Conservative Offline Distributional Reinforcement Learning

Yecheng Jason Ma, Dinesh Jayaraman, Osbert Bastani
University of Pennsylvania
{jasonyma, dineshj, obastani}@seas.upenn.edu

Abstract

Many reinforcement learning (RL) problems in practice are *offline*, learning purely from observational data. A key challenge is how to ensure the learned policy is safe, which requires quantifying the risk associated with different actions. In the online setting, distributional RL algorithms do so by learning the distribution over returns (i.e., cumulative rewards) instead of the expected return; beyond quantifying risk, they have also been shown to learn better representations for planning. We propose Conservative Offline Distributional Actor Critic (CODAC), an offline RL algorithm suitable for both risk-neutral and risk-averse domains. CODAC adapts distributional RL to the offline setting by penalizing the predicted quantiles of the return for out-of-distribution actions. We prove that CODAC learns a conservative return distribution—in particular, for finite MDPs, CODAC converges to an uniform lower bound on the quantiles of the return distribution; our proof relies on a novel analysis of the distributional Bellman operator. In our experiments, on two challenging robot navigation tasks, CODAC successfully learns risk-averse policies using offline data collected purely from risk-neutral agents. Furthermore, CODAC is state-of-the-art on the D4RL MuJoCo benchmark in terms of both expected and risk-sensitive performance. Code is available at: https://github.com/JasonMa2016/CODAC

1 Introduction

In many applications of reinforcement learning, actively gathering data through interactions with the environment can be risky and unsafe. Offline (or batch) reinforcement learning (RL) avoids this problem by learning a policy solely from historical data (called *observational data*) [9, 22, 23].

A shortcoming of most existing approaches to offline RL [11, 46, 20, 21, 48, 18] is that they are designed to maximize the expected value of the cumulative reward (which we call the *return*) of the policy. As a consequence, they are unable to quantify risk and ensure that the learned policy acts in a safe way. In the online setting, there has been recent work on *distributional* RL algorithms [7, 6, 27, 38, 17], which instead learn the full distribution over future returns. They can use this distribution to plan in a way that avoids taking risky, unsafe actions. Furthermore, when coupled with deep neural network function approximation, they can learn better state representations due to the richer distributional learning signal [4, 26], enabling them to outperform traditional RL algorithms even on the risk-neutral, expected return objective [4, 7, 6, 47, 14].

We propose Conservative Offline Distributional Actor-Critic (CODAC), which adapts distributional RL to the offline setting. A key challenge in offline RL is accounting for high uncertainty on out-of-distribution (OOD) state-action pairs for which observational data is limited [23, 20]; the value estimates for these state-action pairs are intrinsically high variance, and may be exploited by the policy without correction due to the lack of online data gathering and feedback. We build on conservative Q-learning [21], which penalizes Q values for OOD state-action pairs to ensure that

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

the learned Q-function lower bounds the true Q-function. Analogously, CODAC uses a penalty to ensure that the quantiles of the learned return distribution lower bound those of the true return distribution. Crucially, the lower bound is data-driven and selectively penalizes the quantile estimates of state-actions that are less frequent in the offline dataset; see Figure 1.

We prove that for finite MDPs, CODAC converges to an estimate of the return distribution whose quantiles uniformly lower bound the quantiles of the true return distribution; in addition, this data-driven lower bound is tight up to the approximation error in estimating the quantiles using finite data. Thus, CODAC obtains a uniform lower bound on *all* integrations of the quantiles, including the standard RL objective of expected return, the risk-sensitive conditional-value-at-risk (CVaR) objective [35], as well as many other risk-sensitive objectives. We additionally prove that CODAC expands the gap in quantile estimates between in-distribution and OOD actions, thus avoiding overconfidence when extrapolating to OOD actions [11].

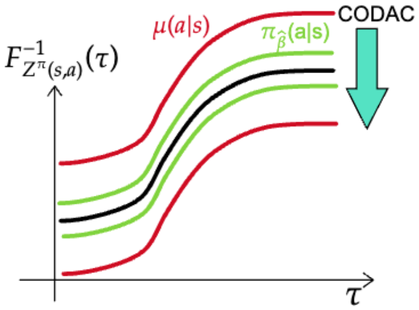

Figure 1: CODAC obtains conservative estimates of the true return quantiles (black); it penalizes out-of-distribution actions, $\mu(a \mid s)$, more heavily than in-distribution actions, $\pi_{\hat{\beta}}(a \mid s)$.

Our theoretical guarantees rely on novel techniques for analyzing the distributional Bellman operator, which is challenging since it acts on the infinite-dimensional function space of return distributions (whereas the traditional Bellman operator acts on a finite-dimensional vector space). We provide several novel results that may be of independent interest; for instance, our techniques can be used to bound the error of the fixed-point of the empirical distributional Bellman operator; see Appendix A.6.

Finally, to obtain a practical algorithm, CODAC builds on existing distributional RL algorithms by integrating conservative return distribution estimation into a quantile-based actor-critic framework.

In our experiments, we demonstrate the effectiveness of CODAC on both risk-sensitive and risk-neutral RL. First, on two novel risk-sensitive robot navigation tasks, we show that CODAC successfully learns risk-averse policies using offline datasets collected purely from a risk-neutral agent, a challenging task that all our baselines fail to solve. Next, on the D4RL Mujoco suite [10], a popular offline RL benchmark, we show that CODAC achieves state-of-art results on both the original risk-neutral version as well a modified risk-sensitive version [43]. Finally, we empirically show that CODAC computes quantile lower-bounds and gap-expanded quantiles even on high-dimensional continuous-control problems, validating our key theoretical insights into the effectiveness of CODAC.

Related work. There has been growing interest in offline (or batch) RL [22, 23]. The key challenge in offline RL is to avoid overestimating the value of out-of-distribution (OOD) actions rarely taken in the observational dataset [40, 44, 20]. The problem is that policy learning optimizes against the value estimates; thus, even if the estimation error is i.i.d., policy optimization biases towards taking actions with high variance value estimates, since some of these values will be large by random chance. In risk-sensitive or safety-critical settings, these actions are exactly the ones that should be avoided.

One solution is to constrain the learned policy to take actions similar to the ones in the dataset (similar to imitation learning)—e.g., by performing support matching [46] or distributional matching [20, 12]. However, these approaches tend to perform poorly when data is gathered from suboptimal policies. An alternative is to regularize the Q-function estimates to be conservative at OOD actions [21, 48, 18]. CODAC builds on these approaches, but obtains conservative estimates of all quantile values of the return distribution rather than just the expected return. Traditionally, the literature on off-policy evaluation (OPE) [32, 16, 39, 25, 37] aims to estimate the expected return of a policy using pre-collected offline data; CODAC proposes an OPE procedure amenable to all objectives that can be expressed as integrals of the return quantiles. Consequently, our fine-grained approach not only enables risk-sensitive policy learning, but also improves performance on risk-neutral domains.

In particular, CODAC builds on recent works on distributional RL [4, 7, 6, 47], which parameterize and estimate the entire return distribution instead of just a point estimate of the expected return (i.e., the Q-function) [29, 28]. Distributional RL algorithms have been shown to achieve state-of-art performance on Atari and continuous control domains [14, 3]; intuitively, they provide richer training signals that stabilize value network training [4]. Existing distributional RL algorithms parameterize

~~the policy return distribution in many different ways, including canonical return atoms [4], the expectiles [36], the moments [31], and the quantiles [7, 6, 47]. CODAC builds on the quantile approach due to its suitability for risk sensitive policy optimization. The quantile representation provides an unified framework for optimizing different objectives of interest [7], such as the risk-neutral expected return, and a family of risk sensitive objectives representable by the quantiles; this family includes, for example, the conditional value at risk (CVaR) return [35, 5, 38, 17], the Wang measure [45], and the cumulative probability weighting (CPW) metric [42]. Recent work has provided theoretical guarantees on learning CVaR policies [17]; however, their approach cannot provide bounds on the quantiles of the estimated return distribution, which is significantly more challenging since there is no closed-form expression for the Bellman update on the return quantiles.~~

~~Finally, there has been some recent work adapting distributional RL to the offline setting [1, 43]. First, REM [1] builds on QR-DQN [7], an online distributional RL algorithm; however, REM can be applied to regular DQN [29] and does not directly utilize the distributional aspect of QR-DQN. The closest work to ours is ORAAC [43], which uses distributional RL to learn a CVaR policy in the offline setting. ORAAC uses imitation learning to avoid OOD actions and stay close to the data distribution. As discussed above, imitation learning strategies can perform poorly unless the dataset comes from an optimal policy; in our experiments, we find that CODAC significantly outperforms ORAAC. Furthermore, unlike ORAAC, we provide theoretical guarantees on our approach.~~

2 Background

Offline RL. Consider a Markov Decision Process (MDP) [33] $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(s' \mid s, a)$ is the transition distribution, $R(r \mid s, a)$ is the reward distribution, and $\gamma \in (0,1)$ is the discount factor, and consider a stochastic policy $\pi(a \mid s) : \mathcal{S} \to \Delta(\mathcal{A})$. A *rollout* using π from state s using initial action a is the random sequence $\xi = ((s_0, a_0, r_0), (s_1, a_1, r_1), ...)$ such that $s_0 = s$, $a_0 = a$, $a_t \sim \pi(\cdot \mid s_t)$ (for $t > 0$), $r_t \sim R(\cdot \mid s_t, a_t)$, and $s_{t+1} \sim P(\cdot \mid s_t, a_t)$; we denote the distribution over rollouts by $D^\pi(\xi \mid s, a)$. The Q-function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ of π is its expected discounted cumulative return $Q^\pi(s,a) = \mathbb{E}_{D^\pi(\xi|s,a)}[\sum_{t=0}^\infty \gamma^t r_t]$. Assuming the rewards satisfy $r_t \in [R_{\text{min}}, R_{\text{max}}]$, then we have $Q^\pi(s,a) \in [V_{\text{min}}, V_{\text{max}}] \subseteq [R_{\text{min}}/(1-\gamma), R_{\text{max}}/(1-\gamma)]$.

A standard goal of reinforcement learning (RL), which we call *risk-neutral* RL, is to learn the optimal policy π^* such that $Q^{\pi^*}(s,a) \geq Q^\pi(s,a)$ for all $s \in \mathcal{S}, a \in \mathcal{A}$ and all π.

In offline RL, the learning algorithm only has access to a fixed dataset $\mathcal{D} := \{(s, a, r, s')\}$, where $r \sim R(\cdot \mid s, a)$ and $s' \sim P(\cdot \mid s, a)$. The goal is to learn the optimal policy without any interaction with the environment. Though we do not assume that $\mathcal{D}$ necessarily comes from a single behavior policy, we define the empirical behavior policy to be $\hat{\pi}_\beta(a \mid s) := \frac{\sum_{s',a' \in \mathcal{D}} \mathbb{1}(s'=s, a'=a)}{\sum_{s' \in \mathcal{D}} \mathbb{1}(s'=s)}$. With slight abuse of notation, we write $(s, a, r, s') \sim \mathcal{D}$ to denote a uniformly random sample from the dataset. Also, in this paper, we broadly refer to actions not drawn from $\hat{\pi}_\beta(\cdot \mid s)$ (i.e., low probability density) as out-of-distribution (OOD).

Fitted Q-evaluation (FQE) [9, 34] uses Q-learning for offline RL, which leverages the fact that $Q^\pi = \mathcal{T}^\pi Q^\pi$ is the unique fixed point of the Bellman operator $\mathcal{T}^\pi : \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \to \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ defined by

$$\mathcal{T}^\pi Q(s,a) = \mathbb{E}_{R(r|s,a)}[r] + \gamma \cdot \mathbb{E}_{P^\pi(s',a'|s,a)}[Q(s',a')],$$

where $P^\pi(s',a'|s,a) = P(s' \mid s,a)\pi(a' \mid s')$. In the offline setting, we do not have access to $\mathcal{T}^\pi$; instead, FQE uses an approximation $\hat{\mathcal{T}}^\pi$ obtained by replacing R and P in $\mathcal{T}^\pi$ with estimates $\hat{R}$ and $\hat{P}$ based on $\mathcal{D}$. Then, we can estimate Q^π by starting from an arbitrary $\hat{Q}^0$ and iteratively computing

$$\hat{Q}^{k+1} := \arg\min_Q \mathcal{L}(\hat{Q}, \hat{\mathcal{T}}^\pi \hat{Q}^k) \quad \text{where} \quad \mathcal{L}(Q, Q') = \mathbb{E}_{\mathcal{D}(s,a)}\left[(Q(s,a) - Q'(s,a))^2\right].$$

Assuming we search over the space of all possible Q (i.e., do not use function approximation), then the minimizer is $\hat{Q}^{k+1} = \hat{\mathcal{T}}^\pi \hat{Q}^k$, so $\hat{Q}^k = (\hat{\mathcal{T}}^\pi)^k Q^0$. If $\hat{\mathcal{T}}^\pi = \mathcal{T}^\pi$, then $\lim_{k\to\infty} \hat{Q}^k = Q^\pi$.

Distributional RL. In distributional RL, the goal is to learn the distribution of the discounted cumulative rewards (i.e., *returns*) [4]. Given a policy π, we denote its return distribution as the random variable $Z^\pi(s,a) = \sum_{t=0}^\infty \gamma^t r_t$, which is a function of a random rollout $\xi \sim D^\pi(\cdot \mid s, a)$;

note that Z^π includes three sources of randomness: (1) the reward $R(\cdot \mid s,a)$, (2) the transition $P(\cdot \mid s,a)$, and (3) the policy $\pi(\cdot \mid s)$. Also, note that $Q^\pi(s,a) = \mathbb{E}_{D^\pi(\xi|s,a)}[Z^\pi(s,a)]$. Analogous to Q-function Bellman operator, the distributional Bellman operator for π is

$$\mathcal{T}^\pi Z(s,a) \overset{D}{:=} r + \gamma Z(s',a') \quad \text{where} \quad r \sim R(\cdot \mid s,a),\ s' \sim P(\cdot \mid s,a),\ a' \sim \pi(\cdot \mid s'), \tag{1}$$

where $\overset{D}{=}$ indicates equality in distribution. As with Q^π, Z^π is the unique fixed point of $\mathcal{T}^\pi$ in Eq. 1.

Next, let $F_{Z(s,a)}(x) : [V_{\text{min}}, V_{\text{max}}] \to [0,1]$ be the cumulative density function (CDF) for return distribution $Z(s,a)$, and $F_{R(s,a)}$ be the CDF of $R(\cdot \mid s,a)$ Then, we have the following equality, which captures how the distributional Bellman operator $\mathcal{T}^\pi$ operates on the CDF $F_{Z(s,a)}$ [17]:

$$F_{\mathcal{T}^\pi Z(s,a)}(x) = \sum_{s',a'} P^\pi(s',a' \mid s,a) \int F_{Z(s',a')}\left(\frac{x-r}{\gamma}\right) dF_{R(s,a)}(r). \tag{2}$$

Let X and Y be two random variables. Then, the *quantile function* (i.e., inverse CDF) F_X^{-1} of X is $F_X^{-1}(\tau) := \inf\{x \in \mathbb{R} \mid \tau \le F_X(x)\}$, and the p-*Wasserstein distance* between X and Y is $W_p(X,Y) = (\int_0^1 |F_Y^{-1}(\tau) - F_X^{-1}(\tau)|^p d\tau)^{1/p}$. Then, the distributional Bellman operator $\mathcal{T}^\pi$ is a γ-contraction in the W_p [4]—i.e., letting $\bar{d}_p(Z_1, Z_2) := \sup_{s,a} W_p(Z_1(s,a), Z_2(s,a))$ be the largest Wasserstein distance over (s,a), and $\mathcal{Z} = \{Z : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathbb{R}) \mid \forall (s,a)\ .\ \mathbb{E}[|Z(s,a)|^p] < \infty\}$ be the space of distributions over $\mathbb{R}$ with bounded p-th moment, then

$$\bar{d}_p(\mathcal{T}^\pi Z_1, \mathcal{T}^\pi Z_2) \le \gamma \bar{d}_p(Z_1, Z_2) \qquad (\forall Z_1, Z_2 \in \mathcal{Z}). \tag{3}$$

As a result, Z^π may be obtained by iteratively applying $\mathcal{T}^\pi$ to an initial distribution Z.

As before, in the offline setting, we can approximate $\mathcal{T}^\pi$ by $\hat{\mathcal{T}}^\pi$ using $\mathcal{D}$. Then, we can compute Z^π (represented as $F_{Z(s,a)}^{-1}$; see below) by starting from an arbitrary $\hat{Z}^0$, and iteratively computing

$$\hat{Z}^{k+1} = \arg\min_Z \mathcal{L}_p(Z, \hat{\mathcal{T}}^\pi \hat{Z}^k) \quad \text{where} \quad \mathcal{L}_p(Z, Z') = \mathbb{E}_{\mathcal{D}(s,a)}\left[W_p(Z(s,a), Z'(s,a))^p\right]. \tag{4}$$

We call this procedure *fitted distributional evaluation (FDE)*.

One distributional RL algorithmic framework is quantile-based distributional RL [7, 6, 47, 27, 38, 43], where the return distribution Z is represented by its quantile function $F_{Z(s,a)}^{-1}(\tau) : [0,1] \to \mathbb{R}$. Given a distribution $g(\tau)$ over $[0,1]$, the *distorted expectation* of Z is $\Phi_g(Z(s,a)) = \int_0^1 F_{Z(s,a)}^{-1}(\tau) g(\tau) d\tau$, and the corresponding policy is $\pi_g(s) := \arg\max_a \Phi_g(Z(s,a))$ [7]. If $g = \text{Uniform}([0,1])$, then $Q^\pi(s,a) = \Phi_g(Z(s,a))$; alternatively, $g = \text{Uniform}([0,\xi])$ corresponds to the CVaR [35, 5, 6] objective, where only the bottom ξ-percentile of the return is considered. Additional risk-sensitive objectives are also compatible. For example, CPW [42] amounts to $g(\tau) = \tau^\beta / (\tau^\beta + (1-\tau)^\beta)^{\frac{1}{\beta}}$, and Wang [45] has $g(\tau) = F_{\mathcal{N}}(F_{\mathcal{N}}^{-1}(\tau) + \beta)$, where $F_{\mathcal{N}}$ is the standard Gaussian CDF.

A drawback of FDE is that it does not account for estimation error, especially for pairs (s,a) that rarely appear in the given dataset $\mathcal{D}$; thus, $\hat{Z}^k(s,a)$ may be an overestimate of $Z^k(s,a)$ [12, 20, 21], even in distributional RL (since the learned distribution does not include randomness in the dataset) [14, 3]. Importantly, since we act by optimizing with respect to $\hat{Z}^k(s,a)$, the optimization algorithm will exploit these errors, biasing towards actions with higher uncertainty, which is the opposite of what is desired. In Section 3, we propose and analyze a penalty designed to avoid this issue.

3 Conservative offline distributional policy evaluation

We describe our algorithm for computing a conservative estimate of $Z^\pi(s,a)$, and provide theoretical guarantees for finite MDPs. In particular, we modify Eq. 4 to include a penalty term:

$$\tilde{Z}^{k+1} = \arg\min_Z \alpha \cdot \mathbb{E}_{U(\tau),\mathcal{D}(s,a)}\left[c_0(s,a) \cdot F_{Z(s,a)}^{-1}(\tau)\right] + \mathcal{L}_p(Z, \hat{\mathcal{T}}^\pi \tilde{Z}^k) \tag{5}$$

for some state-action dependent scale factor c_0; here, $U = \text{Uniform}([0,1])$. This objective adapts the conservative penalty in prior work [21] to the distributional RL setting; in particular, the first term in

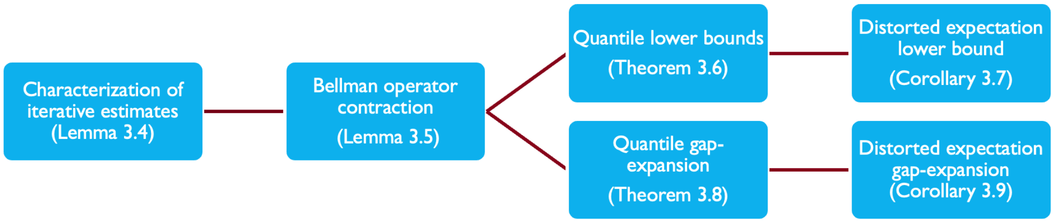
Characterization of iterative estimates (Lemma 3.4)
Bellman operator contraction (Lemma 3.5)
Quantile lower bounds (Theorem 3.6)
Distorted expectation lower bound (Corollary 3.7)
Quantile gap-expansion (Theorem 3.8)
Distorted expectation gap-expansion (Corollary 3.9)

Figure 2: Overview of our theoretical results.

the objective is a penalty that aims to shrink the quantile values for out-of-distribution (OOD) actions compared to those of in-distribution actions; intuitively, $c_0(s,a)$ should be larger for OOD actions. For now, we let c_0 be arbitrary; we describe our choice in Section 4. $\alpha \in \mathbb{R}_{>0}$ is a hyperparameter controlling the magnitude of the penalty term with respect to the usual FDE objective. We call this iterative algorithm *conservative distribution evaluation (CDE)*.

Next, we analyze theoretical properties of CDE in the setting of finite MDPs; Figure 2 overviews these results. First, we prove that CDE iteratively obtains conservative quantile estimates (Lemma 3.4) and defines a contraction operator on return distributions (Lemma 3.5). Then, our main result (Theorem 3.6) is that CDE converges to a fixed point $\tilde{Z}^\pi$ whose quantile function lower bounds that of the true returns Z^π. We also prove that CDE is *gap-expanding* (Theorem 3.8)—i.e., it is more conservative for actions that are rare in $\mathcal{D}$. These results translate to RL objectives computed by integrating the return quantiles, including expected and CVaR returns (Corollaries 3.7 & 3.9).

We begin by describing our assumptions on the MDP and dataset. First, we assume that our dataset $\mathcal{D}$ has nonzero coverage of all actions for states in the dataset [23, 21].

Assumption 3.1. For all $s \in \mathcal{D}$ and $a \in \mathcal{A}$, we have $\hat{\pi}_\beta(a \mid s) > 0$.

This assumption is only needed by our theoretical analysis to avoid division-by-zero and ensure that all estimates are well-defined; alternatively, we could assign a very low value $\hat{\pi}_\beta(a \mid s) := \epsilon$ for all actions not visited at state s in the offline dataset and renormalize $\hat{\pi}_\beta(a \mid s)$ accordingly. Next, we impose regularity conditions on the fixed point Z^π of the distributional Bellman operator $\mathcal{T}^\pi$.

Assumption 3.2. For all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, $F_{Z^\pi(s,a)}$ is smooth. Furthermore, there exists $\zeta \in \mathbb{R}_{>0}$ such that for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, $F_{Z^\pi(s,a)}$ is ζ-strongly monotone—i.e., we have $F'_{Z^\pi(s,a)}(x) \geq \zeta$.

The smoothness assumption ensures that the pth moments of $Z^\pi(s,a)$ are bounded (since $Z^\pi \in [V_{\text{min}}, V_{\text{max}}]$ is also bounded), which in turn ensures that $Z^\pi \in \mathcal{Z}$. The monotonicity assumption is needed to ensure convergence of $F^{-1}_{Z^\pi(s,a)}$. Next, we assume that the search space in (5) includes all possible functions (i.e., no function approximation).

Assumption 3.3. The search space of the minimum over Z in (5) is over all smooth functions $F_{Z(s,a)}$ (for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$) with support on $[V_{\text{min}}, V_{\text{max}}]$.

This assumption is required for us to analytically characterize the solution $\tilde{Z}^{k+1}$ of the CDE objective. Finally, we also assume $p > 1$ (i.e., we use the p-Wasserstein distance for some $p > 1$).

Now, we describe our key results. Our first result characterizes the CDE iterates $\tilde{Z}^{k+1}$; importantly, if $c_0(s,a) > 0$, then these iterates encode successively more conservative quantile estimates.

Lemma 3.4. *For all $s \in \mathcal{D}$, $a \in \mathcal{A}$, $k \in \mathbb{N}$, and $\tau \in [0,1]$, we have*

$$F^{-1}_{\tilde{Z}^{k+1}(s,a)}(\tau) = F^{-1}_{\hat{\mathcal{T}}^\pi \tilde{Z}^k(s,a)}(\tau) - c(s,a) \;\; \textit{where} \;\; c(s,a) = |\alpha p^{-1} c_0(s,a)|^{1/(p-1)} \cdot \text{sign}(c_0(s,a)).$$

We give a proof in Appendix A.1; roughly speaking, it follows by setting the gradient of (5) equal to zero, relying on results from the calculus of variations to handle the fact that $F^{-1}_{Z(s,a)}$ is a function.

Next, we define the *CDE operator* $\tilde{\mathcal{T}}^\pi = \mathcal{O}_c \hat{\mathcal{T}}^\pi$ to be the composition of $\hat{\mathcal{T}}^\pi$ with the *shift operator* $\mathcal{O}_c : \mathcal{Z} \to \mathcal{Z}$ defined by $F^{-1}_{\mathcal{O}_c Z(s,a)}(\tau) = F^{-1}_{Z(s,a)}(\tau) - c(s,a)$; thus, Lemma 3.4 says $\tilde{Z}^{k+1} = \tilde{\mathcal{T}}^\pi \tilde{Z}^k$. Now, we show that $\tilde{\mathcal{T}}^\pi$ is a contraction in the maximum Wasserstein distance $\bar{d}_p$.

Lemma 3.5. *$\tilde{\mathcal{T}}^\pi$ is a γ-contraction in $\bar{d}_p$, so $\tilde{Z}^k$ converges to a unique fixed point $\tilde{Z}^\pi$.*

The first part follows since $\hat{\mathcal{T}}^\pi$ is a γ-contraction in $\bar{d}_p$ [4, 7], and $\mathcal{O}_c$ is a non-expansion in $\bar{d}_p$, so by composition, $\tilde{\mathcal{T}}^\pi$ is a γ-contraction in $\bar{d}_p$; the second follows by the Banach fixed point theorem.

Now, our first main theorem says that the fixed point $\tilde{Z}^\pi$ of $\tilde{\mathcal{T}}^\pi$ is a conservative estimate of Z^π at all quantiles τ—i.e., CDE computes quantile estimates that lower bound the quantiles of the true return; furthermore, it says that this lower bound is tight.

Theorem 3.6. *For any $\delta \in \mathbb{R}_{>0}$, $c_0(s,a) > 0$, with probability at least $1-\delta$,*

$$F^{-1}_{Z^\pi(s,a)}(\tau) \geq F^{-1}_{\tilde{Z}^\pi(s,a)}(\tau) + (1-\gamma)^{-1} \min_{s',a'}\{c(s',a') - \Delta(s',a')\},$$
$$F^{-1}_{Z^\pi(s,a)}(\tau) \leq F^{-1}_{\tilde{Z}^\pi(s,a)}(\tau) + (1-\gamma)^{-1} \max_{s',a'}\{c(s',a') - \Delta(s',a')\}$$

for all $s \in \mathcal{D}$, $a \in \mathcal{A}$, and $\tau \in [0,1]$, where $\Delta(s,a) = \frac{1}{\zeta}\sqrt{\frac{5|\mathcal{S}|}{n(s,a)}\log\frac{4|\mathcal{S}||\mathcal{A}|}{\delta}}$. Furthermore, for α sufficiently large $(i.e., \alpha \geq \max_{s,a}\{\frac{p\cdot\Delta(s,a)^{p-1}}{c_0(s,a)}\})$, we have $F^{-1}_{Z^\pi(s,a)}(\tau) \geq F^{-1}_{\tilde{Z}^\pi(s,a)}(\tau)$.

We give a proof in Appendix A.2. The first inequality says that the quantile estimates computed by CDE form a lower bound on the true quantiles; this bound is not vacuous as long as α satisfies the given condition. Furthermore, the second inequality states that this lower bound is tight.

Many RL objectives (e.g., expected or CVaR return) are distorted expectations (i.e, integrals of the return quantiles). We can extend Theorem 3.6 to obtain conservative estimates for all such objectives:

Corollary 3.7. *For any $\delta \in \mathbb{R}_{>0}$, $c_0(s,a) > 0$, α sufficiently large, and $g(\tau)$, with probability at least $1-\delta$, for all $s \in \mathcal{D}$, $a \in \mathcal{A}$, we have $\Phi_g(Z^\pi(s,a)) \geq \Phi_g(\tilde{Z}^\pi(s,a))$.*

Choosing $g = \text{Uniform}([0,1])$ gives $Q^\pi(s,a) \geq \tilde{Q}^\pi(s,a)$—i.e., a lower bound on the Q-function. CQL [21] obtains a similar lower-bound; thus, CDE generalizes CQL to other objectives—e.g., it can be used in conjunction with any distorted expectation objective (e.g., CVaR, Wang, CPW, etc.) for risk-sensitive offline RL.

Note that Theorem 3.6 does not preclude the possibility that the lower bounds are more conservative for good actions (i.e., ones for which $\hat{\pi}_\beta(a \mid s)$ is larger). We prove that under the choice[1]

$$c_0(s,a) = \frac{\mu(a \mid s) - \hat{\pi}_\beta(a \mid s)}{\hat{\pi}_\beta(a \mid s)} \tag{6}$$

for some $\mu(a \mid s) \neq \hat{\pi}_\beta(a \mid s)$, then $\tilde{\mathcal{T}}^\pi$ is *gap-expanding*—i.e., the difference in quantile values between in-distribution and out-of-distribution actions is larger under $\tilde{\mathcal{T}}^\pi$ than under $\mathcal{T}^\pi$. Intuitively, $c_0(s,a)$ is large for actions a with higher probability under μ than under $\hat{\pi}_\beta$ (i.e., an OOD action).

Theorem 3.8. *For $p = 2$, α sufficiently large, and c_0 as in (6), for all $s \in \mathcal{S}$ and $\tau \in [0,1]$,*

$$\mathbb{E}_{\hat{\pi}_\beta(a|s)} F^{-1}_{\tilde{Z}^\pi(s,a)}(\tau) - \mathbb{E}_{\mu(a|s)} F^{-1}_{\tilde{Z}^\pi(s,a)}(\tau) \geq \mathbb{E}_{\hat{\pi}_\beta(a|s)} F^{-1}_{Z^\pi(s,a)}(\tau) - \mathbb{E}_{\mu(a|s)} F^{-1}_{Z^\pi(s,a)}(\tau).$$

As before, the gap-expansion property implies gap-expansion of integrals of the quantiles—i.e.:

Corollary 3.9. *For $p = 2$, α sufficiently large, c_0 as in (6), and any $g(\tau)$, for all $s \in \mathcal{S}$,*

$$\mathbb{E}_{\hat{\pi}_\beta(a|s)} \Phi_g(\tilde{Z}^\pi(s,a)) - \mathbb{E}_{\mu(a|s)} \Phi_g(\tilde{Z}^\pi(s,a)) \geq \mathbb{E}_{\hat{\pi}_\beta(a|s)} \Phi_g(Z^\pi(s,a)) - \mathbb{E}_{\mu(a|s)} \Phi_g(Z^\pi(s,a)).$$

Together, Corollaries 3.7 & 3.9 say that CDE provides conservative lower bounds on the return quantiles while being less conservative for in-distribution actions.

Finally, we briefly discuss the condition on α in Theorems 3.6 & 3.8. In general, α can be taken to be small as long as $\Delta(s,a)$ is small for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, which in turn holds as long as $n(s,a)$ is large—i.e., the dataset $\mathcal{D}$ has wide coverage.

[1] We may have $c_0(s,a) \leq 0$; we can use $c_0'(s,a) = c_0(s,a) + (1 - \min_{s,a} c_0(s,a))$ to avoid this issue.

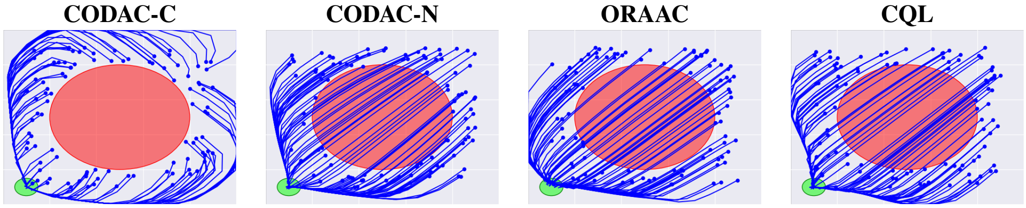

Figure 3: 2D visualization of evaluation trajectories on the Risky PointMass environment. The red region is risky, the solid blue circles indicate initial states, and the blue lines are trajectories. CODAC-C is the only algorithm that successfully avoids the risky region.

4 Conservative offline distributional actor critic

Next, we incorporate the distributional evaluation algorithm in Section 3 into an actor-critic framework. Following [21], we propose a min-max objective where the inner loop chooses the current policy to maximize the CDE objective, and the outer loop minimizes the CDE objective for this policy:

$$\hat{Z}^{k+1} = \arg\min_{Z} \max_{\mu} \left\{ \alpha \cdot \mathbb{E}_{U(\tau)} \left[\mathbb{E}_{\mathcal{D}(s),\mu(a|s)} F^{-1}_{Z(s,a)}(\tau) - \mathbb{E}_{\mathcal{D}(s,a)} F^{-1}_{Z(s,a)}(\tau) \right] + \mathcal{L}_p(Z, \hat{\mathcal{T}}^{\pi^k} \hat{Z}^k) \right\}$$

where we have used c_0 as in (6). We can interpret μ as an actor policy, the first term as the objective for μ, and the overall objective as an actor-critic algorithm [8]. In this framework, a natural choice for μ is a maximum entropy policy $\mu(a \mid s) \propto \exp(Q(s,a))$ [49]. Then, our objective becomes

$$\hat{Z}^{k+1} = \arg\min_{Z} \left\{ \alpha \cdot \mathbb{E}_{U(\tau)} \left[\mathbb{E}_{\mathcal{D}(s)} \log \sum_{a} \exp(F^{-1}_{Z(s,a)}(\tau)) - \mathbb{E}_{\mathcal{D}(s,a)} F^{-1}_{Z(s,a)}(\tau) \right] + \mathcal{L}_p(Z, \hat{\mathcal{T}}^{\pi^k} \hat{Z}^k) \right\},$$

where $U = \text{Uniform}([0,1])$; we provide a derivation in Appendix B. We call this strategy *conservative offline distributional actor critic (CODAC)*. To optimize over Z, we represent the quantile function as a DNN $G_\theta(\tau; s, a) \approx F^{-1}_{Z(s,a)}(\tau)$. The main challenge is optimizing the term $\mathcal{L}_p(Z, \hat{\mathcal{T}}^\pi \hat{Z}^k) = W_p(Z, \hat{\mathcal{T}}^\pi \hat{Z}^k)^p$. We do so using distributional temporal-differences (TD) [6]. For a sample $(s, a, r, s') \sim \mathcal{D}$ and $a' \sim \pi(\cdot \mid s')$ and random quantiles $\tau, \tau' \sim U$, we have

$$\mathcal{L}_p(Z, \hat{\mathcal{T}}^\pi \hat{Z}^k) \approx \mathcal{L}_\kappa(\delta; \tau) \qquad \text{where} \qquad \delta = r + \gamma G_{\theta'}(\tau'; s', a') - G_\theta(\tau; s, a).$$

Here, δ is the distributional TD error, θ' are the parameters of the target Q-network [30], and

$$\mathcal{L}_\kappa(\delta; \tau) = \begin{cases} |\tau - \mathbb{1}(\delta < 0)| \cdot \delta^2/(2\kappa) & \text{if } |\delta| \leq \kappa \\ |\tau - \mathbb{1}(\delta < 0)| \cdot (|\delta| - \kappa/2) & \text{otherwise} . \end{cases} \tag{7}$$

is the τ-Huber quantile regression loss at threshold κ [15]; then, $\mathbb{E}_{U(\tau)}\mathcal{L}_\kappa(\delta; \tau)$ is an unbiased estimator of the Wasserstein distance that can be optimized using stochastic gradient descent (SGD) [19]. With this strategy, our overall objective can be optimized using any off-policy actor-critic method [24, 13, 11]; we use distributional soft actor-critic (DSAC) [27], which replaces the Q-network in SAC [13] with a quantile distributional critic network [7]. We provide the full CODAC pseudocode in Algorithm 1 of Appendix B.

5 Experiments

We show that CODAC achieves state-of-the-art results on both risk-sensitive (Sections 5.1 &5.2) and risk-neutral (Section 5.3) offline RL tasks, including our risky robot navigation and D4RL[2] [10]. We also show that our lower bound (Theorem 3.6) and gap-expansion (Theorem 3.8) results approximately hold in practice, (Section 5.4), validating our theory on CODAC's effectiveness. We provide additional details (e.g., environment descriptions, hyperparameters, and additional results) in Appendix C.

5.1 Risky robot navigation

[2] The D4RL dataset license is Apache License 2.0.

Table 1: **Risky robot navigation quantitative evaluation**. CODAC-C achieves the best performance on most metrics and is the only method that learns risk-averse behavior. This table is reproduced with standard deviations in Table 6 in Appendix C.

Algorithm	**Risky PointMass**				**Risky Ant**			
	Mean	Median	$\text{CVaR}_{0.1}$	Violations	Mean	Median	$\text{CVaR}_{0.1}$	Violations
DSAC (Online)	-7.69	-3.82	-49.9	94	-866.1	-833.3	-1422.7	2247
CODAC-C (Ours)	**-6.05**	-4.89	**-14.73**	**0.0**	-456.0	-433.4	**-686.6**	**347.8**
CODAC-N (Ours)	-8.60	**-4.05**	-51.96	108.3	**-432.7**	**-395.1**	-847.1	936.0
ORAAC	-10.67	-4.55	-64.12	138.7	-788.1	-795.3	-1247.2	1196
CQL	-7.51	-4.18	-43.44	93.4	-967.8	-858.5	-1887.3	1854.3

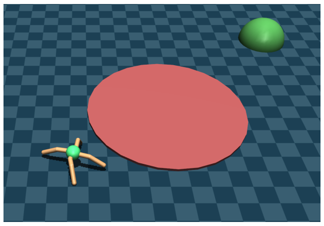

Figure 4: Risky Ant.

Tasks. We consider an Ant robot whose goal is to navigate from a random initial state to the green circle as quickly as possible (see Figure 4 for a visualization). Passing through the red circle triggers a high cost with small probability, introducing risk. A risk-neutral agent may pass through the red region, but a risk-aware agent should not. We also consider a PointMass variant for illustrative purposes.

We construct an offline dataset that is the replay buffer of a risk-neutral distributional SAC [27] agent. Intuitively, this choice matches the practical goals of offline RL, where data is gathered from a diverse range of sources with no assumptions on their quality and risk tolerance levels. [23] See Appendix C.1 for details on the environments and datasets.

Approaches. We consider two variants of CODAC: (i) CODAC-N, which maximizes the expected return and (ii) CODAC-C, which optimizes the $\text{CVaR}_{0.1}$ objective. We compare to Offline Risk-Averse Actor Critic (ORAAC) [43], a state-of-the-art offline risk-averse RL algorithm that combines a distributional critic with an imitation-learning based policy to optimize a risk-sensitive objective, and to Conservative Q-Learning (CQL) [21], a state-of-art offline RL algorithm that is non-distributional.

Results. We evaluate each approach using 100 test episodes, reporting the mean, median, and $\text{CVaR}_{0.1}$ (i.e., average over bottom 10 episodes) returns, and the total number of violations (i.e., time steps spent inside the risky region), all averaged over 5 random seeds. We also report the performance of the online DSAC agent used to collect data. See Appendix C for details. Results are in Table 1.

Performance of CODAC. CODAC-C consistently outperforms the other approaches on the $\text{CVaR}_{0.1}$ return, as well as the number of violations, demonstrating that CODAC-C is able to avoid risky actions. It is also competitive in terms of mean return due to its high $\text{CVaR}_{0.1}$ performance, but performs slightly worse on median return, since it is not designed to optimize this objective. Remarkably, on Risky PointMass, CODAC-C learns a safe policy that completely avoids the risky region (i.e., zero violations), even though such behavior is absent in the dataset. In Appendix C.1, we also show that CODAC can successfully optimize alternative risk-sensitive objectives such as Wang and CPW.

Comparison to ORAAC. While ORAAC also optimizes the CVaR objective, it uses imitation learning to regularize the learned policy to stay close to the empirical behavior policy. However, the dataset contains many sub-optimal trajectories generated early in training, and is furthermore risk-neutral. Thus, imitating the behavioral policy encourages poor performance. In practice, a key use of offline RL is to leverage large datasets available for training, and such datasets will rarely consist of data from a single, high-quality behavioral policy. Our results demonstrate that CODAC is significantly better suited to learned in these settings compared to ORAAC.

Comparison to CQL. On Risky PointMass, CQL learns a risky policy with poor tail-performance, indicated by its high median performance but low $\text{CVaR}_{0.1}$ performance. Interestingly, its mean performance is also poor; intuitively, the mean is highly sensitive to outliers that may not be present in the training dataset. On Risky Ant, possibly due to the added challenge of high-dimensionality, CQL performs poorly on all metrics, failing to reach the goal and to avoid the risky region. As expected, these results show that accounting for risk is necessary in risky environments.

Qualitative analysis. In Figure 3, we show the 100 evaluation rollouts from each policy on Risky PointMass. As can be seen, CODAC-C extrapolates a safe policy that distances itself from the risky

Table 2: **D4RL results.** CODAC achieves the best overall performance in both risk-sensitive (**Left**) and risk-neutral (**Right**) variants of the benchmark. These tables are reproduced with standard deviations in Tables 7 & 9 in Appendix C.

	Algorithm	Medium		Mixed	
		Mean	$\text{CVaR}_{0.1}$	Mean	$\text{CVaR}_{0.1}$
Cheetah	CQL	33.2	-15.0	214.1	12.0
	ORAAC	**361.4**	**91.3**	307.1	118.9
	CODAC-N	338	-41	347.7	149.2
	CODAC-C	335	-27	**396.4**	**238.5**
Hopper	CQL	877.9	693.0	189.2	-21.4
	ORAAC	1007.1	767.6	876.3	524.9
	CODAC-N	993.7	952.5	1483.9	**1457.6**
	CODAC-C	**1014.0**	**976.4**	**1551.2**	1449.6
Walker2d	CQL	1524.3	**1343.8**	74.3	-64.0
	ORAAC	1134.1	663.0	222.0	-69.6
	CODAC-N	**1537.3**	1158.8	358.7	106.4
	CODAC-C	1120.8	902.3	**450.0**	**261.4**

Dataset	BCQ	MOPO	CQL	ORAAC	CODAC
halfcheetah-random	2.2	**35.4**	35.4	13.5	34.6
hopper-random	10.6	**11.7**	10.8	9.8	11.0
walker2d-random	4.9	13.6	7.0	3.2	**18.7**
halfcheetah-medium	40.7	42.3	44.4	41.0	**46.3**
walker2d-medium	53.1	17.8	79.2	27.3	**82.0**
hopper-medium	54.5	28.0	58.0	1.48	**70.8**
halfcheetah-mixed	38.2	**53.1**	46.2	30.0	44.1
hopper-mixed	33.1	67.5	48.6	16.3	**100.2**
walker2d-mixed	15.0	**39.0**	26.7	28	33.2
halfcheetah-med-exp	64.7	63.3	62.4	24.0	**70.4**
walker2d-med-exp	57.5	44.6	98.7	28.2	**106.0**
hopper-med-exp	110.9	23.7	111.0	18.2	**112.0**

region before proceeding to the goal; in contrast, all other agents traverse the risky region. For Ant, we include plots of the trajectories of trained agents in Appendix C.1, and videos in the supplement.

5.2 Risk-sensitive D4RL

Tasks. Next, we consider stochastic D4RL [43]. The original D4RL benchmark [10] consists of datasets collected by SAC agents of varying performance (Mixed, Medium, and Expert) on the Hopper, Walker2d, and HalfCheetah MuJoCo environments [41]; stochastic D4RL relabels the rewards to represent stochastic robot damage for behaviors such as unnatural gaits or high velocities; see Appendix C.2. The Expert dataset consists of rollouts from a fixed SAC agent trained to convergence; the Medium dataset is constructed the same way except the agent is trained to only achieve 50% of the expert agent's return. The Mixed dataset is the replay buffer of the Medium agent.

Results. In Table 2 (Left), we report the mean and $\text{CVaR}_{0.1}$ returns on test episodes from each approach, averaged over 5 random seeds. We show results on the Expert dataset in Appendix C.2; CODAC still achieves the strongest performance. As can be seen, CODAC-C and CODAC-N outperform both CQL and ORAAC on most datasets. Surprisingly, CODAC-N is quite effective on the $\text{CVaR}_{0.1}$ metric despite its risk-neutral objective; a likely explanation is that for these datasets, mean and CVaR performance are highly correlated. Furthermore, we observe that directly optimizing CVaR may lead to unstable training, potentially since CVaR estimates have higher variance. This instability occurs for both CODAC-C and ORAAC—on Walker2d-Medium, they perform worse than the risk-neutral algorithms. Overall, CODAC-C outperforms CODAC-N in terms of $\text{CVaR}_{0.1}$ on about half of the datasets, and often improves mean performance as well. Next, while ORAAC is generally effective on Medium datasets, it performs poorly on Mixed datasets; these results mirror the ones in Section 5.1. Finally, CQL's performance varies drastically across datasets; we hypothesize that learning the full distribution helps stabilize training in CODAC. In Appendix C.2, we also qualitatively analyze the behavior learned by CODAC compared to the baselines, demonstrating that the better CVaR performance CODAC obtains indeed translates to safer locomotion behaviors.

5.3 Risk-neutral D4RL

Task. Next, we show that CODAC is effective even when the goal is to optimize the standard expected return. To this end, we evaluate CODAC-N on the popular D4RL Mujoco benchmark [10].

Baselines. We compare to state-of-art algorithms benchmarked in [10] and [48], including Batch-Constrained Q-Learning (BCQ), Model-Based Offline Policy Optimization (MOPO) [48], and CQL. We also include ORAAC as an offline distributional RL baseline. We have omitted less competitive baselines included in [10] from the main text; a full comparison is included in Appendix C.3.

Results. Results for non-distributional approaches are directly taken from [10]; for ORAAC and CODAC, we evaluate them using 10 test episodes in the environment, averaged over 5 random seeds. As shown in Table 2 (Right), CODAC achieves strong performance across all 12 datasets, obtaining state-of-art results on 5 datasets (walker2d-random, hopper-medium, hopper-mixed, halfcheetah-medium-expert, and walker2d-medium-expert), demonstrating that performance improvements from distributional learning also apply in the offline setting. Note that CODAC's advantage is not solely due to distributional RL—ORAAC also uses distributional RL, but in most cases underperforms prior state-of-the-art, These results suggest that CODAC's use of a conservative penalty is critical for it to achieve strong performance.

5.4 Analysis of Theoretical Insights

We perform additional experiments to validate that our theoretical insights in Section 3 hold in practice, suggesting that they help explain CODAC's empirical performance.

Table 3: **Monte-Carlo estimate vs. critic prediction.** The CODAC-predicted expected and $\text{CVaR}_{0.1}$ return is a lower bound on a MC estimate of the true value.

Regular	**Walker2d-Medium** MC Return	 Q-Estimate	**Walker2d-Mixed** MC Return	 Q-Estimate	**Walker2d-Medium-Expert** MC Return	 Q-Estimate
CODAC	240.2	55.7	127.1	97.6	370.	39.7
CQL	247.2	53.0	124.5	-45.2	369.7	116.4
ORAAC	245.2	302.2	118.2	7.70×10^5	68.2	322.2

Stochastic	**Walker2d-Medium** MC $\text{CVaR}_{0.1}$	 Z-Estimate	**Walker2d-Mixed** MC $\text{CVaR}_{0.1}$	 Z-Estimate	**Walker2d-Medium-Expert** MC $\text{CVaR}_{0.1}$	 Z-Estimate
CODAC	185.7	204.2	85.6	59.9	265.3	-127.8
ORAAC	201.9	367.6	50.9	1.54×10^6	199.5	343.5

Lower bound. We show that in practice, CODAC obtains conservative estimates of the Q and CVaR objectives across different dataset types (i.e., Medium vs. Mixed vs. Medium-Expert). Given an initial state s_0, we obtain a Monte Carlo (MC) estimate of Q and CVaR for $(s_0, \pi(s_0))$ based on sampled rollouts from s_0, and compare them to the values predicted by the critic. In Table 3, we show results averaged over 10 random s_0 and with 100 MC samples for each s_0. CODAC obtains conservative estimates for both Q and CVaR; in contrast, ORAAC overestimates these values, especially on Mixed datasets, and CQL only obtains conservative estimates for Q, not CVaR.

Table 4: **Gap-expansion:** CODAC expands the quantile gap and obtains higher returns than an ablation without the conservative penalty.

	HalfCheetah-Medium-Expert Positive Gap %	 Return	**Hopper-Medium-Expert** Positive Gap %	 Return	**Walker2d-Medium-Expert** Positive Gap %	 Return
CODAC	**95.3**	**93.6**	**91.3**	**111.9**	**91.1**	**111.3**
CODAC w.o. Penalty	4.7	12.1	8.7	25.8	8.9	5.9

Gap-Expansion. Next, we verify that CODAC's quantile estimates expand their gap between in-distribution and out-of-distribution actions. We use the D4RL Medium-Expert datasets where CODAC uniformly performs well, making them ideal for understanding the source of CODAC's empirical performance. We train "CODAC w.o. Penalty", a non-conservative variant of CODAC (i.e., $\alpha = 0$), and use its actor as μ and its critic as $F^{-1}_{Z^\pi}$. Next, for each dataset, we randomly sample 1000 state-action pairs and 32 quantiles τ, resulting in 32000 (s, a, τ) tuples; for each one, we compute the quantile gaps for CODAC and CODAC w.o. Penalty. In Table 4, we show the percentage of tuples where each CODAC variant has a larger quantile gap, along with their average return. As can be seen, CODAC has a larger gap for more than 90% of the tuples on all datasets, as well as significantly higher returns. These results show that gap-expansion holds in practice and suggest that it helps CODAC achieve good performance.

6 Conclusion

We have introduced Conservative Offline Distributional Actor-Critic (CODAC), a general purpose offline distributional reinforcement learning algorithm. We have proven that CODAC obtains conservative estimates of the return quantile, which translate into lower bounds on Q and CVaR values. In our experiments, CODAC outperforms prior approaches on both stochastic, risk-sensitive offline RL benchmarks, as well as traditional, risk-neutral benchmarks.

One limitation of our work is that CODAC has hyperparameters that must be tuned (in particular, the penalty magnitude α). As in prior work, we choose these hyperparameters by evaluate online rollouts in the environment. Designing better hyperparameter selection strategies for offline RL is an important direction for future work. Finally, we do not foresee any societal impacts or ethical concerns for our work, other than the usual risks around algorithms for improving robotics capabilities.

Acknowledgments and Disclosure of Funding

This work is funded in part by an Amazon Research Award, gift funding from NEC Laboratories America, NSF Award CCF-1910769, and ARO Award W911NF-20-1-0080. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pages 104–114. PMLR, 2020.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Gabriel Barth-Maron, Matthew W Hoffman, David Budden, Will Dabney, Dan Horgan, Dhruva Tb, Alistair Muldal, Nicolas Heess, and Timothy Lillicrap. Distributed distributional deterministic policy gradients. *arXiv preprint arXiv:1804.08617*, 2018.

[4] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pages 449–458. PMLR, 2017.

[5] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research*, 18(1):6070–6120, 2017.

[6] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*, pages 1096–1105. PMLR, 2018.

[7] Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[8] Thomas Degris, Martha White, and Richard S Sutton. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*, 2012.

[9] Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

[10] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[11] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1587–1596. PMLR, 2018.

[12] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062. PMLR, 2019.

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

[14] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[15] Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964.

[16] Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning, 2016.

[17] Ramtin Keramati, Christoph Dann, Alex Tamkin, and Emma Brunskill. Being optimistic to be conservative: Quickly learning a cvar policy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4436–4443, 2020.

[18] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *arXiv preprint arXiv:2005.05951*, 2020.

[19] Roger Koenker and Kevin F Hallock. Quantile regression. *Journal of economic perspectives*, z15(4):143–156, 2001.

[20] Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *arXiv preprint arXiv:1906.00949*, 2019.

[21] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191. Curran Associates, Inc., 2020.

[22] Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning*, pages 45–73. Springer, 2012.

[23] 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.

[24] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[25] Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the curse of horizon: Infinite-horizon off-policy estimation. *arXiv preprint arXiv:1810.12429*, 2018.

[26] Clare Lyle, Marc G Bellemare, and Pablo Samuel Castro. A comparative analysis of expected and distributional reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4504–4511, 2019.

[27] Xiaoteng Ma, Qiyuan Zhang, Li Xia, Zhengyuan Zhou, Jun Yang, and Qianchuan Zhao. Distributional soft actor critic for risk sensitive learning. *arXiv preprint arXiv:2004.14547*, 2020.

[28] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[29] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[30] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[31] Thanh Tang Nguyen, Sunil Gupta, and Svetha Venkatesh. Distributional reinforcement learning via moment matching, 2020.

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

[33] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[34] Martin Riedmiller. Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*, pages 317–328. Springer, 2005.

[35] R Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of banking & finance*, 26(7):1443–1471, 2002.

[36] Mark Rowland, Robert Dadashi, Saurabh Kumar, Rémi Munos, Marc G Bellemare, and Will Dabney. Statistics and samples in distributional reinforcement learning. In *International Conference on Machine Learning*, pages 5528–5536. PMLR, 2019.

[37] Simon P. Shen, Yecheng Jason Ma, Omer Gottesman, and Finale Doshi-Velez. State relevance for off-policy evaluation, 2021.

[38] Yichuan Charlie Tang, Jian Zhang, and Ruslan Salakhutdinov. Worst cases policy gradients. *arXiv preprint arXiv:1911.03618*, 2019.

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

[40] Sebastian Thrun and Anton Schwartz. Issues in using function approximation for reinforcement learning. In *Proceedings of the Fourth Connectionist Models Summer School*, pages 255–263. Hillsdale, NJ, 1993.

[41] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[42] Amos Tversky and Daniel Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5(4):297–323, 1992.

[43] Núria Armengol Urpí, Sebastian Curi, and Andreas Krause. Risk-averse offline reinforcement learning. *arXiv preprint arXiv:2102.05371*, 2021.

[44] 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.

[45] Shaun S Wang. A class of distortion operators for pricing financial and insurance risks. *Journal of risk and insurance*, pages 15–36, 2000.

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

[47] Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tieyan Liu. Fully parameterized quantile function for distributional reinforcement learning. *arXiv preprint arXiv:1911.02140*, 2019.

[48] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14129–14142. Curran Associates, Inc., 2020.

[49] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

A Proofs

A.1 Proof of Lemma 3.4

Recall that the p-Wasserstein distance is the L_p metric between quantile functions (see Eq. 3). Thus, we can re-write the CODAC objective as

$$\begin{aligned}
&\alpha \cdot \mathbb{E}_{U(\tau),\mathcal{D}(s,a)}\left[c_0(s,a)\cdot F^{-1}_{Z(s,a)}(\tau)\right] + \mathbb{E}_{\mathcal{D}(s,a)}\int_0^1 \left|F^{-1}_{Z(s,a)}(\tau) - F^{-1}_{\hat{\mathcal{T}}^\pi \hat{Z}^k(s,a)}(\tau)\right|^p d\tau \\
&= \int_0^1 \mathbb{E}_{\mathcal{D}(s,a)}\left[\alpha\cdot c_0(s,a)\cdot F^{-1}_{Z(s,a)}(\tau) + \left|F^{-1}_{Z(s,a)}(\tau) - F^{-1}_{\hat{\mathcal{T}}^\pi \hat{Z}^k(s,a)}(\tau)\right|^p\right] d\tau.
\end{aligned}$$

We consider a perturbation

$$G^{\epsilon}_{s,a}(\tau) = F^{-1}_{Z(s,a)}(\tau) + \epsilon\cdot\phi_{s,a}(\tau)$$

for arbitrary smooth functions $\phi_{s,a}$ with compact support $[V_{\min}, V_{\max}]$, yielding

$$\int_0^1 \mathbb{E}_{\mathcal{D}(s,a)}\left[\alpha c_0(s,a)\cdot G^{\epsilon}_{s,a}(\tau) + \left|G^{\epsilon}_{s,a}(\tau) - F^{-1}_{\hat{\mathcal{T}}^\pi \hat{Z}^k(s,a)}(\tau)\right|^p\right] d\tau.$$

Taking the derivative with respect to ϵ at $\epsilon = 0$, we have

$$\begin{aligned}
&\frac{d}{d\epsilon}\int_0^1 \mathbb{E}_{\mathcal{D}(s,a)}\left[\alpha c_0(s,a)\cdot G^{\epsilon}_{s,a}(\tau) + \left|G^{\epsilon}_{s,a}(\tau) - F^{-1}_{\hat{\mathcal{T}}^\pi \hat{Z}^k(s,a)}(\tau)\right|^p\right] d\tau\Bigg|_{\epsilon=0} \\
&= \mathbb{E}_{\mathcal{D}(s,a)}\int_0^1 \left[\alpha c_0(s,a) + p\left|F^{-1}_{Z(s,a)}(\tau) - F^{-1}_{\hat{\mathcal{T}}^\pi \hat{Z}^k(s,a)}(\tau)\right|^{p-1}\operatorname{sign}\left(F^{-1}_{Z(s,a)}(\tau) - F^{-1}_{\hat{\mathcal{T}}^\pi \hat{Z}^k(s,a)}(\tau)\right)\right]\phi_{s,a}(\tau)d\tau.
\end{aligned}$$

This term must equal 0 for $F^{-1}_{Z(s,a)}$ to minimize the objective; otherwise, some perturbation $G^{\epsilon}_{s,a}$ decreases the objective value. Since $\phi_{s,a}$ are arbitrary, it must equal zero for each s, a individually; otherwise, increasing $\phi_{s,a}$ would increase the term, making it nonzero. Thus, we have

$$\int_0^1 \left[\alpha c_0(s,a) + p\left|F^{-1}_{Z(s,a)}(\tau) - F^{-1}_{\hat{\mathcal{T}}^\pi \hat{Z}^k(s,a)}(\tau)\right|^{p-1}\operatorname{sign}\Big(F^{-1}_{Z(s,a)}(\tau) - F^{-1}_{\hat{\mathcal{T}}^\pi \hat{Z}^k(s,a)}(\tau)\Big)\right]\phi_{s,a}(\tau)d\tau = 0$$

for all s, a. Then, by the fundamental lemma of the calculus of variations, for each s, a, if this term is zero for all $\phi_{s,a}$, then the integrand must be zero—i.e.,

$$\alpha c_0(s,a) + p\left|F^{-1}_{Z(s,a)}(\tau) - F^{-1}_{\hat{\mathcal{T}}^\pi \hat{Z}^k(s,a)}(\tau)\right|^{p-1}\operatorname{sign}\left(F^{-1}_{Z(s,a)}(\tau) - F^{-1}_{\hat{\mathcal{T}}^\pi \hat{Z}^k(s,a)}(\tau)\right) = 0,$$

which holds if and only if

$$F^{-1}_{Z(s,a)}(\tau) = F^{-1}_{\hat{\mathcal{T}}^\pi \hat{Z}^k(s,a)}(\tau) - c(s,a).$$

where $c(s,a) = |\alpha p^{-1} c_0(s,a)|^{1/(p-1)}\cdot\operatorname{sign}(c_0(s,a))$, Clearly, this choice of Z is valid, so the claim follows. □

A.2 Proof of Theorem 3.6

First, we have the following result, which is a concentration bound on the quantile values; this result enables us to bound the estimation error of $\hat{\mathcal{T}}^\pi$ compared to $\mathcal{T}^\pi$:

Lemma A.1. *Let $n(s,a) = |\{(s,a) \mid (s,a,r,s') \in \mathcal{D}\}|$ be the number of times (s,a) occurs in $\mathcal{D}$. For any return distribution Z with ζ-strongly monotone CDF $F_{Z(s,a)}$ and any $\delta \in \mathbb{R}_{>0}$, with probability at least $1-\delta$, for all $s \in \mathcal{D}$ and $a \in \mathcal{A}$, we have*

$$\|F^{-1}_{\hat{\mathcal{T}}^\pi Z(s,a)} - F^{-1}_{\mathcal{T}^\pi Z(s,a)}\|_\infty \le \Delta(s,a) \quad \textit{where} \quad \Delta(s,a) = \frac{1}{\zeta}\sqrt{\frac{5|\mathcal{S}|}{n(s,a)}\log\frac{4|\mathcal{S}||\mathcal{A}|}{\delta}}.$$

This lemma follows by first using the Dvoretzky-Kiefer-Wolfowitz inequality to bound the error of the empirical CDF $F_{\hat{\mathcal{T}}^\pi Z(s,a)}$ compared to the true CDF $F_{\mathcal{T}^\pi Z(s,a)}$ using similar analysis as in [17], and then leveraging monotonicity to bound the quantile functions; we give a proof in Appendix A.4. Next, we have the following key lemma, which relates one-step distributional Bellman contraction to an ∞-norm bound at the fixed point.

Lemma A.2. *If Z satisfies $\|F^{-1}_{Z(s,a)} - F^{-1}_{\mathcal{T}Z(s,a)}\|_\infty \leq \beta$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, then*

$$\|F^{-1}_{Z(s,a)} - F^{-1}_{Z^\pi(s,a)}\|_\infty \leq (1-\gamma)^{-1}\beta \qquad (\forall s \in \mathcal{S}, a \in \mathcal{A}),$$

We give a proof in Appendix A.5. As we discuss in Appendix A.6, we can use this result to obtain bounds on the fixed point of the non-conservative empirical Bellman operator $\hat{\mathcal{T}}$. Now, we prove Theorem 3.6. First, with probability at least $1-\delta$, we have

$$\begin{aligned} F^{-1}_{\tilde{\mathcal{T}}^\pi Z^\pi(s,a)}(\tau) &= F^{-1}_{\hat{\mathcal{T}}^\pi Z^\pi(s,a)}(\tau) - c(s,a) \\ &\leq F^{-1}_{\mathcal{T}^\pi Z^\pi(s,a)}(\tau) - c(s,a) + \Delta(s,a) \\ &= F^{-1}_{Z^\pi(s,a)}(\tau) - c(s,a) + \Delta(s,a), \end{aligned} \tag{8}$$

where the first step follows by Lemma 3.4 (noting that it holds for arbitrary $\tilde{Z}^k$, and substituting $\tilde{Z}^k = Z^\pi$), the second step holds with probability at least $1-\delta$ by Lemma A.1 with $Z = Z^\pi$ (since Z^π is ζ-strongly monotone), and the third step follows since $Z^\pi = \mathcal{T}^\pi Z^\pi$ is the fixed point of $\mathcal{T}^\pi$.

Now, rearranging (8), we have

$$\begin{aligned} F^{-1}_{Z^\pi(s,a)}(\tau) &\geq F^{-1}_{\tilde{\mathcal{T}}^\pi Z^\pi(s,a)}(\tau) + c(s,a) - \Delta(s,a) \\ &\geq F^{-1}_{\tilde{\mathcal{T}}^\pi Z^\pi(s,a)}(\tau) + \min_{s,a}\{c(s,a) - \Delta(s,a)\} \\ &\geq F^{-1}_{\tilde{Z}^\pi(s,a)}(\tau) + (1-\gamma)^{-1} \min_{s,a}\{c(s,a) - \Delta(s,a)\}, \end{aligned} \tag{9}$$

where in the last step, we have applied Lemma A.6 for the case $\geq$ and $\tilde{\mathcal{T}}^\pi$, and with $\beta = \min_{s,a}\{c(s,a) - \Delta(s,a)\}$. Finally, note that for the last term in (9) to be positive, we need

$$\alpha p^{-1} c_0(s,a) \geq \Delta(s,a)^{p-1} \qquad (\forall s, a).$$

Since we have assumed that $c_0(s,a) > 0$, this expression is in turn equivalent to

$$\alpha \geq \max_{s,a}\left\{\frac{p \cdot \Delta(s,a)^{p-1}}{c_0(s,a)}\right\},$$

so the claim holds. $\square$

A.3 Proof of Theorem 3.8

Lemma A.3. *For any Z and any $\bar{\Delta}$, for sufficiently large α, with probability at least $1-\delta$, we have*
$\mathbb{E}_{\hat{\pi}_\beta(a|s)} F^{-1}_{\tilde{\mathcal{T}}^\pi Z(s,a)}(\tau) - \mathbb{E}_{\mu(a|s)} F^{-1}_{\tilde{\mathcal{T}}^\pi Z(s,a)}(\tau) \geq \mathbb{E}_{\hat{\pi}_\beta(a|s)} F^{-1}_{\mathcal{T}^\pi Z(s,a)}(\tau) - \mathbb{E}_{\mu(a|s)} F^{-1}_{\mathcal{T}^\pi Z(s,a)}(\tau) + \bar{\Delta}.$

Proof. First, by Lemma 3.4, we have

$$F^{-1}_{\tilde{\mathcal{T}}^\pi Z(s,a)}(\tau) = F^{-1}_{\hat{\mathcal{T}}^\pi Z(s,a)}(\tau) - c(s,a).$$

Then, by Lemma A.1, with probability at least $1-\delta$, we have

$$F^{-1}_{\mathcal{T}^\pi Z(s,a)}(\tau) - c(s,a) - \Delta(s,a) \leq F^{-1}_{\tilde{\mathcal{T}}^\pi Z(s,a)}(\tau) \leq F^{-1}_{\mathcal{T}^\pi Z(s,a)}(\tau) - c(s,a) + \Delta(s,a).$$

Taking the expectation over $\hat{\pi}_\beta$ (resp., μ) of the lower (resp., upper) bound gives

$$\begin{aligned} \mathbb{E}_{\hat{\pi}_\beta(a|s)} F^{-1}_{\tilde{\mathcal{T}}^\pi Z(s,a)}(\tau) &\geq \mathbb{E}_{\hat{\pi}_\beta(a|s)} F^{-1}_{\mathcal{T}^\pi Z(s,a)}(\tau) - \mathbb{E}_{\hat{\pi}_\beta(a|s)} c(s,a) - \mathbb{E}_{\hat{\pi}_\beta(a|s)} \Delta(s,a) \\ \mathbb{E}_{\mu(a|s)} F^{-1}_{\tilde{\mathcal{T}}^\pi Z(s,a)}(\tau) &\leq \mathbb{E}_{\mu(a|s)} F^{-1}_{\mathcal{T}^\pi Z(s,a)}(\tau) - \mathbb{E}_{\mu(a|s)} c(s,a) + \mathbb{E}_{\mu(a|s)} \Delta(s,a), \end{aligned}$$

respectively. Recall that $p = 2$. Then, subtracting the latter from the former and rearranging terms,

$$\begin{aligned}\mathbb{E}_{\hat{\pi}_\beta(a|s)} F^{-1}_{\tilde{\mathcal{T}}^\pi Z(s,a)}(\tau) - \mathbb{E}_{\mu(a|s)} F^{-1}_{\tilde{\mathcal{T}}^\pi Z(s,a)}(\tau) \geq \mathbb{E}_{\hat{\pi}_\beta(a|s)} F^{-1}_{\mathcal{T}^\pi Z(s,a)}(\tau) - \mathbb{E}_{\mu(a|s)} F^{-1}_{\mathcal{T}^\pi Z(s,a)}(\tau) \\ + (\alpha/2)\bar{c}(s) - \bar{\Delta}(s),\end{aligned}$$

where

$$\begin{aligned}\bar{c}(s) &= \mathbb{E}_{\mu(a|s)} c_0(s,a) - \mathbb{E}_{\hat{\pi}_\beta(a|s)} c_0(s,a) \\ \bar{\Delta}(s) &= \mathbb{E}_{\mu(a|s)} \Delta(s,a) + \mathbb{E}_{\hat{\pi}_\beta(a|s)} \Delta(s,a).\end{aligned}$$

Note that to show the claim, it suffices to show that for sufficient large α, we have

$$(\alpha/2)\bar{c}(s) \geq \bar{\Delta}(s) + \bar{\Delta} \qquad (\forall s). \tag{10}$$

To this end, note that

$$\mathbb{E}_{\hat{\pi}_\beta(a|s)} c(s,a) = \sum_a (\mu(a \mid s) - \hat{\pi}_\beta(a \mid s)) = 0,$$

and

$$\begin{aligned}&\mathbb{E}_{\mu(a|s)} c(s,a) \\ &= \sum_a \left(\frac{\mu(a \mid s) - \hat{\pi}_\beta(a \mid s)}{\hat{\pi}_\beta(a \mid s)}\right) \mu(a \mid s) \\ &= \sum_a \left(\frac{\mu(a \mid s) - \hat{\pi}_\beta(a \mid s)}{\hat{\pi}_\beta(a \mid s)}\right) (\mu(a \mid s) - \hat{\pi}_\beta(a \mid s)) + \sum_a \left(\frac{\mu(a \mid s) - \hat{\pi}_\beta(a \mid s)}{\hat{\pi}_\beta(a \mid s)}\right) \hat{\pi}_\beta(a \mid s) \\ &= \sum_a \left(\frac{\mu(a \mid s) - \hat{\pi}_\beta(a \mid s)}{\hat{\pi}_\beta(a \mid s)}\right) (\mu(a \mid s) - \hat{\pi}_\beta(a \mid s)) \\ &= \sum_a \frac{(\mu(a \mid s) - \hat{\pi}_\beta(a \mid s))^2}{\hat{\pi}_\beta(a \mid s)},\end{aligned}$$

so we have

$$\bar{c}(s) = \sum_a \frac{(\mu(a \mid s) - \hat{\pi}_\beta(a \mid s))^2}{\hat{\pi}_\beta(a \mid s)} = \mathrm{Var}_{\hat{\pi}_\beta(a|s)} \left[\frac{\mu(a \mid s) - \hat{\pi}_\beta(a \mid s)}{\hat{\pi}_\beta(a \mid s)}\right] > 0,$$

where the last inequality holds since $\mu(a \mid s) \neq \hat{\pi}_\beta(a \mid s)$. Thus, for (10) to hold, it suffices to have

$$\alpha \geq 2 \cdot \max_s \left\{ \mathrm{Var}_{\hat{\pi}_\beta(a|s)} \left[\frac{\mu(a \mid s) - \hat{\pi}_\beta(a \mid s)}{\hat{\pi}_\beta(a \mid s)}\right]^{-1} \cdot (\bar{\Delta}(s) + \bar{\Delta}) \right\}.$$

The claim follows. □

Now, let $Z_0 = \tilde{Z}_0$, and let $Z_k = (\mathcal{T}^\pi)^k Z_0$ and $\tilde{Z}_k = (\tilde{\mathcal{T}}^\pi)^k \tilde{Z}_0$. Applying Lemma A.3 with $Z = \tilde{Z}^k$ and $\bar{\Delta} = 4V_{\text{max}}$, we have

$$\begin{aligned}&\mathbb{E}_{\hat{\pi}_\beta(a|s)} F^{-1}_{\tilde{\mathcal{T}}^\pi \tilde{Z}^k(s,a)}(\tau) - \mathbb{E}_{\mu(a|s)} F^{-1}_{\tilde{\mathcal{T}}^\pi \tilde{Z}^k(s,a)}(\tau) \\ &\geq \mathbb{E}_{\hat{\pi}_\beta(a|s)} F^{-1}_{\mathcal{T}^\pi \tilde{Z}^k(s,a)}(\tau) - \mathbb{E}_{\mu(a|s)} F^{-1}_{\mathcal{T}^\pi \tilde{Z}^k(s,a)}(\tau) + \bar{\Delta} \\ &= \mathbb{E}_{\hat{\pi}_\beta(a|s)} F^{-1}_{\mathcal{T}^\pi Z^k(s,a)}(\tau) - \mathbb{E}_{\mu(a|s)} F^{-1}_{\mathcal{T}^\pi Z^k(s,a)}(\tau) + \bar{\Delta} \\ &\qquad + \left(\mathbb{E}_{\hat{\pi}_\beta(a|s)} F^{-1}_{\mathcal{T}^\pi \tilde{Z}^k(s,a)}(\tau) - \mathbb{E}_{\mu(a|s)} F^{-1}_{\mathcal{T}^\pi \tilde{Z}^k(s,a)}(\tau)\right) \\ &\qquad - \left(\mathbb{E}_{\hat{\pi}_\beta(a|s)} F^{-1}_{\mathcal{T}^\pi Z^k(s,a)}(\tau) - \mathbb{E}_{\mu(a|s)} F^{-1}_{\mathcal{T}^\pi Z^k(s,a)}(\tau)\right) \\ &\geq \mathbb{E}_{\hat{\pi}_\beta(a|s)} F^{-1}_{\mathcal{T}^\pi Z^k(s,a)}(\tau) - \mathbb{E}_{\mu(a|s)} F^{-1}_{\mathcal{T}^\pi Z^k(s,a)}(\tau) \\ &\qquad + \bar{\Delta} - 4V_{\text{max}} \\ &= \mathbb{E}_{\hat{\pi}_\beta(a|s)} F^{-1}_{\mathcal{T}^\pi Z^k(s,a)}(\tau) - \mathbb{E}_{\mu(a|s)} F^{-1}_{\mathcal{T}^\pi Z^k(s,a)}(\tau).\end{aligned}$$

The claim follows by taking the limit $k \to \infty$. □

A.4 Proof of Lemma A.1

We first prove a bound on the concentration of the empirical CDF to the true CDF. A similar result has been previously derived in [17]; our proof is based on theirs.

Lemma A.4. *For all $\delta \in \mathbb{R}_{>0}$, with probability at least $1-\delta$, for any $Z \in \mathcal{Z}$, for all $(s,a) \in \mathcal{D}$,*

$$\|F_{\hat{\mathcal{T}}^\pi Z(s,a)} - F_{\mathcal{T}^\pi Z(s,a)}\|_\infty \leq \sqrt{\frac{5|\mathcal{S}|}{n(s,a)} \log \frac{4|\mathcal{S}||\mathcal{A}|}{\delta}} \tag{11}$$

Proof. By the definition of distributional Bellman operator applied to the CDF function, we have that

$$\begin{aligned}
&F_{\hat{\mathcal{T}}^\pi Z(s,a)}(x) - F_{\mathcal{T}^\pi Z(s,a)}(x) \\
&= \sum_{s',a'} \hat{P}(s' \mid s,a)\pi(a' \mid s')F_{\gamma Z(s',a')+\hat{R}(s,a)}(x) - \sum_{s',a'} P(s' \mid s,a)\pi(a' \mid s')F_{\gamma Z(s',a')+R(s,a)}(x).
\end{aligned}$$

Adding and subtracting $\sum_{s',a'} \hat{P}(s' \mid s,a)\pi(a' \mid s')F_{\gamma Z(s',a')+R(s,a)}(x)$ from this expression gives

$$\begin{aligned}
&\sum_{s',a'} \hat{P}(s' \mid s,a)\pi(a' \mid s')\Big(F_{\gamma Z(s',a')+\hat{R}(s,a)}(x) - F_{\gamma Z(s',a')+R(s,a)}(x)\Big) \\
&\qquad + \sum_{s',a'} \Big(\hat{P}(s' \mid s,a) - P(s' \mid s,a)\Big)\pi(a' \mid s')F_{\gamma Z(s',a')+R(s,a)}(x).
\end{aligned}$$

We proceed by bounding the two terms in the summation. For the first term, observe that

$$\begin{aligned}
&F_{\gamma Z(s',a')+\hat{R}(s,a)}(x) - F_{\gamma Z(s',a')+R(s,a)}(x) \\
&= \int \Big[F_{\hat{R}(s,a)}(r) - F_{R(s,a)}(r)\Big] dF_{\gamma Z(s',a')}(x-r) \\
&\leq \int \Big|F_{\hat{R}(s,a)}(r) - F_{R(s,a)}(r)\Big| dF_{\gamma Z(s',a')}(x-r) \\
&\leq \text{sup}_r \Big|F_{\hat{R}(s,a)}(r) - F_{R(s,a)}(r)\Big| \int dF_{\gamma Z(s',a')}(x-r) \\
&= \Big\|F_{\hat{R}(s,a)}(r) - F_{R(s,a)}(r)\Big\|_\infty .
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
&\sum_{s',a'} \hat{P}(s' \mid s,a)\pi(a' \mid s')\Big(F_{\gamma Z(s',a')+\hat{R}(s,a)}(x) - F_{\gamma Z(s',a')+R(s,a)}(x)\Big) \\
&\leq \sum_{s',a'} \hat{P}(s' \mid s,a)\pi(a' \mid s') \Big\|F_{\hat{R}(s,a)}(r) - F_{R(s,a)}(r)\Big\|_\infty \\
&= \Big\|F_{\hat{R}(s,a)}(r) - F_{R(s,a)}(r)\Big\|_\infty
\end{aligned}$$

The second term can be bounded as follows:

$$\begin{aligned}
&\sum_{s',a'} \Big(\hat{P}(s' \mid s,a) - P(s' \mid s,a)\Big)\pi(a' \mid s')F_{\gamma Z(s',a')+R(s,a)}(x) \\
&= \sum_{s'} \Big(\hat{P}(s' \mid s,a) - P(s' \mid s,a)\Big) \sum_{a'} \pi(a' \mid s')F_{\gamma Z(s',a')+R(s,a)}(x) \\
&\leq \Big\|\hat{P}(\cdot \mid s,a) - P(\cdot \mid s,a)\Big\|_1 \cdot \Big\|\sum_{a'} \pi(a' \mid \cdot)F_{\gamma Z(\cdot,a')+R(s,a)}(x)\Big\|_\infty \\
&\leq \Big\|\hat{P}(\cdot \mid s,a) - P(\cdot \mid s,a)\Big\|_1 \cdot \Big\|\sum_{a'} \pi(a' \mid \cdot)\Big\|_\infty \\
&= \Big\|\hat{P}(\cdot \mid s,a) - P(\cdot \mid s,a)\Big\|_1 .
\end{aligned}$$

Together, we have

$$\left|F_{\hat{\mathcal{T}}^\pi Z(s,a)}(x) - F_{\mathcal{T}^\pi Z(s,a)}(x)\right| \leq \left\|F_{\hat{R}(s,a)}(r) - F_{R(s,a)}(r)\right\|_\infty + \left\|\hat{P}(s' \mid s,a) - P(s' \mid s,a)\right\|_1 .$$

Finally, the inequalities can be bounded using the Dvoretzky–Kiefer–Wolfowitz (DKW) inequality and the Hoeffding's inequality, giving us the desired results. By the DKW inequality, we have that with probability $1 - \delta/2$, for all $(s,a) \in \mathcal{D}$,

$$\left\|F_{\hat{R}(s,a)}(r) - F_{R(s,a)}(r)\right\|_\infty \leq \sqrt{\frac{1}{2n(s,a)} \ln \frac{4|\mathcal{S}||\mathcal{A}|}{\delta}}$$

Similarly, by Hoeffding's inequality and an ℓ_1 concentration bound for multinomial distribution[3], we have

$$\max_{s,a} \left\|\hat{P}(\cdot \mid s,a) - P(\cdot \mid s,a)\right\|_1 \leq \sqrt{\frac{2|\mathcal{S}|}{n(s,a)} \ln \frac{4|\mathcal{S}||\mathcal{A}|}{\delta}}$$

The claim follows by combining the two inequalities. □

Next, we prove a general result that translates bounds on CDFs into bounds on quantile functions.

Lemma A.5. *Consider two CDFs F and G with support $\mathcal{X}$. Suppose that F is ζ-strongly monotone and that $\|F - G\|_\infty \leq \epsilon$. Then, $\|F^{-1} - G^{-1}\|_\infty \leq \epsilon/\zeta$.*

Proof. First, note that

$$F^{-1}(y) - G^{-1}(y) = \int_{G^{-1}(y)}^{F^{-1}(y)} dx = \int_{F(G^{-1}(y))}^{y} dF^{-1}(y'),$$

where the first equality follows by fundamental theorem of calculus, and the second by a change of variable $y' = F(x)$. Since $F(F^{-1}(y')) = y'$, we have $F'(F^{-1}(y'))dF^{-1}(y') = dy'$, so

$$dF^{-1}(y') = \frac{dy'}{F'(F^{-1}(y'))} \leq \frac{dy'}{\zeta},$$

where the inequality follows by ζ-strong monotonicity. As a consequence, we have

$$\int_{F(G^{-1}(y))}^{y} dF^{-1}(y') \leq \int_{F(G^{-1}(y))}^{y} \frac{dy'}{\zeta} = \frac{(y - F(G^{-1}(y))}{\zeta} = \frac{G(G^{-1}(y)) - F(G^{-1}(y))}{\zeta} \leq \frac{\epsilon}{\zeta},$$

where the last inequality follows since $\|G - F\|_\infty \leq \epsilon$. The claim follows. □

Finally, Lemma A.1 follows by substituting $F = F_{\hat{\mathcal{T}}^\pi Z(s,a)}(x)$, $G = F_{\mathcal{T}^\pi Z(s,a)}(x)$, and $\epsilon = \sqrt{\frac{5|\mathcal{S}|}{n(s,a)} \log \frac{4|\mathcal{S}||\mathcal{A}|}{\delta}}$ into Lemma A.5, where the condition $\|F - G\|_\infty \leq \epsilon$ holds by Lemma A.4. □

A.5 Proof of Lemma A.2

We prove the following slightly stronger result:

Lemma A.6. *For any $\beta \in \mathbb{R}$, if Z satisfies*

$$F^{-1}_{Z(s,a)}(\tau) \geq F^{-1}_{\mathcal{T}^\pi Z(s,a)}(\tau) + \beta \qquad (\forall \tau \in [0,1]) \tag{12}$$

for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, then we have

$$F^{-1}_{Z(s,a)}(\tau) \geq F^{-1}_{Z^\pi(s,a)}(\tau) + (1-\gamma)^{-1}\beta \qquad (\forall \tau \in [0,1]).$$

The result holds with $\geq$ replaced by $\leq$, or with $\mathcal{T}^\pi$ and Z^π replaced by $\hat{\mathcal{T}}^\pi$ and $\hat{Z}^\pi$ or $\tilde{\mathcal{T}}^\pi$ and $\tilde{Z}^\pi$.

[3] See https://nanjiang.cs.illinois.edu/files/cs598/note3.pdf for a derivation.

Proof. We prove the first case; the cases with $\geq$, and the cases with $\hat{\mathcal{T}}^\pi$ and $\hat{Z}^\pi$ follow by the same argument. First, we show that

$$F_{\mathcal{T}^\pi Z(s,a)}(x) \geq F_{Z(s,a)}(x+\beta) \qquad (\forall x \in [V_{\text{min}}, V_{\text{max}}]). \tag{13}$$

To this end, note that rearranging (12), we have

$$F_{\mathcal{T}^\pi Z(s,a)}(F^{-1}_{Z(s,a)}(\tau) - \beta) \geq \tau.$$

Then, substituting $\tau = F_{\hat{Z}^\pi(s,a)}(x+\beta)$ yields (13); note that such τ must exist since the CDF is defined on all of $\mathbb{R}$. Next, we show that

$$F^{-1}_{\mathcal{T}^\pi Z(s,a)}(\tau) \geq F^{-1}_{\mathcal{T}^\pi(\mathcal{T}^\pi Z(s,a))}(\tau) + \gamma\beta \qquad (\forall \tau \in [0,1]), \tag{14}$$

where the parts changed from (12) are highlighted in red. Intuitively, this claim says that $\mathcal{T}^\pi$ distributes additively to the constant β, and since $\mathcal{T}^\pi$ is a γ-contraction in $\bar{d}_p$, we have $\mathcal{T}^\pi\beta \leq \gamma\beta$. To show (14), first note that

$$\begin{aligned}
F_{\mathcal{T}^\pi(\mathcal{T}^\pi Z(s,a))}(x) &= \sum_{s',a'} P^\pi(s',a' \mid s,a) \int F_{\mathcal{T}^\pi Z(s',a')}\left(\frac{x-r}{\gamma}\right) dF_{R(s,a)}(r) \\
&\geq \sum_{s',a'} P^\pi(s',a' \mid s,a) \int F_{Z(s',a')}\left(\frac{x-r}{\gamma} + \beta\right) dF_{R(s,a)}(r) \\
&= \sum_{s',a'} P^\pi(s',a' \mid s,a) \int F_{\gamma Z(s',a')}(x - r + \gamma\beta) dF_{R(s,a)}(r) \\
&= \sum_{s',a'} P^\pi(s',a' \mid s,a) F_{R(s,a)+\gamma Z(s',a')}(x+\gamma\beta) \\
&= F_{\mathcal{T}^\pi Z(s,a)}(x+\gamma\beta),
\end{aligned}$$

where the first step follows by derivation of the Bellman operator for the CDF, the second step follows from (13), and the third step follows from the property of a CDF function. It follows that

$$F^{-1}_{\mathcal{T}^\pi Z(s,a)}(F_{\mathcal{T}^\pi(\mathcal{T}^\pi Z(s,a))}(x)) \geq x + \gamma\beta.$$

Setting $\tau = F_{\mathcal{T}^\pi(\mathcal{T}^\pi Z(s,a))}(x)$, we have

$$F^{-1}_{\mathcal{T}^\pi Z(s,a)}(\tau) \geq F^{-1}_{\mathcal{T}^\pi(\mathcal{T}^\pi Z(s,a))}(\tau) + \gamma\beta$$

for all $\tau \in [0,1]$; thus, we have shown (14). Now, by induction on $\mathcal{T}^\pi$, we have

$$F^{-1}_{(\mathcal{T}^\pi)^k Z(s,a)}(\tau) \geq F^{-1}_{(\mathcal{T}^\pi)^{k+1} Z(s,a)}(\tau) + \gamma^k\beta$$

for all $k \in \mathbb{N}$. Summing these inequalities over $k \in \{0, 1, ..., n\}$ inequality gives

$$\sum_{k=0}^{n} F^{-1}_{(\mathcal{T}^\pi)^k Z(s,a)}(\tau) \geq \sum_{k=0}^{n} F^{-1}_{(\mathcal{T}^\pi)^k(\mathcal{T}^\pi Z(s,a))}(\tau) + \sum_{k=0}^{n} \gamma^k\beta$$

Subtracting common terms from both sides and evaluating the sum over γ^k, we have

$$F^{-1}_{Z(s,a)}(\tau) \geq F^{-1}_{(\mathcal{T}^\pi)^{n+1} Z(s,a)}(\tau) + \frac{1-\gamma^{n+1}}{1-\gamma}\beta.$$

Taking $n \to \infty$, we have

$$F^{-1}_{Z(s,a)}(\tau) \geq F^{-1}_{Z^\pi(s,a)}(\tau) - (1-\gamma)^{-1}\beta,$$

where we have used the fact that Z^π is the fixed point of $\mathcal{T}^\pi$. The claim follows. □

A.6 Bound on error of the fixed-point of the empirical distributional bellman operator

We can use our techniques to prove finite-sample bounds on the error of using value iteration with the empirical Bellman operator $\hat{\mathcal{T}}$ compared to the true Bellman operator $\mathcal{T}$.

Theorem A.7. *We have $\|F^{-1}_{\hat{Z}^\pi(s,a)} - F_{Z^\pi(s,a)}\|_\infty \leq (1-\gamma)^{-1}\Delta_{max}$, where $\hat{Z}^\pi$ and Z^π are the fixed-points of $\hat{\mathcal{T}}^\pi$ and $\mathcal{T}^\pi$, respectively.*

Proof. Let $\Delta_{\text{max}} = \max_{s,a} \Delta(s,a)$. We have $\|F^{-1}_{\hat{Z}^\pi(s,a)} - F_{\mathcal{T}^\pi \hat{Z}^\pi(s,a)}\|_\infty \leq \Delta_{\text{max}}$ by Lemma A.1 with $Z = \hat{Z}^\pi$. Thus, we have $\|F^{-1}_{\hat{Z}^\pi(s,a)} - F_{Z^\pi(s,a)}\|_\infty \leq (1-\gamma)^{-1}\Delta_{\text{max}}$ by Lemma A.2. □

B Algorithm and implementation details

In this section, we describe our practical implementation of CODAC in detail.

B.1 Actor-Critic objective

We first describe a modification to the CODAC objective, which admits *learnable* α, instead of having to fix it to a constant value throughout the entirety of training. Recall that the original objective is

$$\hat{Z}^{k+1} = \arg\min_{Z} \left\{ \alpha \cdot \mathbb{E}_{U(\tau)} \left[\mathbb{E}_{\mathcal{D}(s)} \log \sum_{a} \exp(F^{-1}_{Z(s,a)}(\tau)) - \mathbb{E}_{\mathcal{D}(s,a)} F^{-1}_{Z(s,a)}(\tau) \right] + \mathcal{L}_p(Z, \hat{\mathcal{T}}^{\pi^k} \hat{Z}^k) \right\},$$

We first provide a derivation of the above objective; this portion largely follows from [21]. We first introduce a *regularization* term $\mathcal{R}(\mu)$ to obtain a well-defined optimization problem:

$$\hat{Z}^{k+1} = \arg\min_{Z} \max_{\mu} \left\{ \alpha \cdot \mathbb{E}_{U(\tau)} \left[\mathbb{E}_{\mathcal{D}(s), \mu(a|s)} F^{-1}_{Z(s,a)}(\tau) - \mathbb{E}_{\mathcal{D}(s,a)} F^{-1}_{Z(s,a)}(\tau) \right] + \mathcal{L}_p(Z, \hat{\mathcal{T}}^{\pi^k} \hat{Z}^k) \right\} + \mathcal{R}(\mu)$$

If we set $\mathcal{R}(\mu)$ to be the entropy $\mathcal{H}(\mu)$, then we can see that $\mu(a \mid s) \propto \exp(Q(s,a)) = \exp(\int_0^1 F^{-1}_{Z(s,a)}(\tau) d\tau)$ is the solution to the inner-maximization. Plugging this choice into the above regularized objective gives

$$\hat{Z}^{k+1} = \arg\min_{Z} \left\{ \alpha \cdot \mathbb{E}_{U(\tau)} \left[\mathbb{E}_{\mathcal{D}(s)} \log \sum_{a} \exp(F^{-1}_{Z(s,a)}(\tau)) - \mathbb{E}_{\mathcal{D}(s,a)} F^{-1}_{Z(s,a)}(\tau) \right] + \mathcal{L}_p(Z, \hat{\mathcal{T}}^{\pi^k} \hat{Z}^k) \right\},$$

as desired. As in [21], we introduce a parameter $\zeta \in \mathbb{R}_{>0}$ that thresholds the quantile value difference between μ and $\hat{\pi}_\beta$. In addition, we scale this difference by $\omega \in \mathbb{R}_{>0}$. This gives a learnable formulation of α via dual gradient descent:

$$\min_{Z} \max_{\alpha \geq 0} \left\{ \alpha \cdot \mathbb{E}_{U(\tau)} \left[\omega \cdot \left[\mathbb{E}_{\mathcal{D}(s)} \log \sum_{a} \exp(F^{-1}_{Z(s,a)}(\tau)) - \mathbb{E}_{\mathcal{D}(s,a)} F^{-1}_{Z(s,a)}(\tau) \right] - \zeta \right] + \mathcal{L}_p(Z, \hat{\mathcal{T}}^{\pi^k} \hat{Z}^k) \right\},$$

Because our experiments are all conducted in continuous-control domains, we cannot enumerate all actions a and compute $\log \sum_a \exp(F^{-1}_{Z(s,a)}(\tau))$ directly. To circumvent this issue, we use the importance sampling approximation scheme introduced in [21]. To this end, we use the following approximation in our implementation:

$$\log \sum_{a} \exp(F^{-1}_{Z(s,a)}(\tau)) \approx \log \left(\frac{1}{2M} \sum_{a_i \sim U(\mathcal{A})}^{N} \left[\frac{\exp(F^{-1}_{Z(s,a)}(\tau))}{U(\mathcal{A})} \right] + \frac{1}{2M} \sum_{a_i \sim \pi(a|s)}^{N} \left[\frac{\exp(F^{-1}_{Z(s,a)}(\tau))}{\pi(a_i \mid s)} \right] \right) \tag{15}$$

where $U(\mathcal{A}) = \text{Uniform}(\mathcal{A})$ denotes the uniform distribution over actions, and where we pick $M = 10$. We summarize a single step of the actor and critic updates used by CODAC in Algorithm 1.

B.2 Neural network architecture

The policy network $\pi(\cdot \mid s; \phi)$ consists of a two-layer fully connected architecture with 256 hidden units and ReLU activations. For the quantile network, we use the architecture from [27], which builds on top of the implicit quantile network (IQN) architecture [6]. Specifically, we represent the quantile function $F^{-1}_{Z(s,a)}(\tau)$ as an element-wise (Hadamard) product of a state-action feature representation $\psi(s,a)$ and a quantile embedding $\varphi(\tau)$—i.e., $F^{-1}_{Z(s,a)}(\tau) = \psi(s,a) \odot \varphi(\tau)$. Following IQN, we use the following embedding formula for $\varphi(\tau)$:

$$\varphi_j(\tau) := h \left(\sum_{i=1}^{n} \cos(i\pi\tau) w_{ij} + b_j \right),$$

where w_{ij}, b_j are weights of the neural network φ, and h is the sigmoid function. We use a one-layer 256-unit fully connected neural network for $\psi(s,a)$, and a one-layer 64-unit fully connected neural network for $\varphi(\tau)$, followed with one-layer 256-unit fully connected network applied to $\psi(s,a) \odot \varphi(\tau)$. We apply layer normalization [2] after each activation layer to ensure stable training.

Algorithm 1 CODAC Update

```
1: **Hyperparameters:** Number of generated quantiles $N$, quantile Huber loss threshold $\kappa$, CODAC penalty scale $\omega$, CODAC penalty threshold $\zeta$, discount rate $\gamma$, learning rates $\eta_{\text{actor}}, \eta_{\text{critic}}, \eta_\alpha$
2: **Parameters:** Critic parameters $\theta$, Actor parameters $\phi$, Penalty $\alpha$
3: **Inputs:** Tuple $s, a, r, s'$
4: Sample quantiles $\tau_i$ (for $i = 1, \dots, N$) and $\tau'_j$ (for $j = 1, \dots, N$) i.i.d. from Uniform($[0, 1]$)
5: # Compute distributional TD loss
6: Get next actions for calculating target $a' \sim \pi(\cdot \mid s'; \phi)$
7: **for** $i = 1$ **to** $N$ **do**
8:     **for** $j = 1$ **to** $N$ **do**
9:         $\delta_{\tau_i,\tau'_j} = r + \gamma F^{-1}_{Z(s',a'),\theta'}(\tau'_j) - F^{-1}_{Z(s,a),\theta}(\tau_i)$
10:     **end for**
11: **end for**
12: Compute $\mathcal{L}_{\text{critic}}(\theta) = N^{-2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathcal{L}_\kappa(\delta_{\tau_i,\tau'_j}; \tau_i)$
13: # Compute CODAC penalty
14: Sample $i \sim U(\{1, ..., N\})$ and use quantile $\tau_i$
15: Estimate $\log \sum_a \exp(F^{-1}_{Z(s,a),\theta}(\tau_i))$ according to (15)
16: Compute $\mathcal{L}_{\text{CODAC}}(\theta, \alpha) = \alpha \cdot \Big(\omega \cdot \Big(\log \sum_a \exp(F^{-1}_{Z(s,a),\theta}(\tau_i)) - N^{-1} \sum_{j=1}^{N} F^{-1}_{Z(s,a),\theta}(\tau_j)\Big) - \zeta\Big)$
17: Update $\theta \leftarrow \theta - \eta_{\text{critic}} \nabla_\theta (\mathcal{L}_{\text{critic}}(\theta) + \mathcal{L}_{\text{CODAC}}(\theta, \alpha))$
18: Update $\alpha \leftarrow \alpha - \eta_\alpha \nabla_\alpha \mathcal{L}_{\text{CODAC}}(\theta, \alpha)$
19: # Update Policy Network $\pi_\phi(a \mid s)$ with $\Phi_g$ objective
20: Get new actions with re-parameterized samples $\tilde{a} \sim \pi(\cdot \mid s; \phi)$
21: Compute$\Phi_g(s, \tilde{a})$ using $F^{-1}_{Z(s,\tilde{a}),\theta}(\tau_i), i = 1, ..., N$
22: $\mathcal{L}_{\text{actor}}(\phi) = \log(\pi(\tilde{a} \mid s; \phi)) - \Phi_g(s, \tilde{a})$
23: Update $\phi \leftarrow \phi + \eta_{\text{actor}} \nabla \mathcal{L}_{\text{actor}}(\phi)$
```

B.3 Actor-Critic updates

We summarize a single actor-critic update performed by CODAC in Algorithm 1. We briefly discuss a few implementation details. First, since computing the CODAC penalty to all quantiles is prohibitively expensive, we apply the conservative penalty to a randomly chosen τ_i on each update step (Line 13-15). This practical choice aligns well with our theoretical objective, whose outer expectation is taken with respect to the uniform distribution $U(\tau)$ over quantiles. We also found subtracting the *average* quantile values (i.e., $N^{-1} \sum_{j=1}^{N} F^{-1}_{Z(s,a),\theta}(\tau_j)$) to be more stable than just subtracting the corresponding quantile value $F^{-1}_{Z(s,a),\theta}(\tau_i)$. This step can be viewed as rewriting

$$\mathbb{E}_{U(\tau)} \left[\mathbb{E}_{\mathcal{D}(s)} \log \sum_a \exp(F^{-1}_{Z(s,a)}(\tau)) - \mathbb{E}_{\mathcal{D}(s,a)} F^{-1}_{Z(s,a)}(\tau) \right]$$

as

$$\mathbb{E}_{U(\tau)} \left[\mathbb{E}_{\mathcal{D}(s)} \log \sum_a \exp(F^{-1}_{Z(s,a)}(\tau)) \right] - \mathbb{E}_{U(\tau)} \left[\mathbb{E}_{\mathcal{D}(s,a)} F^{-1}_{Z(s,a)}(\tau) \right]$$

and implementing the latter as in Line 15. Finally, to compute $\Phi_g(s, \tilde{a})$ in Line 21, we take the average of all $F^{-1}_{Z(s,\tilde{a}),\theta}(\tau_i)$ where τ_i is less than or equal to the risk threshold value. For the expected-return (i.e., risk-neutral objective), the threshold is 1, and $\Phi_g(s, \tilde{a}) = \sum_{i=1}^{N} F^{-1}_{Z(s,\tilde{a}),\theta}(\tau_i)/N$. For CVaR0.1, the threshold is 0.1, and $\Phi_g(s, \tilde{a}) = \sum_{i=1}^{\max_i : \tau_i < 0.1} F^{-1}_{Z(s,\tilde{a}),\theta}(\tau_i)/(\max_i : \tau_i < 0.1)$.

C Experiment details and additional results

C.1 Risky robot navigation

Risky PointMass environment. The state space of the PointMass agent 4-dimensional, including the agent's position as well as the goal position, which is fixed to $[0.1, 0.1]$. The state space constraint

Table 5: CODAC can optimize various distorted expectation based risk-sensitive objectives.

Algorithm	**Risky PointMass**			
	Mean	Median	CVaR$_{0.1}$	Violations
CODAC-CVaR	-6.05	-4.89	**-14.73**	**0.0**
CODAC-CPW	-8.34	**-4.00**	-54.18	103.0
CODAC-Neutral	-8.60	-4.05	-51.96	108.3
CODAC-Wang	**-6.01**	-4.46	-16.80	7.0

Table 6: Risky robot navigation quantitative evaluation.

Algorithm	**Risky PointMass**				**Risky Ant**			
	Mean	Median	CVaR$_{0.1}$	Violations	Mean	Median	CVaR$_{0.1}$	Violations
DSAC (Online)	-7.69	-3.82	-49.9	94	-866.1	-833.3	-1422.7	2247
CODAC-C (Ours)	**-6.05** ± 0.42	-4.89 ± 0.35	**-14.73** ± 0.95	**0.0** ± 0.0	-456.0 ± 24.0	-433.4 ± 17.1	**-686.6** ± 149.8	**347.8** ± 69.7
CODAC-N (Ours)	-8.60 ± 1.62	**-4.05** ± 0.12	-51.96 ± 12.34	108.3 ± 11.90	**-432.7** ± 41.3	**-395.1** ± 11.5	-847.1 ± 309.3	936.0 ± 186.1
ORAAC	-10.67 ± 1.18	-4.55 ± 0.55	-64.12 ± 5.14	138.7 ± 16.4	-788.1 ± 82.0	-795.3 ± 144.4	-1247.2 ± 48.0	1196 ± 49.7
CQL	-7.51 ± 1.05	-4.18 ± 0.13	-43.44 ± 10.57	93.4 ± 0.94	-967.8 ± 66.9	-858.5 ± 22.0	-1887.3 ± 236.1	1854.3 ± 369.1

is $[0, 1]$. Hence, the agent cannot enter a location outside of this unit square. The risky red region is centered at $[0.5, 0.5]$ with radius of 0.3. The agent's initial state is randomly chosen inside the $[0.1, 0.9]^2$ box outside the risky red region. The agent dynamics is holomorphic, allowing the agent to move freely in any direction with its x-axis and y-axis displacement capped at 0.1. The reward the agent receives at each step is its negative Euclidean distance to the goal plus a constant -0.1, which encourages the agent to reach the goal as fast as possible. When the agent is inside the risky red region, with probability 0.1, an additional -50 reward is incurred. The episode terminates when the agent is within 0.1 distance to the goal. An episode may last up to 100 steps.

Risky Ant environment. The state space of the Ant agent is identical to the original state space of the Mujoco Ant agent. The goal is located at $[10, 10]$, and the risky red region is centered at $[5, 5]$ with a radius of 3. The agent's initial state is randomly chosen inside the $[0, 7]^2$ box outside the risky red region. The agent dynamics is also identical to the Mujoco Ant environment. At each timestep, the agent receives its negative Euclidean distance to the goal plus $0.1 \times$ velocity as its reward. When the agent is inside the risky red region, with probability 0.1, an additional -50 reward is incurred. The episode terminates when the agent is within 0.1 distance to the goal. When the agent is inside the risky red region, with probability 0.05, an additional -90 reward is incurred. The episode terminates when the agent is within distance 1 of the goal. An episode may last up to 200 steps.

Dataset and training details. We train a distributional SAC agent online for 100 (resp., 5000) episodes in the PointMass (resp., Ant) environment, and use this agent's replay buffer as the dataset for offline RL training. All offline RL algorithms are trained for 10^4 (resp., 10^6) gradient steps. We use the default hyperparameters for ORAAC, and use $\omega = 0.01$ and $\zeta = 10$ for both CODAC and CQL. Our results are reported using 100 evaluation episodes with same set of initial states.

Additional results. In Table 6, we show full results for the risky robot navigation environments. As can be seen, CODAC-C achieves the best performance on most metrics and is the only method that learns risk-averse behavior. In addition, in Figure 5, we visualize trajectories for various Ant agents. As can be seen, CODAC-C avoids the risky region shown in red, while still making it to the goal.

Alternative risk-sensitive objectives. On the risky pointmass domain, we also show that CODAC can optimize CPW and Wang risk-sensitive objectives using the same offline dataset. As for CODAC-CVaR (CODAC-C) and CODAC-Neutral (CODAC-N), we train CODAC-Wang and CODAC-CPW using 5 random seeds and report the results in Table 5. As shown, CODAC-Wang performs similarly to CODAC-CVaR, trading off slightly better average performance at the cost of safety. On the other hand, CODAC-CPW is on par with CODAC-Neutral. These findings match our intuition that Wang is slightly more risk-seeking than CVaR since it gives non-zero (but vanishingly small) weight to quantile values above the risk cutoff threshold, and CPW is similar to risk-neutral due to its intended modeling of human game-play behavior. These findings are also consistent with those in prior work [6], which investigates these risk objectives for online distributional reinforcement learning.

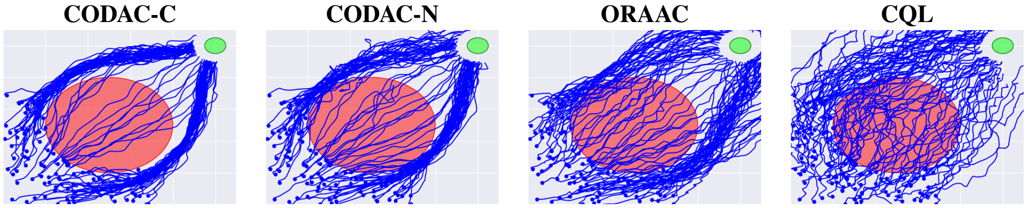

Figure 5: 2D visualization of evaluation trajectories on the Risky Ant environment. The red region is risky, the solid blue circles indicate initial states, and the blue lines are trajectories. CODAC-C learns the most risk-averse behavior while consistently approaching the goal.

Table 7: Normalized Return on the Stochastic D4RL Mujoco Suite, averaged over 5 random seeds.

	Algorithm	Medium		Mixed		Expert	
		Mean	$\text{CVaR}_{0.1}$	Mean	$\text{CVaR}_{0.1}$	Mean	$\text{CVaR}_{0.1}$
Cheetah	CQL	33.2 ± 21.6	-15.0 ± 14.3	214.1 ± 52.0	12.0 ± 23.8	-74.8 ± 22.6	-206.6 ± 46.9
	ORAAC	361.4 ± 14.2	91.3 ± 42.1	307.1 ± 5.8	118.9 ± 27.1	598.3 ± 47.0	99.7 ± 71.3
	CODAC-N	338.9 ± 65.7	-41.6 ± 16.7	347.7 ± 32.3	149.2 ± 79.2	686.3 ± 128.8	123.2 ± 90.1
	CODAC-C	335.8 ± 80.6	-27.7 ± 60.3	396.4 ± 56.1	238.5 ± 58.9	551.6 ± 129.4	151.3 ± 133.0
Hopper	CQL	877.9 ± 193.3	693.0 ± 160.9	189.2 ± 63.0	-21.4 ± 62.5	1165.0 ± 59.4	886.0 ± 132.7
	ORAAC	1007.1 ± 58.5	767.6 ± 101.0	876.3 ± 86.7	524.9 ± 323.0	1156.8 ± 340.5	767.4 ± 372.6
	CODAC-N	993.7 ± 32.9	952.5 ± 29.0	1483.9 ± 16.2	1457.6 ± 20.7	1292.7 ± 34.9	1024.0 ± 45.6
	CODAC-C	1014.0 ± 281.7	976.4 ± 272.1	1551.2 ± 33.4	1449.6 ± 101.3	1270.6 ± 74.8	986.4 ±99.7
Walker2d	CQL	1524.3 ± 87.9	1343.8 ± 248.2	74.3 ± 76.7	-64.0 ± -77.7	2045.2 ± 37.6	1868.2 ±55.1
	ORAAC	1134.1 ± 235.4	663.0 ± 349.8	222.0 ± 37.4	-69.6 ± 76.3	991.2 ± 203.5	108.9 ± 73.2
	CODAC-N	1537.3 ± 65.8	1158.8 ± 357.3	358.7 ± 125.4	106.4 ± 146.9	2170.3 ± 22.7	2035.4 ± 39.9
	CODAC-C	1120.8 ± 319.3	902.3 ± 492.0	450.0 ± 193.2	261.4 ± 231.3	2056.7 ± 43.1	1889.4 ± 28.6

C.2 Stochastic D4RL Mujoco suite

Our experimental protocol largely follows [10]. All algorithms are trained for 500k gradient steps. We use 10 evaluation episodes on the modified Mujoco environments (see below). Hyperparameters are detailed in Appendix C.4.

Dataset descriptions. We describe the stochastic reward modification made to the original HalfCheetah, Hopper, and Walker2d environments [43]. These reward modifications are used to relabel the reward label in D4RL datasets; the modified environments are also used for evaluation in this set of experiments. The following paragraphs are adapted from [43]:

- **Half-Cheetah:** We use $R_t(s,a) = \bar{r}_t(s,a) - 70 \cdot \mathbb{1}_{v>\bar{v}} \cdot \mathcal{B}_{0.1}$, where $\bar{r}_t(s,a)$ is the original environment reward, v is the forward velocity, and $\bar{v}$ is a threshold velocity ($\bar{v} = 4$ for Medium/Mixed datasets and $\bar{v} = 10$ for the Expert dataset). The maximum episode length is reduced to 200 steps.
- **Walker2D/Hopper:** We use $R_t(s,a) = \bar{r}_t(s,a) - p \cdot \mathbb{1}_{|\theta|>\bar{\theta}} \cdot \mathcal{B}_{0.1}$, where $\bar{r}_t(s,a)$ is the original environment reward, θ is the pitch angle, $\bar{\theta}$ is a threshold angle ($\bar{\theta} = 0.5$ for Walker2d and $\bar{\theta} = 0.1$ for Hopper) and $p = 30$ for Walker2d and $p = 50$ for Hopper. When $|\theta|> 2\bar{\theta}$ the robot falls, the episode terminates. The maximum episode length is reduced to 500 steps.

Additional results. In Table 7, we present the full Stochastic D4RL Mujoco results, including results on the Expert dataset. We repeat the results on the Medium and Mixed datasets in the main text here for completeness. Recall that the Expert (resp., Medium) dataset consists of rollouts from a fixed SAC agent trained to Expert (resp., Medium) performance, Expert is convergence and Medium is 50% of Expert performance. The Mixed dataset is the replay buffer of a SAC agent trained to achieve 50% of the expert return.

Qualitative analysis. To better interpret the stochastic D4RL results, we have collected behavioral statistics of the agents trained on the risk-sensitive HalfCheetah-Mixed-v0 and Walker2d-Mixed-v0 datasets. We execute one trained agent for each method reported in Table 2 for 10 episodes in the environment and record the percentage of timesteps where the agent violates the threshold and their average velocity over these evaluation episodes.

Table 8: Stochastic D4RL qualitative results

Algorithm	HalfCheetah-Mixed-v0		Walker2d-Mixed-v0	
	% Violation	Average Velocity	% Violation	Average Velocity
CODAC-C (Ours)	**11**	**1.49**	15	**0.28**
CODAC-N (Ours)	54	2.02	**9**	0.34
CQL	23	1.71	13	0.19
ORAAC	37	1.76	48	0.49

Table 9: Normalized Return on the D4RL Mujoco Suite, averaged over 5 random seeds.

Dataset	BC	BEAR	BRAC-v	BCQ	MOPO	CQL	ORAAC	CODAC
halfcheetah-random	2.1	25.1	24.1	2.2	**35.4**	35.4	13.5	34.6 ± 1.27
hopper-random	9.8	11.4	**12.2**	10.6	11.7	10.8	9.8	11 ± 0.43
walker2d-random	1.6	7.3	1.9	4.9	13.6	7.0	3.2	**18.7** ± 4.5
halfcheetah-medium	36.1	41.7	43.8	40.7	42.3	44.4	41.0	**46.3**± 0.98
walker2d-medium	6.6	59.1	81.1	53.1	17.8	79.2	27.3	**82.0** ± 0.45
hopper-medium	29.0	52.1	31.1	54.5	28.0	58.0	1.48	**70.8** ± 11.4
halfcheetah-mixed	38.4	38.6	47.7	38.2	**53.1**	46.2	30.0	44 ± 0.76
hopper-mixed	11.8	33.7	0.6	33.1	67.5	48.6	16.3	**100.2** ± 1.0
walker2d-mixed	11.3	19.2	0.9	15.0	**39.0**	26.7	28	33.2 ± 17.6
halfcheetah-medium-expert	35.8	53.4	41.9	64.7	63.3	62.4	24.0	**70.4** ± 19.4
walker2d-medium-expert	6.4	40.1	81.6	57.5	44.6	98.7	28.2	**106.0** ± 4.6
hopper-medium-expert	111.9	96.3	0.8	110.9	23.7	111.0	18.2	**112.0** ± 1.7

As shown in Table 8, CODAC-C achieves the lowest percentage of violations in the HalfCheetah environment, indicating that it has learned a safer policy than all other methods. On Walker2d, CQL appears to be the safest; however, this result is due to the fact that CQL failed to learn the desirable walking behavior as indicated by its low reward in the paper. Among the methods that learned to walk, CODAC-C achieves the lowest average angular velocity while maximizing the return.

C.3 D4RL Mujoco suite

Our experimental protocol largely follows from [10]. All algorithms are trained for 1M gradient steps. We use 10 evaluation episodes on the original Mujoco environments, which all last 1000 steps long. Hyperparameters are detailed in Appendix C.4. In Table 9, we show the full results on the risk-neutral D4RL Mujoco Suite, which includes additional baselines such as BEAR [20] and BRAC [46].

C.4 Hyperparameters

As CODAC builds on top of distributional SAC (DSAC), we keep the DSAC-specific hyperparameters identical as the original work. These hyperparameters are shown in Table 10.

CODAC additionally introduces hyperparameters α, ω, ζ (see Appendix B). In most cases, α is a learnable parameter initialized to 1 with learning rate $\eta_\alpha = 3 \times 10^{-4}$; in few cases, we fix it to 1 throughout the entirety of training, which we indicate by setting $\zeta = -1$, as in [21]. For ORAAC, we use the default hyperparameters tuned on the stochastic D4RL Mujoco suite for all experiments; for CQL, we use the default hyperparameters tuned on the original D4RL Mujoco suite for all experiments. Below, we describe the specific CODAC hyperparameters we use for the risk-neutral and risk-sensitive D4RL experiments.

Risk-neutral D4RL. We restrict the search range of the hyperparameters as follow: $\omega \in \{0.1, 1, 10\}, \zeta \in \{-1, 10\}$. We also experiment with enabling entropy tuning in DSAC and tune the value network learning rate η_{critic} between $3e-4$ and $3e-5$, which improves performance on some datasets. Table 11 summarizes the hyperparameters used for each dataset in our reported results. At a high level, we find $\omega = 1$ to be effective for Mixed and Random datasets and $\omega = 10$ effective for Medium and Medium-Expert datasets. These empirical findings match our intuition that the penalty needs not to be high when the dataset has wide coverage.

Table 10: CODAC backbone hyperparameters

Hyper-parameter	Value
Policy network learning rate η_{actor}	3e-4
(Quantile) Value network learning rate η_{critic}	3e-5
Optimizer	Adam
Discount factor γ	0.99
Target smoothing	5e-3
Batch size	256
Replay buffer size	1e6
Minimum steps before training	1e4
Number of quantile fractions N	32
Quantile fraction embedding size	64
Huber regression threshold κ	1

Table 11: CODAC hyperparameters for risk-neutral D4RL

dataset	ω	ζ	η_{critic}	entropy tuning
halfcheetah-random	1	10	3e-5	yes
hopper-random	1	10	3e-5	yes
walker2d-random	1	10	3e-5	yes
halfcheetah-medium	10	10	3e-5	no
hopper-medium	10	10	3e-4	yes
walker2d-medium	10	10	3e-5	no
halfcheetah-mixed	1	10	3e-5	yes
hopper-mixed	1	10	3e-5	yes
walker2d-mixed	1	10	3e-5	yes
halfcheetah-medium-expert	0.1	-1	3e-4	no
hopper-medium-expert	10	10	3e-5	no
walker2d-medium-expert	10	10	3e-5	no

Risk-sensitive D4RL. We use the same hyperparameter range as in risk-neutral D4RL for a grid search. Interestingly, the best value of ω is smaller across most datasets, suggesting less conservatism may be needed due to the increased stochasticity in the environment. Table 12 summarizes the hyperparameter choices.

C.5 Compute resources

We use a single Nvidia 2080-Ti with 32 cores to run our experiments. Each CODAC run takes about 10 hours in clock time.

Table 12: CODAC hyperparameters for risk-sensitive D4RL

dataset	ω	ζ	η_{critic}	entropy tuning
halfcheetah-medium	1	-1	3e-5	no
hopper-medium	0.1	10	3e-5	yes
walker2d-medium	1	-1	3e-5	yes
halfcheetah-mixed	0.1	10	3e-5	yes
hopper-mixed	1	10	3e-5	yes
walker2d-mixed	1	10	3e-5	yes
halfcheetah-medium-expert	1	-1	3e-5	yes
hopper-medium-expert	10	10	3e-5	no
walker2d-medium-expert	10	10	3e-5	no