Q}^{k}) - \operatorname{softmax}(Q)|$. Plugging this in Eq. 3 yields a lower bound of 3. Following one step of policy update the bound becomes,

$$\left|\frac{\|\operatorname{softmax}(\hat{Q}^{k+1}) - \operatorname{softmax}(Q)\|_2 + \|\operatorname{softmax}(\hat{Q}^{k+2}) - \operatorname{softmax}(Q)\|_2}{\|\operatorname{softmax}(\hat{Q}^{k+1}) - \operatorname{softmax}(Q)\|_2 - \|\operatorname{softmax}(\hat{Q}^{k+2}) - \operatorname{softmax}(Q)\|_2}\right| \leq \kappa \tag{30}$$

Considering the case of increasing estimation errors such that the error at iteration $k+2$ is thrice the error at iteration k, $|\operatorname{softmax}(\hat{Q}^{k+2}) - \operatorname{softmax}(Q)| = 3|\operatorname{softmax}(\hat{Q}^{k}) - \operatorname{softmax}(Q)|$. Utilizing this in Eq. 30 results in a lower bound of 5 which is greater than the previous bound of 3. The above indicates that κ increases with increasing values of estimation errors.

A.3 Rank Collapse & κ

We derive Eq. 4 adapted from Mobahi et al. (2020). Starting with the assumption $\|\pi^*\| > \sqrt{K\epsilon}$ which translates into $r_t > 1$ for the iterate r_t in the search space. Using the lower bound of Mobahi

et al. (2020) (Eq. 141), we arrive at the following results with $a(\kappa) = \frac{(r_0-1)^2+s(2r_0-1)}{(r_0-1+s)^2}$, $b(\kappa) = \frac{r_0^2 s}{(r_0-1+s)^2}$ and $r_0 = \frac{1}{\sqrt{K\epsilon}}\|\pi\|$.

$$r_t = a^t(\kappa) r_0 - b(\kappa)\frac{a^t(\kappa)-1}{a(\kappa)-1} \tag{31}$$

Setting the above expression to 1 yields the following,

$$t = \frac{\log\left(\frac{1-a(\kappa)+b(\kappa)}{b(\kappa)+r_0(1-a(\kappa))}\right)}{\log\left(a(\kappa)\right)} \tag{32}$$

Simplifying the log terms by plugging the expressions for $a(\kappa)$ and $b(\kappa)$ results in the following inequality,

$$\frac{\log\left(\frac{1+\frac{\kappa-1}{r_0^2}}{1+\frac{\kappa-1}{r_0}}\right)}{\log\left(1-\frac{(\frac{\kappa-1}{r_0}+\frac{1}{r_0})(\frac{\kappa-1}{r_0})}{(1+\frac{\kappa-1}{r_0})^2}\right)} \geq \frac{r_0-1}{\kappa} \tag{33}$$

Thus,

$$t \geq \frac{r_0-1}{\kappa} = \frac{\frac{\|\pi^*\|}{\sqrt{K\epsilon}}}{\kappa} \geq \frac{\frac{\hat{\pi}_\beta}{\sqrt{K\epsilon}}-1}{\kappa} \tag{34}$$

A.4 Additional Related Work

Value Function Regularization: Various recent methods in literature utilize value function penalties (Wu et al., 2019; Kumar et al., 2019; Jin et al., 2021; Buckman et al., 2021). Provision of penalties to the value function results in constraining agents towards data transitions (Kumar et al., 2021b). An additional benefit of using behavior penalties is the improvement in policy performance and overall runtime (Fujimoto & Gu, 2021). Regularization methods may constrain the policy using predefined sets of divergence metrics such as MMD kernels (Kumar et al., 2019) or KL regularization (Wu et al., 2019) in the case of actor-critic algorithms. An alternate way to regularize values is using explicit parameterizations of agent's policy (Fujimoto et al., 2019). Beyond policy constraints, alternate schemes such as Bellman uncertainty functions (Buckman et al., 2021) and uncertainty quantifiers are also found effective in practice (Jin et al., 2021). Lastly, the more recent class of constrained RL methods (Anonymous, 2022) use Bernstein inequalities to obtain generic regularization terms. These utilize estimates of dynamics and state action pair counts. Our regularization scheme is motivated by these prior efforts.

Imitation Learning combined with Offline RL: Offline RL methods adapt BC as regularizations (Fujimoto & Gu, 2021) and auxilary objectives (Rajeswaran et al., 2018; Rajaraman et al., 2020) to tackle distributional shift. Various works in this domain have witnessed improvements in learning of behaviors off-policy (Hester et al., 2018). A concrete use-case of this scenario is of learning from demonstrations (Zolna et al., 2020). BC allows learning of policies from complex multi-modal human data transitions where naive TD estimates fail to capture behaviors (Wu et al., 2021). Additionally, provision of demonstrations motivates exploration in real world applications (Nair et al., 2018) as well as guided search of policies (Li et al., 2018). We combine BC with Offline RL based on the above directions.

B Conservatism & Approximations

B.1 Contractive Nature of Bohn(Q)

This section theoretically shows that the contractive nature of softmax $\text{lse}(Q)$ is preserved upon replacing it with $\text{Bohn}(Q)$ in CQL($\mathcal{H}$). More specifically, we show that the quadratic approximation $\text{Bohn}(Q)$ is a contraction for specific values of ψ. Consider two Q-value functions, Q and $Q^{'}$ and define a norm $||\cdot||$. Consider the following expression,

$$\left\| \text{Bohn}(Q) - \text{Bohn}(Q^{'}) \right\| \tag{35}$$

$$= \left\| \frac{1}{2} Q^{\mathrm{T}} \hat{\mathbb{H}} Q - bQ + c - \frac{1}{2} Q^{'\,\mathrm{T}} \hat{\mathbb{H}} Q^{'} + bQ^{'} - c \right\| \tag{36}$$

$$= \left\| \frac{1}{2} (Q^{\mathrm{T}} \hat{\mathbb{H}} Q - Q^{'\,\mathrm{T}} \hat{\mathbb{H}} Q^{'}) + b(Q^{'} - Q) \right\| \tag{37}$$

$$\leq \frac{1}{2} \left\| Q^{\mathrm{T}} \hat{\mathbb{H}} Q - Q^{'\,\mathrm{T}} \hat{\mathbb{H}} Q^{'} \right\| + b \left\| Q^{'} - Q \right\| \tag{38}$$

Wherein the first inequality results from the triangle inequality. Considering the first term in Eq. 38,

$$\leq \frac{1}{4} \left\| Q^{\mathrm{T}} (I_{|\mathcal{A}|} - \frac{1}{|\mathcal{A}|+1} 1_{|\mathcal{A}|} 1_{|\mathcal{A}|}^{\mathrm{T}}) Q - Q^{'\,\mathrm{T}} (I_{|\mathcal{A}|} - \frac{1}{|\mathcal{A}|+1} 1_{|\mathcal{A}|} 1_{|\mathcal{A}|}^{\mathrm{T}}) Q^{'} \right\| \tag{39}$$

$$= \frac{1}{4} \left\| Q^{\mathrm{T}} Q - \frac{1}{|\mathcal{A}|+1} Q^{\mathrm{T}} 1_{|\mathcal{A}|} 1_{|\mathcal{A}|}^{\mathrm{T}} Q - Q^{'\,\mathrm{T}} Q^{'} + \frac{1}{|\mathcal{A}|+1} Q^{'\,\mathrm{T}} 1_{|\mathcal{A}|} 1_{|\mathcal{A}|}^{\mathrm{T}} Q^{'} \right\| \tag{40}$$

$$= \frac{1}{4} \left\| Q^{\mathrm{T}} Q - Q^{'\,\mathrm{T}} Q^{'} + \frac{1}{|\mathcal{A}|+1} ((1_{|\mathcal{A}|}^{\mathrm{T}} Q^{'})^{\mathrm{T}} (Q^{'\,\mathrm{T}} 1_{|\mathcal{A}|})^{\mathrm{T}} - (1_{|\mathcal{A}|}^{\mathrm{T}} Q)^{\mathrm{T}} (Q^{\mathrm{T}} 1_{|\mathcal{A}|})^{\mathrm{T}}) \right\| \tag{41}$$

$$= \frac{1}{4} \left\| Q^{\mathrm{T}} Q - Q^{'\,\mathrm{T}} Q^{'} + \frac{1}{|\mathcal{A}|+1} (Q^{'\,\mathrm{T}} Q^{'} - Q^{\mathrm{T}} Q) \right\| \tag{42}$$

$$\leq \frac{1}{4(|\mathcal{A}|+1)} \left\| |\mathcal{A}| Q^{\mathrm{T}} Q - |\mathcal{A}| Q^{'\,\mathrm{T}} Q^{'} \right\| \tag{43}$$

$$= \frac{|\mathcal{A}|}{4(|\mathcal{A}|+1)} \left\| Q^{\mathrm{T}} Q - Q^{'\,\mathrm{T}} Q^{'} \right\| \tag{44}$$

Using this in Eq. 38, one obtains the following,

$$= \frac{|\mathcal{A}|}{4(|\mathcal{A}|+1)} \left\| Q^{\mathrm{T}} Q - Q^{'\,\mathrm{T}} Q^{'} \right\| + b \left\| - (Q - Q^{'}) \right\| \tag{45}$$

$$= \frac{|\mathcal{A}|}{4(|\mathcal{A}|+1)} \left\| (Q - Q^{'})^{\mathrm{T}} (Q + Q^{'}) \right\| + b \left\| (Q - Q^{'}) \right\| \tag{46}$$

$$\leq \frac{|\mathcal{A}|}{4(|\mathcal{A}|+1)} \left\| (Q - Q^{'}) \right\| \left\| (Q + Q^{'}) \right\| + b \left\| (Q - Q^{'}) \right\| \tag{47}$$

$$= \left[\frac{|\mathcal{A}| \| Q + Q^{'} \|}{4(|\mathcal{A}|+1)} + b \right] \left\| Q - Q^{'} \right\| \tag{48}$$

Wherein the first inequality results from Cauchy-Schwarz inequality. The above result would be a contraction if $(\frac{|\mathcal{A}||Q+Q^{'}||}{4(|\mathcal{A}|+1)} + b) < 1$. More concretely,

$$\frac{|\mathcal{A}| \left\| Q + Q^{'} \right\|}{4(|\mathcal{A}|+1)} + b < 1 \tag{49}$$

$$= \left\| Q + Q^{'} \right\| < \frac{4(1-b)(|\mathcal{A}|+1)}{|\mathcal{A}|} \tag{50}$$

$$= \left\| Q + Q^{'} \right\| < \frac{4(1 - \hat{\mathbb{H}}\psi - g(\psi))(|\mathcal{A}|+1)}{|\mathcal{A}|} \tag{51}$$

Thus, a suitable choice of ψ would allow the upper bound to hold, making the Bohning approximation a contraction.

B.2 LOCAL CONVERGENCE OF $\mathrm{Bohn}(Q)$

The study of $\mathrm{Bohn}(Q)$ convergence involves isolating its optimization from the CQL($\mathcal{H}$) objective. Our approach closely follows the convergence analysis of convex quadratics in Nocedal & Wright (2006). Consider the gradient of $\mathrm{Bohn}(Q)$ for an estimate $\hat{Q}$,

$$\nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q}) = \hat{\mathbb{H}}\hat{Q} - b \tag{52}$$

The minimizer $\hat{Q} = Q$ is the unique solution to the linear system $\hat{\mathbb{H}}\hat{Q} = b$. One can compute the optimal step length η_k that minimizes $\mathrm{Bohn}(\hat{Q} - \eta\nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q}))$.

$$\mathrm{Bohn}(\hat{Q} - \eta\nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})) = \frac{1}{2}\left(\hat{Q} - \eta\nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})\right)^{\mathrm{T}} \hat{\mathbb{H}} \left(\hat{Q} - \eta\nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})\right) - b^{\mathrm{T}}\left(\hat{Q} - \eta\nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})\right)^{\mathrm{T}} + c \tag{53}$$

Setting the derivative w.r.t η to 0, we get,

$$\eta_k = \frac{\nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})^{\mathrm{T}} \nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})}{\nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})^{\mathrm{T}} \hat{\mathbb{H}} \nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})} \tag{54}$$

Utilizing this in the steepest descent update rule yields the following,

$$\hat{Q}^{k+1} = \hat{Q}^k - \left(\frac{\nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})^{\mathrm{T}} \nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})}{\nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})^{\mathrm{T}} \hat{\mathbb{H}} \nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})}\right) \nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q}) \tag{55}$$

We now study the rate of convergence. Using $\hat{\mathbb{H}}Q = b$, we have,

$$\frac{1}{2}\|\hat{Q} - Q\|_2^2 = \mathrm{Bohn}(\hat{Q}) - \mathrm{Bohn}(Q) \tag{56}$$

Utilizing Eq. 55 and noting that $\nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q}) = \hat{\mathbb{H}}(\hat{Q} - Q)$, we have the following,

$$\|\hat{Q}^{k+1} - Q\|_2^2 = \chi\|\hat{Q}^k - Q\|_2^2 \tag{57}$$

where $\chi = \left(1 - \frac{\left(\nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})^{\mathrm{T}} \nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})\right)^2}{\left(\nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})^{\mathrm{T}} \hat{\mathbb{H}} \nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})\right)\left(\nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})^{\mathrm{T}} \hat{\mathbb{H}}^{-1} \nabla_{\hat{Q}} \mathrm{Bohn}(\hat{Q})\right)}\right)$.

The above results in the following bound,

$$\|\hat{Q}^{k+1} - Q\|_2^2 \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^2 \|\hat{Q}^k - Q\|_2^2 \tag{58}$$

$$\|\hat{Q}^{k+1} - Q\|_2^2 \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^2 \|\hat{Q}^k - Q\|_2^2 \tag{59}$$

where $0 \leq \lambda_1 \leq \lambda_2 .. \leq \lambda_n$ are the eigenvalues of $\hat{\mathbb{H}}$ and $\kappa = \frac{\lambda_n}{\lambda_1}$.

Dependence of $\mathrm{Bohn}(Q)$ convergence on κ highlights its necessity for a well-conditioned objective. If κ is large, $\mathrm{Bohn}(Q)$ optimization will be ill-conditioned. On the other hand, if $\kappa \to 1$, $\left(\frac{\kappa-1}{\kappa+1}\right)^2 \to 0$, resulting in sound convergence.

We thus establish a lower bound on κ which captures estimation errors in Q values. Simply rearranging Eq. 59 yields the desired result,

$$\frac{\|\hat{Q}^{k+1} - Q\|_2^2}{\|\hat{Q}^k - Q\|_2^2} \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^2 \tag{60}$$

$$\frac{\|\hat{Q}^{k+1} - Q\|_2}{\|\hat{Q}^k - Q\|_2} + 1 \leq \kappa - \kappa\left(\frac{\|\hat{Q}^{k+1} - Q\|_2}{\|\hat{Q}^k - Q\|_2}\right) \tag{61}$$

$$\left|\frac{1 + \frac{\|\hat{Q}^{k+1} - Q\|_2}{\|\hat{Q}^k - Q\|_2}}{1 - \frac{\|\hat{Q}^{k+1} - Q\|_2}{\|\hat{Q}^k - Q\|_2}}\right| \leq \kappa \tag{62}$$

$$\left|\frac{\|\hat{Q}^k - Q\|_2 + \|\hat{Q}^{k+1} - Q\|_2}{\|\hat{Q}^k - Q\|_2 - \|\hat{Q}^{k+1} - Q\|_2}\right| \leq \kappa \tag{63}$$

C CONSERVATISM WITH BEHAVIOR CLONING

C.1 LOWER BOUND ON MUTUAL INFORMATION

Here we will derive the lower bound on Mutual Information between the agent's and behavior policy $\mathcal{MI}\{\pi; \hat{\pi}_\beta\}$. Denote $p(\pi, \hat{\pi}_\beta)$ as the joint probability distribution of actions sampled from π and $\hat{\pi}_\beta$, expanding $\mathcal{MI}\{\pi; \hat{\pi}_\beta\}$ as per definition,

$$\mathcal{MI}\{\pi; \hat{\pi}_\beta\} = \int p(\pi, \hat{\pi}_\beta) \log \left(\frac{p(\pi, \hat{\pi}_\beta)}{p(\pi)p(\hat{\pi}_\beta)} \right) d\pi d\hat{\pi}_\beta \tag{64}$$

$$= \int p(\pi, \hat{\pi}_\beta) \log \left(\frac{p(\pi|\hat{\pi}_\beta)p(\hat{\pi}_\beta)}{p(\pi)p(\hat{\pi}_\beta)} \right) d\pi d\hat{\pi}_\beta \tag{65}$$

$$= \int p(\pi, \hat{\pi}_\beta) \log \left(\frac{p(\pi|\hat{\pi}_\beta)}{p(\pi)} \right) d\pi d\hat{\pi}_\beta \tag{66}$$

Upon seeking a tractable approximation $q(\pi|\hat{\pi}_\beta)$ to the posterior $p(\pi|\hat{\pi}_\beta)$,

$$= \int p(\pi, \hat{\pi}_\beta) \left(\log \frac{q(\pi|\hat{\pi}_\beta)}{p(\pi)} + \log \frac{p(\pi|\hat{\pi}_\beta)}{q(\pi|\hat{\pi}_\beta)} \right) d\pi d\hat{\pi}_\beta \tag{67}$$

$$= \int p(\pi, \hat{\pi}_\beta) \log \left(\frac{q(\pi|\hat{\pi}_\beta)}{p(\pi)} \right) d\pi d\hat{\pi}_\beta + \int p(\pi, \hat{\pi}_\beta) \log \left(\frac{p(\pi|\hat{\pi}_\beta)}{q(\pi|\hat{\pi}_\beta)} \right) d\pi d\hat{\pi}_\beta \tag{68}$$

$$= \int p(\pi, \hat{\pi}_\beta) \log \left(\frac{q(\pi|\hat{\pi}_\beta)}{p(\pi)} \right) d\pi d\hat{\pi}_\beta + \int p(\pi|\hat{\pi}_\beta)p(\hat{\pi}_\beta) \log \left(\frac{p(\pi|\hat{\pi}_\beta)}{q(\pi|\hat{\pi}_\beta)} \right) d\pi d\hat{\pi}_\beta \tag{69}$$

The second term simplifies to $\mathbb{E}_{\hat{\pi}_\beta}[\mathbb{KL}(p(\pi|\hat{\pi}_\beta)||q(\pi|\hat{\pi}_\beta))]$ which must be non-negative,

$$\geq \int p(\pi, \hat{\pi}_\beta) \log q(\pi|\hat{\pi}_\beta) d\pi d\hat{\pi}_\beta - \int p(\pi|\hat{\pi}_\beta)p(\hat{\pi}_\beta) \log p(\pi) d\pi d\hat{\pi}_\beta \tag{70}$$

Marginalizing the second term over $\hat{\pi}_\beta$ yields the lower bound,

$$= \mathbb{E}_{p(\pi, \hat{\pi}_\beta)}[\log q(\pi|\hat{\pi}_\beta)] + \mathcal{H}(p(\pi)) \tag{71}$$

C.2 PRACTICAL IMPLEMENTATION

This section derives the off-policy variant of IM formulation incorporating the $\mathcal{MI}\{\pi; \hat{\pi}_\beta\}$ lower bound. Begin by explicitly writing the augmented Q value function with r_t as the reward at step t,

$$Q = \mathbb{E}_{s_t \sim \mathcal{D}, a_t \sim \pi} \left[\sum_{t=1}^{T} \gamma^{t-1} r_t \right] \tag{72}$$

Expanding the Q value recursion till the termination step T,

$$r_1 + \mathbb{E}_{p(\pi, \hat{\pi}_\beta)}[\log q_1(\pi|\hat{\pi}_\beta)] + \gamma r_2 + \gamma \mathbb{E}_{p(\pi, \hat{\pi}_\beta)}[\log q_2(\pi|\hat{\pi}_\beta)] + \ldots + \gamma^{T-1} r_T + \gamma^{T-1} \mathbb{E}_{p(\pi, \hat{\pi}_\beta)}[\log q_T(\pi|\hat{\pi}_\beta)] \tag{73}$$

Grouping the reward and log terms separately,

$$Q = \left(r_1 + \gamma r_2 + \ldots + \gamma^{T-1} r_T\right) + \left(\mathbb{E}_{p(\pi, \hat{\pi}_\beta)}[\log q_1(\pi|\hat{\pi}_\beta)] + \gamma \mathbb{E}_{p(\pi, \hat{\pi}_\beta)}[\log q_2(\pi|\hat{\pi}_\beta)] + \ldots + \gamma^{T-1} \mathbb{E}_{p(\pi, \hat{\pi}_\beta)}[\log q_T(\pi|\hat{\pi}_\beta)]\right) \tag{74}$$

where the discounted return in the first term simplifies to $\hat{Q}^k$ and the second term to an expected sum of logs,

$$Q = \hat{Q}^k + \mathbb{E}_{p(\pi, \hat{\pi}_\beta)} \left[\sum_{t=1}^{T} \gamma^{t-1} \log q_t(\pi|\hat{\pi}_\beta) \right] \tag{75}$$

Moving γ^{t-1} inside the log,

$$Q = \hat{Q}^k + \mathbb{E}_{p(\pi, \hat{\pi}_\beta)} \left[\sum_{t=1}^{T} \log q_t(\pi|\hat{\pi}_\beta)^{\gamma^{t-1}} \right] \tag{76}$$

Finally, moving the sum inside log yields the result.

$$Q = \hat{Q}^k + \mathbb{E}_{p(\pi, \hat{\pi}_\beta)} \left[\log \prod_{t=1}^{T} q_t(\pi|\hat{\pi}_\beta)^{\gamma^{t-1}} \right] \tag{77}$$

C.3 CONVERGENCE BOUND

We theoretically show that CQL-IM converges faster to true Q when compared to CQL($\mathcal{H}$). Begin by considering the gap between underestimated values $\hat{\mathcal{B}}^{\pi}\hat{Q}^{k} - \alpha\left(\frac{\text{softmax}(\hat{Q})}{\hat{\pi}_{\beta}} - 1\right)$ and true values Q,

$$\left\|\hat{\mathcal{B}}^{\pi}\hat{Q}^{k} - \alpha\left(\frac{\text{softmax}(\hat{Q})}{\hat{\pi}_{\beta}} - 1\right) - Q\right\|_2 \tag{78}$$

Using the fixed point property of Bellman operator,

$$\left\|\hat{Q}^{k} - Q\alpha\left(\frac{\text{softmax}(\hat{Q})}{\hat{\pi}_{\beta}} - 1\right)\right\|_2 \tag{79}$$

Using the triangle inequality,

$$\leq \|\hat{Q}^{k} - Q\|_2 + \alpha\left\|\frac{\text{softmax}(\hat{Q})}{\hat{\pi}_{\beta}} - 1\right\|_2 \tag{80}$$

We now consider the two terms separately. For the first term we have, following the results of approximation value iteration convergence (Bertsekas & Tsitsiklis, 1995),

$$\leq \|(\hat{\mathcal{B}}^{\pi})^{k}\hat{Q}^{0} - Q\|_2 \tag{81}$$

$$\gamma^{k}\|\hat{Q}^{0} - \hat{Q} + \hat{Q} - Q\|_2 \tag{82}$$

Here, $\hat{Q}$ denotes an approximation to the true Q function.

$$\leq \gamma^{k}\|\hat{Q}^{0} - \hat{Q}\|_2 + \gamma^{k}\|\hat{Q} - Q\|_2 \tag{83}$$

which results in the following bound (Bertsekas, 2018),

$$\leq \gamma^{k}\sqrt{r_{\max}} + \gamma^{k}\sqrt{\frac{r_{\max}|\mathcal{S}|}{1-\gamma}} \tag{84}$$

The bound denotes the geometric convergence of policy optimization process. With $\mathcal{O}(\gamma^{k})$, policy updates approach local convergence asymptotically in increasing number of iterations k.

Considering the second term, we have,

$$\alpha\left\|\frac{\exp(\hat{Q})}{\hat{\pi}_{\beta}\sum_{a\in\mathcal{A}}\exp(\hat{Q}(s,a))} - 1\right\|_2 \tag{85}$$

which is bounded as follows,

$$\alpha\left\|\frac{\exp(\hat{Q})}{\sum_{a\in\mathcal{A}}\exp(\hat{Q}(s,a))} - 1\right\|_2 \leq \alpha\left\|\frac{\exp(\hat{Q})}{\hat{\pi}_{\beta}\sum_{a\in\mathcal{A}}\exp(\hat{Q}(s,a))} - 1\right\|_2 \leq \infty \tag{86}$$

Irrespective of the value of $\hat{\pi}_{\beta}$, exponential values of $\hat{Q}$ do not decay. This leaves the error to persist at an $\mathcal{O}(\exp(\hat{Q}))$ rate.

We now repeat the same computation for CQL-IM. Explicitly stating the CQL-IM underestimations,

$$\hat{Q} = \hat{\mathcal{B}}^{\pi}\hat{Q}^{k} - \alpha\left(\frac{\text{softmax}(\hat{Q} + \log q)}{\hat{\pi}_{\beta}} - 1\right) - \log q \tag{87}$$

Utilizing this to compute the difference from true Q values,

$$\left\|\hat{\mathcal{B}}^{\pi}\hat{Q}^{k} - \alpha\left(\frac{\text{softmax}(\hat{Q} + \log q)}{\hat{\pi}_{\beta}} - 1\right) - \log q - Q\right\|_2 \tag{88}$$

As before, using the fixed point of Bellman operator and separating the terms using triangle inequality,

$$\|\hat{Q}^{k} - Q\|_2 + \left\|\alpha\left(\frac{\text{softmax}(\hat{Q} + \log q)}{\hat{\pi}_{\beta}} - 1\right) + \log q\right\|_2 \tag{89}$$

Solving for the first term results in $\mathcal{O}(\gamma^k)$ convergence for policy updates,

$$\|\hat{Q}^k - Q\|_2 \leq \gamma^k \sqrt{r_{\max}} + \gamma^k \sqrt{\frac{r_{\max}|\mathcal{S}|}{1-\gamma}} \tag{90}$$

And for the second term, we have,

$$\left\| \alpha \left(\frac{\exp(\hat{Q} + \log q)}{\hat{\pi}_\beta \sum_{a \in \mathcal{A}} \exp(\hat{Q}(s,a) + \log q(s,a))} - 1 \right) + \log q \right\|_2 \tag{91}$$

which yields the following after manipulation,

$$\left\| \alpha \left(\frac{\hat{\pi}_{\text{BC}} \exp(\hat{Q}) q}{\hat{\pi}_\beta \sum_{a \in \mathcal{A}} \exp(Q(\hat{s},a)) q(s,a)} - 1 \right) + \log q \right\|_2 \tag{92}$$

Denoting $\hat{\pi}_{\text{BC}} = \frac{q}{\sum_{a \in \mathcal{A}} q(s,a)}$ gives us the following result.

$$\left\| \alpha \left(\frac{\text{softmax}(\hat{Q})}{\hat{\pi}_\beta} - 1 \right) + \log q \right\|_2 \tag{93}$$

Compared to Eq. 86, Eq. 93 presents the additional $\hat{\pi}_{\text{BC}}$ and $\log q$ terms. Here, $\log q$ presents bounded values and hence, bounded convergence following prior results (Cai et al., 2019; Yin et al., 2021a). On the other hand, $\hat{\pi}_{\text{BC}}$ improves the rate of convergence. Note that $\hat{\pi}_{\text{BC}} \in (0, 1]$ depends on each state-action pair as $\text{softmax}(\hat{Q})$ and acts as a scaling term for these exponential values. This way, the effective rate reduces from $\mathcal{O}(\exp(\hat{Q}))$ of Eq. 86 to $\mathcal{O}(\hat{\pi}_{\text{BC}} \exp(\hat{Q}))$. Thus, CQL-IM converges faster than CQL($\mathcal{H}$) to true Q values.

C.4 Stability Guarantee

Next, we show that CQL-IM is ϵ-stable when compared to CQL($\mathcal{H}$). To derive this result, we assume that the gradient of Q values for both objectives is bounded by a constant ϵ and Q values can locally approximate the Bellman update, i.e.- at convergence $Q \approx \hat{\mathcal{B}}^\pi \hat{Q}^k$.

We need to show that the gradient of CQL-IM is strictly negative definite and lies in the open left-half plane. Note that this is one of the conditions for Lyapunov stability (Bhatia & Szegö, 2002), a stronger condition for nonlinear systems.

First, differentiating CQL($\mathcal{H}$) w.r.t parameters of Q, we get,

$$\left(\alpha \left[\frac{\exp(Q)}{\sum_a \exp(Q)} - \hat{\pi}_\beta \right] + (Q - \hat{\mathcal{B}}^\pi \hat{Q}^k) \hat{\pi}_\beta \right) \nabla Q \tag{94}$$

Since $\nabla Q \leq \epsilon$ and $Q \approx \hat{\mathcal{B}}^\pi \hat{Q}^k$ at convergence, the equation reduces to $\alpha \text{softmax}(Q) - \hat{\pi}_\beta$. This results in 0 for extreme values of $\text{softmax}(Q)$ and $\hat{\pi}_\beta$ ($\text{softmax}(Q) = 1$ and $\hat{\pi}_\beta = 1$) indicating that the gradient is not strictly negative. Intuitively, the result denotes that the critic update of CQL($\mathcal{H}$) may not alaways be a descent direction.

Considering CQL-IM and differentiating the objective w.r.t parameters of Q,

$$\left(\alpha \left[\frac{\exp(Q + \log q)}{\sum_a \exp(Q + \log q)} - \hat{\pi}_\beta \right] + (Q + \log q - \hat{\mathcal{B}}^\pi \hat{Q}^k) \hat{\pi}_\beta \right) \nabla Q \tag{95}$$

Under the same conditions of bounded gradient $\nabla Q \leq \epsilon$ and approximate TD convergence $Q \approx \hat{\mathcal{B}}^\pi \hat{Q}^k$, Eq. 95 simplifies to $-\log q$. This indicates that CQL-IM is ϵ-stable. Upto a constant ϵ, the gradient always corresponds to a descent direction when $Q \approx \hat{\mathcal{B}}^\pi \hat{Q}^k$.

D Conservatism in the Conjugate Space

D.1 CQL($\mathcal{H}$) as Entropy Minimization

Optimizing Q values in the value function space does not necessarily highlight its consequences in alternative basis spaces. For instance, policy gradient methods optimize policies directly in the

policy space which implicitly explains asymptotic growth of the value function (Sutton & Barto, 2018). However, a converse for the same is hard to accomplish. A similar argument can be made for CQL($\mathcal{H}$) upon realizing its implications in the conjugate space.

As a starting point, consider the Legendre transform (convex conjugate of a differentiable function) (Boyd & Vandenberghe, 2004) of $\text{lse}(Q)$. The $\text{lse}(Q)$ function is convex in $\mathbb{R}^{|\mathcal{A}|}$ and has $\text{softmax}(Q)$ as its gradient map (Gao & Pavel, 2017). Denote $f^*(Q_{\text{conj}})$ as the Legendre transform of $f(Q)$ expressed in Eq 96.

$$f^*(Q_{\text{conj}}) = \sup_{Q \in \text{dom } f} (Q_{\text{conj}}^{\text{T}} Q - f(Q)) \; ; \; Q_{\text{conj}} = \nabla_Q f(Q) \tag{96}$$

The quantity $Q_{\text{conj}} = \nabla_Q f(Q)$ represents the Q function in conjugate space with $\text{dom } f$ representing the feasible set of Q values. Substituting $f(Q) = \text{lse}(Q)$ to obtain the conjugate $f^*(Q_{\text{conj}})$,

$$f(Q) = \text{lse}(Q) \implies f^*(Q_{\text{conj}}) = \sum_a Q_{\text{conj}}(s,a) \log(Q_{\text{conj}}(s,a)) \tag{97}$$

Eq. 97 establishes a direct link between $f^*(Q_{\text{conj}})$ and the entropy of conjugate value distribution. $f^*(Q_{\text{conj}})$ exactly represents the negative entropy of Q_{conj} function restricted to $|\mathcal{A}|$ dimensional simplex $\triangle^{|\mathcal{A}|}$, i.e.- $-\mathcal{H}(Q_{\text{conj}}) \in \triangle^{|\mathcal{A}|}$. One can construct an auxilary objective consisting of $-\mathcal{H}(Q_{\text{conj}})$ which is easier to optimize in comparison to $\text{lse}(Q)$. A simple inspection of the Fenchel-Young Inequality (Boyd & Vandenberghe, 2004; Kumar et al., 2020a) $f(Q) + f^*(Q_{\text{conj}}) \geq Q_{\text{conj}}^{\text{T}} Q$ reveals that $f^*(Q_{\text{conj}}) \geq -f(Q)$. Utilizing this for $\text{lse}(Q)$ presents a tractable entropy minimization objective in the conjugate space, $\text{lse}(Q) \geq \min_{Q_{\text{conj}}} \mathcal{H}(Q_{\text{conj}})$,

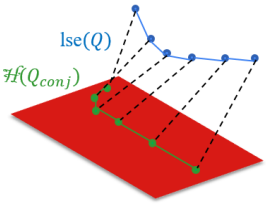

Figure 8: Intuition behind $\min_{Q_{\text{conj}}} \mathcal{H}(Q_{\text{conj}})$

$$f^*(\text{y}) + f(\text{x}) \geq \text{x}^{\text{T}}\text{y} \;\; , \forall \text{x} \in \text{dom } f, \forall \text{y} \in \text{dom } f^* \tag{98}$$

$$= -\mathcal{H}(Q_{\text{conj}}) \geq -\text{lse}(Q) + Q_{\text{conj}}^{\text{T}} Q \;\; , \forall Q, \forall Q_{\text{conj}} \tag{99}$$

$$= -\mathcal{H}(Q_{\text{conj}}) \geq -\text{lse}(Q) \;\; , \forall Q, \forall Q_{\text{conj}} \tag{100}$$

$$= \mathcal{H}(Q_{\text{conj}}) \leq \text{lse}(Q) \;\; , \forall Q, \forall Q_{\text{conj}} \tag{101}$$

$$= \min_{Q_{\text{conj}}} \mathcal{H}(Q_{\text{conj}}) \leq \text{lse}(Q) \;\; , \forall Q \tag{102}$$

That is, minimizing the entropy in conjugate space is a tractable lower bound objective on the $\text{lse}(Q)$ function. Utilizing this insight in CQL($\mathcal{H}$) yields the auxilary entropy minimization objective.

$$\min_Q \alpha \, \mathbb{E}_{s \sim \mathcal{D}} \underbrace{\left[\min_{Q_{\text{conj}}} \mathcal{H}(Q_{\text{conj}}) \right]}_{\text{approximation}} - \underbrace{\mathbb{E}_{a \sim \hat{\pi}_\beta}[Q(s,a)]}_{\text{expected value}} + \underbrace{\frac{1}{2} \mathbb{E}_{s,a,s' \sim \mathcal{D}} \left[\left(Q - \hat{\mathcal{B}}^\pi \hat{Q}^k \right)^2 \right]}_{\text{empirical Bellman error}} \tag{103}$$

This is illustrated in Fig. 8. In addition to the prior expected value and empirical Bellman error terms, Eq. 103 now additionally represents an entropy minimization objective $\min_{Q_{\text{conj}}} \mathcal{H}(Q_{\text{conj}})$ in the conjugate space. Intuitively, the objective enforces an agent to minimize uncertainty within the learning signal arising from batched data transitions. Note that this is in addition to predefined maximum entropy regularization objectives which are a common precept in learning behaviors (Haarnoja et al., 2018; Singh et al., 2020; Kumar et al., 2019; Levine et al., 2020). *But how does minimization of entropy link to underestimations in the conservative learning framework?*

The insight can be further studied by optimizing over Eq. 103 and setting its derivatives w.r.t. Q and Q_{conj} to 0 while adhering to the KKT conditions. We first optimize over $\min_{Q_{\text{conj}}} \mathcal{H}(Q_{\text{conj}})$ to obtain the optimal $Q_{\text{conj}} = \frac{1}{e}$. Utilizing this to solve the global optimization problem, one obtains the following expression,

$$-\alpha \hat{\pi}_\beta + \hat{\pi}_\beta \left(Q - \hat{\mathcal{B}}^\pi \hat{Q}^k \right) = 0 \implies Q = \hat{\mathcal{B}}^\pi \hat{Q}^k + \alpha \tag{104}$$

Eq. 104 paints a direct significance between the entropy minimization scheme and Q value estimates. The second term in Eq. 104 represents overestimations at each subsequent policy iteration which regularize the Bellman estimate $\hat{\mathcal{B}}^\pi \hat{Q}^k$. This leads to an accumulation of positive biases over policy updates which directly depend on α. Since α is a free parameter, its value may be tuned for reducing overestimation errors. Additionally, Kumar et al. (2020b) note that in the limit of infinite data, small values of α suffice to yield a lower bound. Thus, elimination of overoptimistic estimates require either a hand-engineered value or infinite data samples.

D.2 Explaining High Variance

An alternate process to understand conservatism in conjugate space is by considering the Legendre transform of the complete CQL($\mathcal{H}$) objective. Similar to initial steps, consider $f(Q_{\text{conj}})$ as the CQL($\mathcal{H}$) objective in Eq. 1. Upon computing $Q_{\text{conj}} = \nabla_Q f(Q)$, one arrives at a value for Q_{conj} which depends on Q and $\hat{\pi}_\beta$,

$$Q_{\text{conj}} = \nabla_Q f(Q) = \alpha \left[\text{softmax}(Q(s,a)) - \hat{\pi}_\beta(a|s)\right] + (Q(s,a) - \hat{\mathcal{B}}^\pi \hat{Q}^k)\hat{\pi}_\beta(a|s) \tag{105}$$

$$= (\alpha \,\text{softmax}(Q(s,a)) + Q(s,a)\hat{\pi}_\beta(a|s)) - \hat{\pi}_\beta(a|s)\left[\alpha + \hat{\mathcal{B}}^\pi \hat{Q}^k\right] \tag{106}$$

Following the precept from CQL (Kumar et al., 2020b) and only considering the setting wherein values of α are small,

$$Q_{\text{conj}} = \hat{\pi}_\beta(a|s)\left(Q(s,a) - \hat{\mathcal{B}}^\pi \hat{Q}^k\right) \tag{107}$$

Intuitively, Q_{conj} denotes the gap between current Q and empirical Bellman estimate $\hat{\mathcal{B}}^\pi \hat{Q}^k$ which is scaled by the likelihood of actions under behavior policy $\hat{\pi}_\beta$. The expression quantifies *how far is the current estimate from previous Bellman estimate under the behavior policy distribution*.

Rearranging Eq. 107 yields $Q = \frac{Q_{\text{conj}}}{\hat{\pi}_\beta} + \hat{\mathcal{B}}^\pi \hat{Q}^k$. Substituting this in CQL($\mathcal{H}$) and optimizing over Q_{conj} by setting the derivative w.r.t. Q_{conj} to 0 yields the following,

$$\min_{Q_{\text{conj}}} Q_{\text{conj}}\left(\frac{Q_{\text{conj}}}{\hat{\pi}_\beta} + \hat{\mathcal{B}}^\pi \hat{Q}^k\right) - \alpha\, \mathbb{E}_{s\sim\mathcal{D}}\left[\text{lse}\left(\frac{Q_{\text{conj}}}{\hat{\pi}_\beta} + \hat{\mathcal{B}}^\pi \hat{Q}^k\right) - \mathbb{E}_{a\sim\hat{\pi}_\beta}\left[\frac{Q_{\text{conj}}}{\hat{\pi}_\beta} + \hat{\mathcal{B}}^\pi \hat{Q}^k\right]\right] - \frac{1}{2}\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(\frac{Q_{\text{conj}}}{\hat{\pi}_\beta}\right)^2\right] \tag{108}$$

$$\left(\frac{Q_{\text{conj}}}{\hat{\pi}_\beta} + \hat{\mathcal{B}}^\pi \hat{Q}^k\right) + \frac{Q_{\text{conj}}}{\hat{\pi}_\beta} - \alpha\left(\frac{1}{\hat{\pi}_\beta}\text{softmax}\left(\frac{Q_{\text{conj}}}{\hat{\pi}_\beta} + \hat{\mathcal{B}}^\pi \hat{Q}^k\right) - 1\right) - \frac{Q_{\text{conj}}}{\hat{\pi}_\beta} = 0 \tag{109}$$

$$\frac{Q_{\text{conj}}}{\hat{\pi}_\beta} + \hat{\mathcal{B}}^\pi \hat{Q}^k = \alpha\left(\frac{1}{\hat{\pi}_\beta}\text{softmax}\left(\frac{Q_{\text{conj}}}{\hat{\pi}_\beta} + \hat{\mathcal{B}}^\pi \hat{Q}^k\right) - 1\right) \tag{110}$$

$$Q_{\text{conj}} = \underbrace{\hat{\pi}_\beta}_{\text{scaling}}\ \underbrace{|\hat{\mathcal{B}}^\pi \hat{Q}^k|}_{\text{estimate}} + \underbrace{\alpha}_{\text{tradeoff}}\ \underbrace{\left[\text{softmax}\left(\frac{Q_{\text{conj}}}{\hat{\pi}_\beta} + \hat{\mathcal{B}}^\pi \hat{Q}^k\right) - \hat{\pi}_\beta(a|s)\right]}_{\text{underestimation}} \tag{111}$$

Eq. 111 consists of 4 terms. The first term indicates the rate of decay at which subsequent estimates in the second term diminish towards finitely small values. The third term forms the tradeoff factor denoting the scale at which CQL underestimates values. Note the dependence of the third term on constant α which, as in the previous analysis, intuitively suggests that uncertainty may be traded off for estimation errors (Kumar et al., 2020b). Lastly, the fourth term indicates potential underestimation of Q values arising from the squashed $\text{softmax}(Q)$ distribution. Since $\mu \propto \exp(Q)$, the gap $\text{softmax}(Q) - \hat{\pi}_\beta$ intuitively represents a change in $\mu - \hat{\pi}_\beta$. This highlights a set of fluctuating Q values which lead to increasing/decreasing $\text{softmax}(Q)$ resulting in high-variance updates.

E Extensions to Model-based Setting

This section highlights the extension of our analysis to the setting of Conservative Model-based Policy Optimization (COMBO) (Yu et al., 2021b). In order to strictly adhere to the notation of

COMBO, we slightly abuse our notation and denote the state marginal distribution as $d^{\pi}_{\mathcal{M}}(s)$ corresponding to the distribution of states obtained by rolling out policy π in the MDP $\mathcal{M}$. Further, define the MDP $\overline{\mathcal{M}}$ as the empirical MDP induced by the dataset $\mathcal{D}$ corresponding to the behavior policy $\hat{\pi}_{\beta}$. The model-based setting trains a dynamics model $\widehat{T}$ such that the objective corresponds to maximum likelihood estimation $\min_{\widehat{T}} \mathbb{E}_{s,a,s' \sim \mathcal{D}} \left[\log \widehat{T}(s'|s,a) \right]$. Corresponding to the model, one can construct the learned MDP $\widehat{\mathcal{M}} = (\mathcal{S}, \mathcal{A}, r, \widehat{T}, \gamma)$. COMBO makes use of a data aggregation framework wherein at each iteration, the agent performs k-step rollouts using $\widehat{T}$ starting from state $s \in \mathcal{D}$ and adds the data generated by the model to a separate dataset $\mathcal{D}_{\text{model}}$. The policy is optimized corresponding to a batch of data sampled from $\mathcal{D} \cup \mathcal{D}_{\text{model}}$ wherein each datapoint is drawn from $\mathcal{D}$ with probability $f \in [0,1]$ and $\mathcal{D}_{\text{model}}$ with probability $1-f \in [0,1]$. The empirical objective utilized by COMBO (Yu et al., 2021b) (Eq. 23) is based on CQL($\mathcal{H}$) and is expressed as per Eq. 112.

$$\hat{Q}^k \leftarrow \arg\min_{Q} \alpha \left(\mathbb{E}_{s \sim d^{\pi}_{\widehat{\mathcal{M}}}} \left[\text{lse}(Q) \right] - \mathbb{E}_{s,a \sim \mathcal{D}}[Q(s,a)] \right) + \frac{1}{2} \mathbb{E}_{s,a,s' \sim d_f} \left[\left(Q(s,a) - \hat{\mathcal{B}}^{\pi} \hat{Q}^k \right)^2 \right] \tag{112}$$

Eq. 112 consists of $d^{\pi}_{\widehat{\mathcal{M}}}$ as the discounted state marginal distribution obtained when policy π is executed in the MDP $\widehat{\mathcal{M}}$ and d_f as an f-interpolation between the offline dataset and model rollouts $d_f = fd + (1-f)d^{\pi}_{\widehat{\mathcal{M}}}$ wherein the notation uses d as a shorthand for the discounted state marginal distribution corresponding to the dataset $d^{\hat{\pi}_{\beta}}(s_t)$.

E.1 Underestimation in COMBO

Prior to our analysis, we summarize the conservative nature of COMBO. Upon optimizing over Eq. 112 by setting the derivative w.r.t. Q to 0, one obtains,

$$\alpha \left[\text{softmax}(Q) - d \right] + d_f (Q - \hat{\mathcal{B}}^{\pi} \hat{Q}^k) = 0 \tag{113}$$

$$Q = \hat{\mathcal{B}}^{\pi} \hat{Q}^k - \alpha \left(\frac{\text{softmax}(Q) - d}{d_f} \right) \tag{114}$$

Eq. 114 denotes that with high probability, COMBO underestimates its subsequent Q values when $\text{softmax}(Q) > d$.

E.2 COMBO in Conjugate Space

We analyze COMBO in the conjugate space by substituting the Legendre transform of $\text{lse}(Q)$ in Eq. 112,

$$\hat{Q}^{k+1} \leftarrow \arg\min_{Q} \alpha \left(\mathbb{E}_{s \sim d^{\pi}_{\widehat{\mathcal{M}}}} \left[\min_{Q_{\text{conj}}} \mathcal{H}(Q_{\text{conj}}) \right] - \mathbb{E}_{s,a \sim \mathcal{D}}[Q(s,a)] \right) + \frac{1}{2} \mathbb{E}_{s,a,s' \sim d_f} \left[\left(Q - \hat{\mathcal{B}}^{\pi} \hat{Q}^k \right)^2 \right] \tag{115}$$

Similar to CQL($\mathcal{H}$), Eq. 115 presents an entropy minimization objective. We first optimize $\min_{Q_{\text{conj}}} \mathcal{H}(Q_{\text{conj}})$ by setting its derivative to 0 and obtain the optimal Q_{conj} value as $\frac{1}{e}$. Optimizing Eq. 115 by using the KKT conditions for $\min_{Q_{\text{conj}}} \mathcal{H}(Q_{\text{conj}})$ and setting derivative w.r.t Q to 0,

$$-d\alpha + d_f \left(Q - \hat{\mathcal{B}}^{\pi} \hat{Q}^k \right) = 0 \tag{116}$$

$$Q = \hat{\mathcal{B}}^{\pi} \hat{Q}^k + \alpha \frac{d}{d_f} \tag{117}$$

Eq. 117 highlights the direct dependence of Q values on overestimations.

E.3 COMBO with Approximations

We evaluate COMBO by fitting a standard tractable approximation to $\text{lse}(Q)$ as in the case of CQL($\mathcal{H}$),

$$\text{lse}(Q) \leq \frac{1}{2} Q^{\mathrm{T}} \hat{\mathbb{H}} Q - bQ + c \tag{118}$$

The above results in Eq. 119.

$$\hat{Q}^{k+1} \leftarrow \arg\min_{Q} \alpha \left(\mathbb{E}_{s\sim d^{\pi}_{\widehat{\mathcal{M}}}}\left[\text{Bohn}(Q)\right] - \mathbb{E}_{s,a\sim\mathcal{D}}[Q(s,a)]\right) + \frac{1}{2}\mathbb{E}_{s,a,s'\sim d_f}\left[\left(Q(s,a) - \hat{\mathcal{B}}^{\pi}\hat{Q}^{k}\right)^2\right] \tag{119}$$

Optimizing Eq. 119 w.r.t. Q and setting the derivative to 0,

$$\alpha\left(\hat{\mathbb{H}}(Q+\psi) - \text{softmax}(\psi) - d\right) + d_f\left(Q - \hat{\mathcal{B}}^{\pi}\hat{Q}^{k}\right) = 0 \tag{120}$$

$$Q\left(\alpha\hat{\mathbb{H}} + d_f\right) = d_f\hat{\mathcal{B}}^{\pi}\hat{Q}^{k} - \alpha\left[\hat{\mathbb{H}}\psi - \text{softmax}(\psi) - d\right] \tag{121}$$

$$Q = \left(\frac{d_f}{\alpha\hat{\mathbb{H}} + d_f}\right)\hat{\mathcal{B}}^{\pi}\hat{Q}^{k} - \left(\frac{\alpha}{\alpha\hat{\mathbb{H}} + d_f}\right)\left[\hat{\mathbb{H}}\psi - \text{softmax}(\psi) - d\right] \tag{122}$$

Eq. 122 presents the same scaling problem as found in CQL($\mathcal{H}$) analysis.

F Additional Bounds

This section further highlights underestimations in CQL while obtaining tighter bounds on CQL($\mathcal{H}$). Prior to the complete derivation, we will show that the CQL($\mathcal{H}$) variant may not always underestimate the true Q values.

F.1 Underestimation in CQL

Here we present the proof of underestimation in CQL Q values as derived in (Kumar et al., 2020b). Note that the original CQL objective, in the absence of regularization, is given by Eq. 123.

$$\arg\min_{Q} \alpha\, \mathbb{E}_{\mu}\left[Q(s,a)\right] + \frac{1}{2}\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(Q(s,a) - \hat{\mathcal{B}}^{\pi}\hat{Q}^{k}\right)^2\right] \tag{123}$$

Optimizing over Eq. 123 by taking the derivative w.r.t Q and setting it to 0, one obtains,

$$\alpha\mu + (Q - \hat{\mathcal{B}}^{\pi}\hat{Q}^{k})\hat{\pi}_{\beta} = 0 \tag{124}$$

$$Q = \hat{\mathcal{B}}^{\pi}\hat{Q}^{k} - \alpha\frac{\mu}{\hat{\pi}_{\beta}} \tag{125}$$

Since $\hat{\mathcal{B}}^{\pi}\hat{Q}^{k} \geq Q$, CQL underestimates Q values at each subsequent iteration.

F.2 Underestimation in CQL($\mathcal{H}$)

We now follow the same process as described above and show that CQL($\mathcal{H}$) does not always underestimate Q values. Recall that the CQL($\mathcal{H}$) objective is expressed as per Eq. 126.

$$\min_{Q} \alpha\, \mathbb{E}_{s\sim\mathcal{D}}\left[\log\sum_{a}\exp\left(Q\right) - \mathbb{E}_{a\sim\hat{\pi}_{\beta}}\left[Q(s,a)\right]\right] + \frac{1}{2}\mathbb{E}_{s,a,s'\sim\mathcal{D}}\left[\left(Q(s,a) - \hat{\mathcal{B}}^{\pi}\hat{Q}^{k}\right)^2\right] \tag{126}$$

Optimizing over Eq. 126 by setting the derivative w.r.t Q to 0,

$$\alpha\left[\frac{\exp(Q)}{\sum_a \exp(Q)} - \hat{\pi}_{\beta}\right] + (Q - \hat{\mathcal{B}}^{\pi}\hat{Q}^{k})\hat{\pi}_{\beta} = 0 \tag{127}$$

$$= \alpha\left[\frac{\text{softmax}(Q)}{\hat{\pi}_{\beta}} - 1\right] + Q = \hat{\mathcal{B}}^{\pi}\hat{Q}^{k}$$

$$= Q = \hat{\mathcal{B}}^{\pi}\hat{Q}^{k} - \alpha\left[\frac{\text{softmax}(Q)}{\hat{\pi}_{\beta}} - 1\right] \tag{128}$$

Since $\hat{\mathcal{B}}^{\pi}\hat{Q}^{k} \geq Q$ only if $\text{softmax}(Q) \geq \hat{\pi}_{\beta}$, the CQL($\mathcal{H}$) does not underestimate Q values at each iteration. However, the relation holds with high probability for small values of α.

The derivation for a tighter bound consists of two stages, (1) first we will show that bound is an upper bound on the CQL Q values. (2) The second step consists of obtaining Q values as a lower bound on the true Q values.

F.3 UPPER BOUND ON CQL($\mathcal{H}$)

We begin by writing the first expectation exclusively on both its terms,

$$\alpha \left(\mathbb{E}_{s \sim \mathcal{D}} \left[\log \sum_a \exp(Q) \right] - \mathbb{E}_{s \sim \mathcal{D}, a \sim \hat{\pi}_\beta} \left[Q \right] \right) + \frac{1}{2} \mathbb{E}_{s,a,s' \sim \mathcal{D}} \left[\left(Q - \hat{\mathcal{B}}^\pi \hat{Q}^k \right)^2 \right] \tag{129}$$

By applying Jensen's inequality twice to the first expectation, one obtains,

$$\leq \alpha \left(\log \sum_a \mathbb{E}_{s \sim \mathcal{D}} \left[\exp(Q) \right] - \mathbb{E}_{s \sim \mathcal{D}, a \sim \hat{\pi}_\beta} \left[Q \right] \right) + \frac{1}{2} \mathbb{E}_{s,a,s' \sim \mathcal{D}} \left[\left(Q - \hat{\mathcal{B}}^\pi \hat{Q}^k \right)^2 \right] \tag{130}$$

Utilizing Hoeffding's Lemma for exponent in the first inner expectation with the difference $(\hat{Q}^k - Q_{\min})^2$ wherein $Q_{\min}$ denotes the minimum Q value up till iteration k, we obtain the following upper bound,

$$\leq \alpha \left(\log \sum_a \exp \left(\frac{1}{8} \left(\hat{Q}^k - Q_{\min} \right)^2 \right) - \mathbb{E}_{s \sim \mathcal{D}, a \sim \hat{\pi}_\beta} [Q] \right) + \frac{1}{2} \mathbb{E}_{s,a,s' \sim \mathcal{D}} [(Q - \hat{\mathcal{B}}^\pi \hat{Q}^k)^2] \tag{131}$$

F.4 LOWER BOUND ON Q

Following the same process of optimizing over the upper bound by taking the derivative w.r.t Q and setting it to 0,

$$\alpha \left[\frac{1}{4} (\hat{Q}^k - Q_{\min}) \text{softmax} \left(\frac{1}{8} \left(\hat{Q}^k - Q_{\min} \right) \right) - \hat{\pi}_\beta \right] + (Q - \hat{\mathcal{B}}^\pi \hat{Q}^k) \hat{\pi}_\beta = 0 \tag{132}$$

$$= (Q - \hat{\mathcal{B}}^\pi \hat{Q}^k) \hat{\pi}_\beta = -\alpha \left[\frac{1}{4} \left(\hat{Q}^k - Q_{\min} \right) \text{softmax} \left(\frac{1}{8} \left(\hat{Q}^k - Q_{\min} \right) \right) - \hat{\pi}_\beta \right] \tag{133}$$

$$Q = \hat{\mathcal{B}}^\pi \hat{Q}^k - \frac{\alpha}{\hat{\pi}_\beta} \left[\frac{1}{4} (\hat{Q}^k - Q_{\min}) \text{softmax} \left(\frac{1}{8} \left(\hat{Q}^k - Q_{\min} \right) \right) - \hat{\pi}_\beta \right] \tag{134}$$

Since $\hat{Q}^k - Q_{\min} \geq 0$, $\text{softmax}(\hat{Q}^k - Q_{\min}) \geq 0$ and $\frac{1}{4} (\hat{Q}^k - Q_{\min}) \text{softmax} \left(\frac{1}{8} (\hat{Q}^k - Q_{\min}) \right) \geq \hat{\pi}_\beta$ with high probability, Q values at each iteration k are underestimated resulting in $Q \leq \hat{\mathcal{B}}^\pi \hat{Q}^k$.

F.5 AN ALTERNATE PERSPECTIVE ON APPROXIMATIONS

This section provides a statistical perspective towards approximations. Starting with the CQL($\mathcal{H}$) objective and applying Jensen's inequality twice to the first expectation as in the previous section,

$$\leq \alpha \left(\log \sum_a \mathbb{E}_{s \sim \mathcal{D}} \left[\exp(Q) \right] - \mathbb{E}_{s \sim \mathcal{D}, a \sim \hat{\pi}_\beta} \left[Q \right] \right) + \frac{1}{2} \mathbb{E}_{s,a,s' \sim \mathcal{D}} \left[\left(Q - \hat{\mathcal{B}}^\pi \hat{Q}^k \right)^2 \right] \tag{135}$$

One can formulate Eq. 135 as a statistical relation which incorporates the probability of overestimation $\Pr(\exp(Q) \geq \exp(\hat{\mathcal{B}}^\pi \hat{Q}^k))$. Upon utilizing the Chernoff bound $\Pr(\exp(Q) \geq \exp(\hat{\mathcal{B}}^\pi \hat{Q}^k)) \leq \frac{\mathbb{E}[e^Q]}{\exp(\hat{\mathcal{B}}^\pi \hat{Q}^k))}$ one obtain the following,

$$\leq \alpha \Bigg(\log \sum_a \exp(\hat{\mathcal{B}}^\pi \hat{Q}^k) \Pr(\exp(Q) \geq \exp(\hat{\mathcal{B}}^\pi \hat{Q}^k)) - \mathbb{E}_{s \sim \mathcal{D}, a \sim \hat{\pi}_\beta} \left[Q \right] \Bigg) + \frac{1}{2} \mathbb{E}_{s,a,s' \sim \mathcal{D}} \left[\left(Q - \hat{\mathcal{B}}^\pi \hat{Q}^k \right)^2 \right] \tag{136}$$

We now make the sub-Gaussianity assumption which bounds the probability of underestimation with regards to the previous Bellman iterates $\Pr(\exp(Q) \geq \exp(\hat{\mathcal{B}}^\pi \hat{Q}^k)) \leq 2 \exp(-c(\hat{\mathcal{B}}^\pi \hat{Q}^k)^2)$ with $c > 0$ as a constant. The assumption, although a strong one in various respects, allows one

to reformulate the objective with external tunable parameters such as c. Utilizing the assumption in Eq. 136 provides an alternate upper bound on CQL($\mathcal{H}$).

$$\leq \alpha \Bigg(\log \sum_a 2 \exp(\hat{\mathcal{B}}^\pi \hat{Q}^k (1 - c\hat{\mathcal{B}}^\pi \hat{Q}^k)) \Pr(\exp(Q) \geq \exp(\hat{\mathcal{B}}^\pi \hat{Q}^k)) \\ - \mathbb{E}_{s \sim \mathcal{D}, a \sim \hat{\pi}_\beta} [Q] \Bigg) + \frac{1}{2} \mathbb{E}_{s,a,s' \sim \mathcal{D}} \left[\left(Q - \hat{\mathcal{B}}^\pi \hat{Q}^k \right)^2 \right] \quad (137)$$

Taking the derivative of Eq. 137 w.r.t. Q values and setting it to 0,

$$\alpha \left[\nabla_Q \log \left(\Pr(\exp(Q) \geq \exp(\hat{\mathcal{B}}^\pi \hat{Q}^k)) \right) - \hat{\pi}_\beta \right] + \hat{\pi}_\beta (Q - \hat{\mathcal{B}}^\pi \hat{Q}^k) = 0 \quad (138)$$

$$Q = \hat{\mathcal{B}}^\pi \hat{Q}^k - \alpha \left[\frac{\nabla_Q \Pr(\exp(Q) \geq \exp(\hat{\mathcal{B}}^\pi \hat{Q}^k))}{\hat{\pi}_\beta \Pr(\exp(Q) \geq \exp(\hat{\mathcal{B}}^\pi \hat{Q}^k))} - 1 \right] \quad (139)$$

Eq. 139 depicts the probabilistic nature of underestimations dependent on the probability of overestimations, a result that may be understood for future work.

G Conservatism as Dual Problems

The CQL framework, by virtue of explicit underestimations, gives rise to a theoretically rich set of objectives in the dual space. One can interpret CQL as a constrained optimization problem and gain intuition towards its individual components. More concretely, we investigate dual problems of CQL($\mathcal{H}$) in detail and study the variation of its individual components by establishing a link between the conjugate formulations.

G.1 Primer on Duality

For completeness, we revisit the Lagrange dual function. Consider a constrained optimization problem as in Eq. 140.

$$\min f(\mathrm{x}) \quad \text{subject to } h_i(\mathrm{x}) \leq 0\,,\ i = 1, ..m. \quad (140)$$

The Lagrangian $L(\{\mathrm{x}, \lambda\})$ of the above problem can be expressed as per Eq. 141 with $\operatorname{dom} L(\{\mathrm{x}, \lambda\}) = \operatorname{dom} f \times \mathbb{R}^m$.

$$L(\{\mathrm{x}, \lambda\}) = f(\mathrm{x}) + \sum_{i=1}^{m} \lambda_i h_i(x) \quad (141)$$

Here, λ is denoted as the Lagrange multiplier. One can define the Lagrange dual function (Boyd & Vandenberghe, 2004) $g : \mathbb{R}^m \to \mathbb{R}$ as the minimum value of the Lagrangian over $\mathrm{x}\, \forall \lambda \in \mathbb{R}^m$.

$$g(\{\lambda\}) = \min_{\mathrm{x} \in \operatorname{dom} f} L(\{\mathrm{x}, \lambda\}) = \min_{\mathrm{x} \in \operatorname{dom} f} \left(f(\mathrm{x}) + \sum_{i=1}^{m} \lambda_i h_i(\mathrm{x}) \right) \quad (142)$$

The dual function, being a pointwise minimum of a family of affine functions of λ, is concave. We define the dual problem as finding the best lower bound from the Lagrange dual function. This is mathematically expressed in Eq. 143.

$$\max_{\lambda} g(\{\lambda\}) \quad \text{subject to } \lambda \geq 0 \quad (143)$$

G.2 Relationship between Conjugate and Dual Problems

Following the precept from Boyd & Vandenberghe (2004), we establish a direct relationship between the conjugate and dual of the optimization problem,

$$L(\{x, \lambda\}) = \min_{\mathrm{x} \in \operatorname{dom} f} \left(f(\mathrm{x}) + \sum_{i=1}^{m} \lambda_i h_i(\mathrm{x}) \right) \quad (144)$$

Considering $h_i(\mathrm{x})$ of the form $A_i \mathrm{x}_i \leq b_i$, we have the following dual in compact notation,

$$g(\{\lambda\}) = \min_{\mathrm{x} \in \mathrm{dom}\, f} \left(f(\mathrm{x}) + \lambda (A^{\mathrm{T}} \mathrm{x} - b) \right) \tag{145}$$

$$= -b\lambda + \min_{\mathrm{x}} \left(f(\mathrm{x}) + A^{\mathrm{T}} \lambda \mathrm{x} \right) \tag{146}$$

resulting in the following relationship between conjugate and dual expressions,

$$g(\{\lambda\}) = -b\lambda - f^*(-A^{\mathrm{T}} \lambda) \tag{147}$$

G.3 Dual of Soft-Maximum Transform

Minimization of the soft-maximum $\mathrm{lse}(Q)$ can be realized as a sub-problem to the CQL($\mathcal{H}$) with the constraint enforcing Q values to be underestimated at each subsequent iteration, $Q \leq \hat{\mathcal{B}}^\pi \hat{Q}^k$. Mathematically,

$$\min_Q \log \sum_a \exp(Q(s,a)) \quad \text{subject to } Q \leq \hat{\mathcal{B}}^\pi \hat{Q}^k \tag{148}$$

$$L(\{Q, \lambda\}) = \log \sum_a \exp(Q(s,a)) + \lambda \left(Q - \hat{\mathcal{B}}^\pi \hat{Q}^k \right) \tag{149}$$

which yields the following dual,

$$g(\{\lambda\}) = \min_Q \left(\log \sum_a \exp(Q(s,a)) + \lambda \left(Q - \hat{\mathcal{B}}^\pi \hat{Q}^k \right) \right) \tag{150}$$

$$g(\{\lambda\}) = -\lambda \hat{\mathcal{B}}^\pi \hat{Q}^k + \mathcal{H}(-\lambda) \tag{151}$$

Thus, we have,

$$\max_\lambda -\lambda \hat{B}^\pi \hat{Q}^k + \mathcal{H}(-\lambda) \quad \text{subject to } \lambda \geq 0 \tag{152}$$

A meticulous inspection of Eq. 152 reveals that the dual problem is an enropy maximization problem in the inverse variable space of $-\lambda$ logits.

G.4 Dual of CQL($\mathcal{H}$)

Analogous to previous section, one can consider the dual of complete CQL($\mathcal{H}$) objective with explicit underestimation constraints.

Formulating the primal problem, we get,

$$\min_Q \alpha\, \mathbb{E}_{s \sim \mathcal{D}} \left[\mathrm{lse}(Q) - \mathbb{E}_{a \sim \hat{\pi}_\beta}[Q(s,a)] \right] + \frac{1}{2} \mathbb{E}_{s,a,s' \sim \mathcal{D}} \left[\left(Q(s,a) - \hat{\mathcal{B}}^\pi \hat{Q}^k \right)^2 \right] \quad \text{s.t. } Q \leq \hat{\mathcal{B}}^\pi \hat{Q}^k \tag{153}$$

$$g(\{\lambda\}) = \min_Q \alpha\, \mathbb{E}_{s \sim \mathcal{D}} \left[\mathrm{lse}(Q) - \mathbb{E}_{a \sim \hat{\pi}_\beta}[Q(s,a)] \right] + \frac{1}{2} \mathbb{E}_{s,a,s' \sim \mathcal{D}} \left[\left(Q(s,a) - \hat{\mathcal{B}}^\pi \hat{Q}^k \right)^2 \right] + \lambda \left(Q - \hat{\mathcal{B}}^\pi \hat{Q}^k \right) \tag{154}$$

$$= -\lambda \hat{\mathcal{B}}^\pi \hat{Q}^k + \min_Q \left(\mathrm{CQL}(\mathcal{H}) + \lambda Q \right) \tag{155}$$

yielding the following dual problem,

$$\max_\lambda -\lambda \hat{\mathcal{B}}^\pi \hat{Q}^k - \mathrm{CQL}_{\mathrm{conj}}(-\lambda) \quad \text{subject to } \lambda \geq 0 \tag{156}$$

Observing Eq. 156, one notices that the dual problem maximizes the conjugate CQL objective with inverse dual variable $-\lambda$.

H ADDITIONAL RESULTS

H.1 VALUE FUNCTION OPTIMIZATION

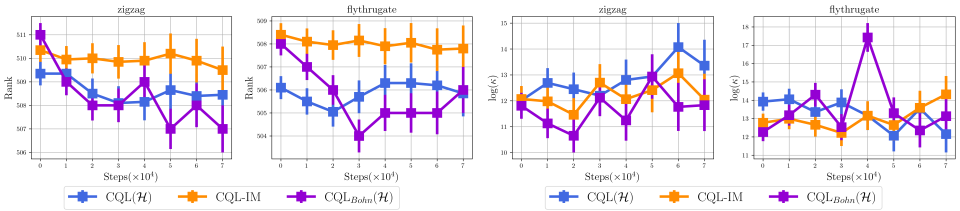

Figure 9: **(left-center left)** Variation of rank, **(center right-right)** Variation of $\log(\kappa)$ during training

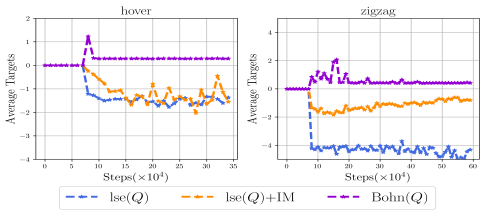

Figure 10: Average targets of CQL variants.

The setting further reasons about approximations by comparing the various sources of estimation errors. Fig. 10 links the variation in average target values to their respective objectives. CQL($\mathcal{H}$) exhibits conservatism by virtue of minimizing $\mathrm{lse}(Q)$ values. Underestimations accumulating over the course of learning steer estimates away from the true Q function. $\mathrm{Bohn}(Q)$, being an upper bound on $\mathrm{lse}(Q)$, provably overestimates Q values. This in turn, enforces $\mathrm{CQL}_{\mathrm{Bohn}}(\mathcal{H})$ to yield over-optimistic estimates which hurt convergence. Lastly, $\mathrm{lse}(Q)$ with the additional IM objective ($\mathrm{lse}(Q)$+IM) approximates analogously to $\mathrm{lse}(Q)$ but highlights potential towards convergence.

H.2 ACTOR VISUALIZATIONS

We further establish a link between conservative approximations and their effects on learning of offline policies. Fig. 11, 12 and 13 present the 2D t-SNE (van der Maaten & Hinton, 2008) embeddings of representations learned by the last, second and first layers of policies respectively. While overestimations incurred by the upper bounding $\mathrm{CQL}_{\mathrm{Bohn}}(\mathcal{H})$ are vividly reflected in the offline policy distribution, it is only seldomly clear to distinguish between conservative and near-accurate approximations learned by CQL($\mathcal{H}$) and CQL-IM respectively. The above indicates that although the value function of CQL-IM is less conservative than that of CQL($\mathcal{H}$), it is still able to learn conservative parameters with significant accuracy.

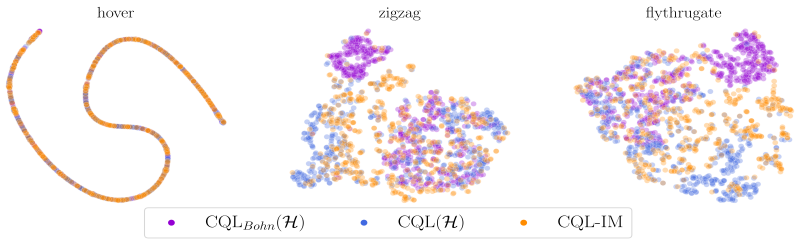

Figure 11: 2D t-SNE embeddings of representations learned by the last layer of policies on *hover*, *zigzag* and *flythrugate* tasks.

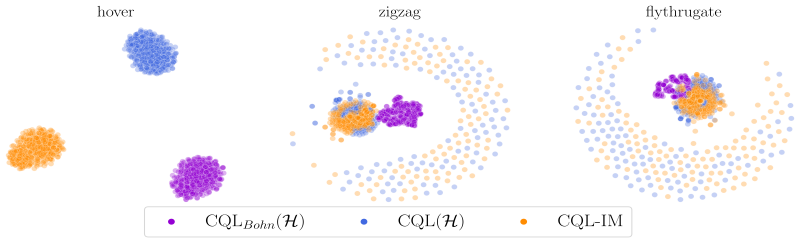

Figure 12: 2D t-SNE embeddings of representations learned by the second layer of policies on *hover*, *zigzag* and *flythrugate* tasks.

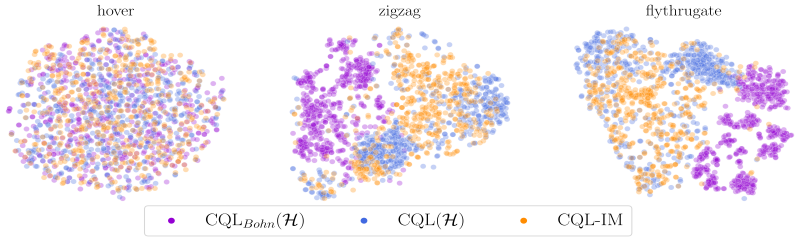

Figure 13: 2D t-SNE embeddings of representations learned by the first layer of policies on *hover*, *zigzag* and *flythrugate* tasks.

H.3 Experiments in D4RL Benchmark

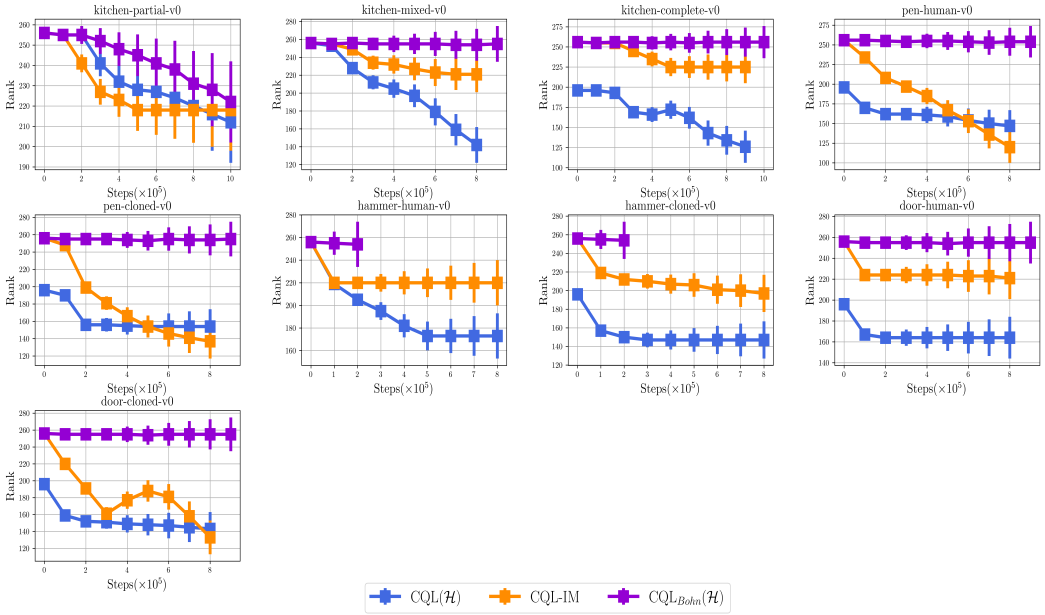

Figure 14: Variation of rank on 9 challenging D4RL datasets

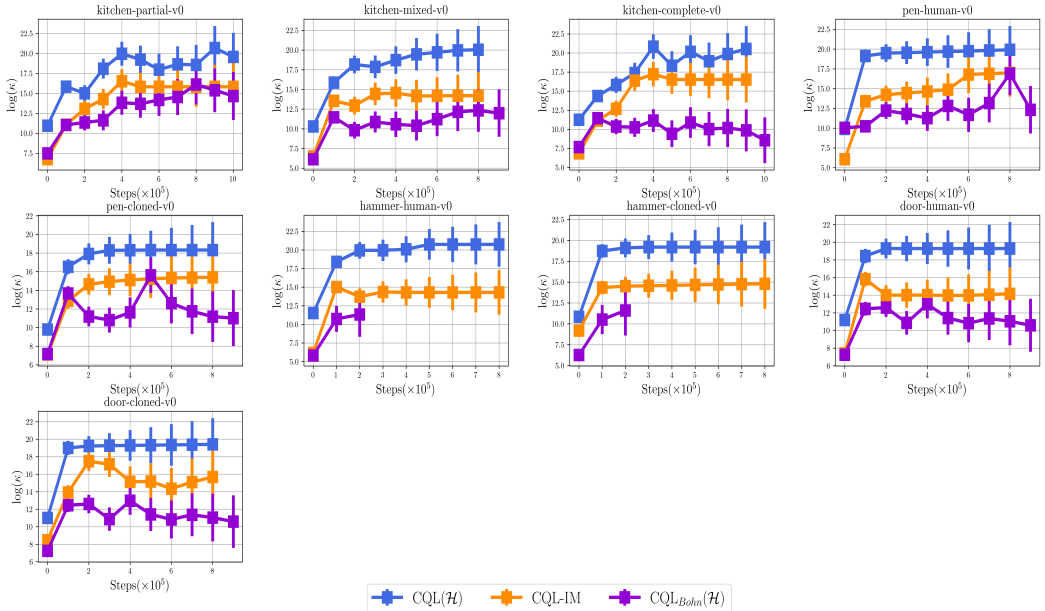

Figure 15: Variation of $\log(\kappa)$ on 9 challenging D4RL datasets

This section extends our analysis to the D4RL benchmark. We consider a set of challenging datasets and domains from the benchmark in order to highlight the improvement in optimization objective when using CQL-IM. We consider a set of diverse datasets (*expert,medium,cloned*) including transitions from the significantly challenging *human* demonstrations. In addition to these, we consider domains with high-dimensional complex control (*Franka* and *adroit*) and sparse reward settings (*antmaze navigation*).

Fig. 14 presents the variation in rank on 9 of the most challenging tasks for CQL($\mathcal{H}$) and its variants. CQL($\mathcal{H}$) presents a faster collapse in rank of value function while $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ and CQL-IM are found robust to these changes. We also note that while $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ is found more robust than CQL-IM, it is due to its inability to learn in the presence of the assumption imposed on $\tilde{\mathbb{H}}$. Value function estimates of $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ either explode or collapse, hindering learning. In certain scenarios (such as the *hamnmer* domain), $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ collapses midway due to the $\tilde{\mathbb{H}}$ assumption.

Fig. 15 presents the variation in κ (on log scale) for the considered 9 tasks. As before, CQL($\mathcal{H}$) is found to maximize condition number values greater than $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ and CQL-IM. It is worth noting the high errors in $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ for different runs. These indicate that although $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ optimizes a well-conditioned objective, it still presents instabilities arising from the assumption on $\tilde{\mathbb{H}}$. CQL-IM, on the other hand, presents consistent values of $\log(\kappa)$ with fewer errors across random runs.

Dataset	CQL($\mathcal{H}$)	$\text{CQL}_{\text{Bohn}}(\mathcal{H})$	CQL-IM
hopper-expert-v0	-242.81	407.26	-377.36
hopper-medium-v0	-274.01	134.81	-267.06
antmaze-umaze-v0	-23.48	40.61	-23.48
antmaze-umaze-diverse-v0	-37.63	88.97	-27.61
kitchen-complete-v0	4.6×10^{11}	10.53×10^{4}	2.03×10^{11}
kitchen-partial-v0	4.27×10^{11}	8.65×10^{6}	3.71×10^{11}
kitchen-mixed-v0	5.31×10^{11}	3.33×10^{4}	3.47×10^{11}

Table 3: Comparison of the gap $\hat{Q}^k - Q$ between value estimates $\hat{Q}^k$ and actual Monte-Carlo returns Q. Note that negative values denote conservatism and positive values overestimations.

Next, we compare conservatism between CQL($\mathcal{H}$) and its variants. We utilize the metric $\hat{Q}^k - Q$ following Kumar et al. (2020b) which presents the gap between estimated Q values $\hat{Q}^k$ and true Q values Q. Estimated Q values are obtained by the critic function in the actor-critic framework of CQL. True Q values Q, on the other hand, are obtained as Monte-Carlo returns by explicitly rolling out the average discounted return. A negative value of $\hat{Q}^k - Q$ denotes that the critic underestimates true Q function while a positive value denotes overestimations. Optimal estimates are obtained in the case of $\hat{Q}^k = Q$.

Table 3 presents the comparison of $\hat{Q}^k - Q$ on *hopper*, *antmaze* and *kitchen* tasks. When compared to CQL($\mathcal{H}$), $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ presents positive gaps demonstrating overestimations in the Q function. CQL-IM, on the other hand, presents low negative values closer to 0 on *hopper* and *antmaze* tasks. The result verifies our theoretical insight of CQL-IM being a better approximation to the true Q function. We further note the large gaps on *kitchen* tasks which arise as a consequence of high-dimensional control and significantly challenging datasets. To note the severity of these gaps, we emphasize that CQL($\mathcal{H}$) itself finds it challenging to lower bound $\hat{Q}^k$. This results in large values which are further observed in CQL-IM. However, it is also worth noting that these values for CQL-IM are lower when compared to CQL($\mathcal{H}$). This indicates that the variational regularization aids CQL-IM to bring values closer to the true Q function.

Lastly, we provide comprehensive comparison of average returns on the D4RL benchmark by expanding our analysis to more recent baselines. We include policy constrained as well as value constrained methods. Additionally, we include state-of-the-art algorithms which have demonstrated success on learning locomotion, navigation and dexterous control from offline transistions. Namely,

we consider Bootstrapping Error Accumulation Reduction (BEAR) (Kumar et al., 2019), Behavior Regularized Actor-Critic (BRAC) (both versions) (Wu et al., 2019), Advantage Weighted Regression (AWR) (Peng et al., 2019), AlgaeDICE (aDICE) (Nachum et al., 2019), Batch Constrained Q-learning (BCQ) (Fujimoto et al., 2019) and Random Ensemble Mixture (REM) (Agarwal et al., 2020). Additionally, we include the recent TD3+BC (Fujimoto & Gu, 2021) baseline which, similar to our method, regularizes the value function with naive BC for improved runtime and simplicity. We also add Uncertainty Weighted Actor-Critic (UWAC) (Wu et al., 2021) which demonstrates significant improvements in learning from human demonstrations. Lastly, we include Implicit Q-Learning (IQL) (Kostrikov et al., 2021) which highlights a new state-of-the-art in offline RL.

Table 4 presents the comparison between all methods on 13 tasks from the D4RL benchmark. TD3+BC presents improvements on locomotion tasks. This arises as a result of the simplicity of the algorithm where TD learning is combined with BC. UWAC presents improved returns on complex human demonstration datasets with multi-modalities in data transitions. IQL, being the recent state-of-the-art, improves over prior baselines on locomotion, navigation and dexterous control tasks. CQL($\mathcal{H}$) is found competitive to or improves over baseline methods. CQL-IM presents similar or improved performance when compared to CQL($\mathcal{H}$). In the *antmaze diverse* and *kitchen partial* settings, CQL-IM presents best returns by outperforming CQL($\mathcal{H}$) with a slight margin. This improvement in performance is a direct consequence of variational regularization introduced by the combination of BC objective. In settings where data transitions are optimal (such as *expert*), CQL-IM learns quickly by leveraging good coverage of the dataset. *In settings where dataset is sub-optimal, the variational regularization in CQL-IM only improves the optimization process. This way, CQL-IM enjoys the best of both worlds by interpolating between offline RL and Imitation Learning.*

Before concluding, we also note the decrease in $\text{CQL}_{\text{Bohn}}(\mathcal{H})$ returns which arises as a direct consequence of the assumption on $\tilde{\mathbb{H}}$ interfering with optimization of value estimates.

I Implementation Details

I.1 LineWorld Details

The setup consists of two settings, namely *online* and *offline* agents. Policies corresponding to both agents are approximated using neural networks with 1 hidden layer of 32 units with relu nonlinearity. Both policies were trained with a learning rate of 0.001, batch size as 32 and $\gamma = 0.95$ as these were the optimal parameters determined after a conventional grid search selection. Agent policies collect data transitions every 20 steps in a rolling replay buffer with a capacity of 1000 transitions. In case of the second experiment, the above parameters are kept same corresponding to both agents. The state space of the environment is changed to the number of states in the set $\{10, 15, 100, 200, 500, 1000\}$.

I.2 Environment Details

At each timestep, the agent observes a 20 dimensional vector as its state consisting of the drone's position, translational and angular velocities, orientation and quaternion representations in 3D cartesian system. Following are the task setup corresponding to each environment for our experiments-

Takeoff: The agent is tasked to lift the drone along Z-axis of world frame. Reward awarded to the agent is inversely proportional to the distance of drone's center from goal position $(0, 0, 1)$.

Hover: The agent is tasked to lift the drone along Z-axis of world frame and maintain its position above the origin. Reward awarded to the agent is proportional to the norm distance of drone's center of mass from the goal position.

Flythrugate: The agent is tasked to fly through a gate opening placed at a wide angle on the drone's X-axis. Note that this is a significantly challenging scenario in comparison to other tasks as it requires the agent to solve two problems. Firstly, the policy must learn to identify the location of the gate. And secondly, the policy must learn to control the drone (by taking off and moving forward). Reward awarded to the agent is proportional to the norm distance of drone's center of mass from the gate opening and inversely proportional to the time spent during simulation.

ZigZag: The agent is tasked to move in a zig-zag flight pattern during simulation. The agent must first travel leftwards and then move rightwards towards its final goal location. Reward awarded to the

Dataset	SAC	BC	BEAR	BRAC-p	BRAC-v	AWR	aDICE	BCQ	REM	TD3+BC	UWAC	IQL	CQL($\mathcal{H}$)	CQL$_{\text{Bohn}}$($\mathcal{H}$)	CQL-IM
hopper-expert-v0	0.7	109.0	110.3	6,6	3.7	-	-	-	-	112.2	**135.0**	-	109.9	96.38	117.21
hopper-medium-v0	0.8	29.0	47.6	31.2	32.3	35.9	1.2	54.5	0.6	**99.5**	88.9	-	58.0	51.78	64.22
antmaze-umaze-v0	0.0	65.0	73.0	50.0	70.0	-	-	-	-	78.6	-	**87.5**	74.0	79.13	82.44
antmaze-umaze-diverse-v0	0.0	55.0	61.0	40.0	70.0	-	-	-	-	71.4	-	62.2	84.0	62.97	**85.32**
antmaze-medium-play-v0	0.0	0.0	0.0	0.0	0.0	-	-	-	-	10.6	-	**71.2**	61.2	53.45	61.76
antmaze-medium-diverse-v0	0.0	0.0	8.0	0.0	0.0	-	-	-	-	3.0	-	**70.0**	53.7	34.55	52.19
kitchen-complete-v0	15.0	33.8	0.0	0.0	0.0	-	-	-	-	-	-	**62.5**	43.8	48.22	54.25
kitchen-partial-v0	0.0	33.8	13.1	0.0	0.0	-	-	-	-	-	-	46.3	49.8	32.76	**52.43**
kitchen-mixed-v0	51.5	0.0	0.0	0.0	0.0	-	-	-	-	-	-	51.0	**66.53**	47.68	54.14
pen-human-v0	6.3	34.4	-1.0	8.1	0.6	12.3	-3.3	68.9	3.5	-	65.0	**71.0**	46.56	41.97	54.59
pen-cloned-v0	23.5	56.9	26.5	1.6	-2.5	28.0	-2.9	44.0	-3.4	-	45.1	37.3	**39.2**	23.5	38.4
hammer-human-v0	0.5	1.5	0.3	0.3	0.2	1.2	0.3	0.5	0.2	-	**8.3**	1.4	6.51	4.46	6.51
hammer-cloned-v0	0.2	0.8	0.3	0.3	0.3	0.4	0.3	0.4	0.2	-	1.2	**2.1**	1.61	0.4	1.46
door-human-v0	3.9	0.5	-0.3	-0.3	-0.3	0.4	0.0	0.0	-0.1	-	**10.7**	4.3	9.9	5.2	10.1
door-cloned-v0	0.0	-0.1	-0.1	-0.1	-0.1	0.0	0.0	0.0	-0.1	-	1.2	**1.6**	0.4	0.0	0.4

Table 4: Average returns on all 13 D4RL datasets (results averaged over 4 random seeds). Values in **bold** denote highest returns. We additionally compare with state-of-the-art baselines BEAR (Kumar et al., 2019), BRAC (Wu et al., 2019), AWR (Peng et al., 2019), AlgaeDICE (aDICE) (Nachum et al., 2019), BCQ (Fujimoto et al., 2019), REM (Agarwal et al., 2020), TD3+BC (Fujimoto & Gu, 2021), UWAC (Wu et al., 2021) and IQL (Kostrikov et al., 2021).

agent is proportional to the norm distance of drone's center of mass from the intermediate location for first half of simulation, and the distance from goal location for second half of simulation.

I.3 HYPERPARAMETERS

All policies are trained for 1×10^5 timesteps and evaluated over 5 episodes every 1000 timesteps. Experiments are carried out over 10 random seeds. All policies consists of two hidden layers of 1024 units each with ReLU activations and an output layer of 1024 units with tanh nonlinearity. Following are training details corresponding to each algorithm-

BC: The expert, being a trained SAC policy, optimally executes actions in the environment for 5 episodes, resulting in state transitions. These transitions are stored in the dataset every 10000 steps. BC learners were trained with a fixed batch size of 1024 and learning rate of 0.001 as these present the best results for all tasks.

SAC: We utilize the diagonal Gaussian policy with standard deviation bounds rescaled in the range [2,-10] and entropy temperature tuned manually to 0.001. We additionally tried lower values of temperature but it led to deteriorating performance during training. Learning rate for all tasks was tuned to 0.00003 with higher values presenting significant variance across random seeds.

CQL & Variants: Our CQL implementation is based on the original implementation of CQL($\mathcal{H}$) (Kumar et al., 2020b) and makes use of the same architecture setup. The policy is trained with an increased batch size of 1024 and reduced learning rate of 0.0001 for our experiments as this presents marginal improvement. Our experiments additionally tried tuning the temperature parameters but policies were found robust to these components. In the interest of a fair comparison with CQL, we keep the parameter values of CQL-IM and CQL$_{\text{Bohn}}(\mathcal{H})$ unchanged. Temperature parameter for intrinsic motivation in CQL-IM was tuned with the optimal value being equivalent as the entropy temperature for CQL.

I.4 D4RL DETAILS

Experiments in D4RL were conducted using the author-provided implementation of CQL($\mathcal{H}$). We implemented CQL-IM similar to our Aerial Control tasks. BC, in its log form, was found to equally work well. Following are the taskwise details of the implementation-

Franka: The only change we make is of temperature parameter for the BC term. We considered values of $0.1, 0.01$ & 0.001. The value of 0.01 was found to work best with lower values resembling behaviors similar to CQL.

Adroit: Temperature parameter was tuned for 0.1 & 0.01 with 0.01 found to work well. Additionally, we tuned the learning rate and found smaller values $1e^{-4}$ to work well across random runs.

Locomotion & Antmaze Navigation: We use author-provided default parameters with a temperature value of 0.001 tuned across 3 random seeds.

I.5 ADDITIONAL DESIGN DETAILS

This section further elucidates the implementational aspects of our algorithms. Specifically, we enlist the algorithms utilized in the study along with their advantages and limitations.

Computation of IM: We utilize the variational regularization as intrinsic motivation to the Q value function. This is implemented using an off-policy variant in our Aerial control experiments and as likelihood maximization in the D4RL setting. Algorithms 1 and 2 present their implementation with the term `temp` as the temperature parameter to balance Q and IM terms. We omit the maximum entropy regularization step as this is a fairly standard procedure implemented in the CQL framework.

Computation of $\text{Bohn}(Q)$: We utilize efficient Jacobian Vector Product (JVP) and Hessian Vector Product (HVP) to compute the Taylors formulation of $\text{Bohn}(Q)$. We restate the quadratic formulation with $b = -(\hat{\mathbb{H}}\psi - g(\psi))$ and $c = \frac{1}{2}\psi^{\mathrm{T}}\hat{\mathbb{H}}\psi - g(\psi)^{\mathrm{T}}\psi + \text{lse}(\psi)$,

$$\text{Bohn}(Q) = \frac{1}{2}Q^{\mathrm{T}}\hat{\mathbb{H}}Q - bQ + c \tag{157}$$

Here, we initialize ψ as a random parameter vector (alternatively a vector of 1s). At each step, we exactly compute the above expression using efficient JVP (`jvp`) and HVP (`hvp`) functions.

Algorithm 3 presents the computation of $\mathrm{Bohn}(Q)$. The resulting expression is used as is in place of $\mathrm{lse}(Q)$ in CQL.

In addition to $\mathrm{Bohn}(Q)$ and MI approximations, we considered other second order methods such as conventional root finding and quasi-Newton methods. Although suitable for large-scale optimization (Boyd & Vandenberghe, 2004; Nocedal & Wright, 2006), these methods did not provide sufficient improvements in the RL setting. Most algorithms present large computational and memory requirements which are difficult to fulfill in the offline RL setting. We provide a summary of these limitations along with their time and memory complexities (with n as dimension of approximation matrix) in Table 5.

Algorithm 1 CQL-IM (Off-Policy Variant PyTorch-like)

```
Initialize temp=0.01, pi, critic, q, buffer, reg=1, T, gamma, CQL;
while not converged do
    batch = buffer.sample()
    value = critic.forward(batch)
    for t in range(1,T) do
        regt = q.forward(batch)
        regt **= (gamma**t-1)
        reg *= regt
    end for
    reg = torch.log(reg)
    value += temp*reg.mean()
    CQL.train(batch, value)
end while
return pi
```

Algorithm 2 CQL-IM (PyTorch-like)

```
Initialize temp=0.01, pi, critic, q, buffer, reg=1, T, gamma, CQL;
while not converged do
    batch = buffer.sample()
    value = critic.forward(batch)
    reg = q.forward(batch)
    reg = torch.log(reg)
    value += temp*reg.mean()
    CQL.train(batch, value)
end while
return pi
```

Algorithm 3 Computation of $\mathrm{Bohn}(Q)$

```
Initialize jvp, hvp, Ĥ, ψ;
procedure BOHN(ψ,Q)
    lse = torch.logsumexp(ψ)
    g = lse.grad()
    b = -(hvp(Ĥ, ψ) - g)
    first = 0.5*hvp(hvp(Ĥ, ψ),ψ)
    c = first - jvp(g,ψ) + lse
    bohn = 0.5*hvp(hvp(Ĥ, Q),Q) - jvp(b,Q) + c
    return bohn
end procedure
```

Approximation Method	Limitation	Computational Cost	Memory Cost
Newton's Method	Requires exact computation of the inverse of full Hessian	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$
Gauss-Newton Hessian	Requires solving the full high-dimensional Newton system with line search	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$
Proximal Dogleg Method	Requires defining two hand-engineered thresholds	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$
BFGS Method	Requires an inverse of the Hessian approximation with line search	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$
Broyden Class	Requires an SVD with line search	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$
Limited Memory Quasi-Newton	Requires two-loop recursions to track last iterates	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$

Table 5: Additional approximation methods considered with their practical limitations.

The advantages of Bohning approximation when compared to methods in Table 5 are as follows-

- $\mathrm{Bohn}(Q)$ does not require inverse of the approximation
- $\mathrm{Bohn}(Q)$ does not require line search
- $\mathrm{Bohn}(Q)$ does not require hand-engineered thresholds
- $\mathrm{Bohn}(Q)$ does not require recursions or SVD
- The method is simple to implement (5 lines of code)
- Approximation is easier to optimize (in libraries such as JAX)
- The method is easier to scale to higher dimensions by using a limited memory variant

The computational complexity of $\mathrm{Bohn}(Q)$ is same as $\mathcal{O}(n^2)$ and its memory complexity matches the $\mathcal{O}(n)$ rate of Limited Memory Quasi-Newton method.