Scientific Research Publishing

Advances in Pure Mathematics, 2025, 15(4), 269-290
https://www.scirp.org/journal/apm
ISSN Online: 2160-0384
ISSN Print: 2160-0368

Offline Robustness of Distributional Actor-Critic Ensemble Reinforcement Learning

Zhongcui Ma[1], Dandan Lai[1], Jianxiang Zhu[1], Yaxin Peng[2*]

[1]Department of Mathematics, College of Sciences, Shanghai University, Shanghai, China
[2]School of Future Technology, and Department of Mathematics, College of Sciences, Shanghai University, Shanghai, China
Email: mzc1217@shu.edu.cn, 2328140942@shu.edu.cn, zjx0828@shu.edu.cn, *yaxin.peng@shu.edu.cn

How to cite this paper: Ma, Z.C., Lai, D.D., Zhu, J.X. and Peng, Y.X. (2025) Offline Robustness of Distributional Actor-Critic Ensemble Reinforcement Learning. *Advances in Pure Mathematics*, **15**, 269-290.
https://doi.org/10.4236/apm.2025.154013

Received: March 12, 2025
Accepted: April 15, 2025
Published: April 18, 2025

Abstract

Offline reinforcement learning (RL) focuses on learning policies using static datasets without further exploration. With the introduction of distributional reinforcement learning into offline RL, current methods excel at quantifying the risk and ensuring the security of learned policies. However, these algorithms cannot effectively balance the distribution shift and robustness, and even a minor perturbation in observations can significantly impair policy performance. In this paper, we propose the algorithm of Offline Robustness of Distributional actor-critic Ensemble Reinforcement Learning (ORDER) to improve the robustness of policies. In ORDER, we introduce two approaches to enhance the robustness: 1) introduce the smoothing technique to policies and distribution functions for states near the dataset; 2) strengthen the quantile network. In addition to improving the robustness, we also theoretically prove that ORDER converges to a conservative lower bound, which can alleviate the distribution shift. In our experiments, we validate the effectiveness of ORDER in the D4RL benchmark through comparative experiments and ablation studies.

Keywords

Offline Reinforcement Learning, Distributional Reinforcement Learning, Robustness

1. Introduction

Offline reinforcement learning (RL) [1]-[3] concerns the problem of learning a policy from a fixed dataset without further interactions. Offline RL can reduce risk

and costs since it eliminates the need for online interaction. In this way, offline RL can be well used in real-world applications such as autonomous driving [4], healthcare [5] and robot control [6].

Applying the standard policy improvement approaches to an offline dataset typically results in the distribution shift problem, making offline RL a challenging task [7] [8]. Some prior works have relieved this issue by penalizing the action-value of the out-of-distribution (OOD) actions [9]-[11]. Nevertheless, simply learning the expectation of action-value is unable to quantify risk and ensure that the learned policy acts safely. To overcome this problem, some efforts have been made to import distributional RL [12]-[14] into offline RL to learn the full distribution over future returns, which is used to make plans to avoid risky and unsafe actions. In addition, with the establishment of risk-sensitive objectives [14], distributional offline RL [15] [16] learns state representations better since they can acquire richer distributed signals, making them superior to traditional reinforcement learning algorithms even on risk-seeking and risk-averse objectives.

Unfortunately, research on distributional offline RL is less complete. CODAC [15] brings distributional RL into the offline setting by penalizing the predicted return quantiles for OOD actions. Meanwhile, MQN-CQR [16] learns a worst-case policy by optimizing the conditional value-at-risk of the distributional value function. However, existing distributional offline RL methods only focus on the safety of the learned policy. These methods leverage a conservative return distribution to impair the robustness, and will make policies highly sensitive, even a minor perturbation in observations [17]. As a result, merely possessing safety can not make a fine balance between conservatism and robustness, which does not pay enough attention to robustness.

In this paper, we propose Offline Robustness of Distributional actor-critic Ensemble Reinforcement Learning (ORDER) by introducing a smoothing technique to quantile networks. Firstly, we consider the dynamic entropy regularizer of the quantile function instead of an unchangeable constant to ensure sufficient exploration. Secondly, the increasing number of quantile networks is also beneficial in obtaining a more robust distribution value function. Thirdly, smooth regularization is brought into the distribution and policies of states near the dataset. In theory, we prove that ORDER obtains a uniform lower bound on all integrations of the quantiles with the distribution soft Bellman operator, which controls the distribution shift. Such bound also achieves the same effect for both expected returns and risk-sensitive objectives. Overall, ORDER can mitigate the OOD problem and simultaneously balance conservatism and robustness.

In our experiments, ORDER outperforms the existing distributional offline RL methods in the D4RL benchmark [18]. Meanwhile, our algorithm is also competitive against the current advanced algorithms. Our ablation experiments demonstrate that strengthening the quantile network is critical to the performance of ORDER. In addition, choosing different risk measure functions does not have a great impact on the performance of ORDER, which also shows the robustness of

our method.

2. Preliminaries

2.1. Markov Decision Process and Offline Reinforcement Learning

Consider an episodic Markov decision process (MDP) $M=(\mathcal{S},\mathcal{A},\mathbb{P},T,R,\gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathbb{P}(s'|s,a)$ is the transition distribution, T is the length of the episode, $R(s,a)$ is the reward function, and γ is the discount factor. For a stochastic policy $\pi(a|s):\mathcal{S}\times\mathcal{A}\to\mathbb{R}$, action-value function Q^{π} is defined as

$$Q^{\pi}(s,a):=\mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty}\gamma^{t}R(s_t,a_t)\right]|\,a_t\sim\pi(\cdot|s_t),s_{t+1}\sim\mathbb{P}(\cdot|s_t,a_t),s_0=s,a_0=a.$$

Standard RL aims at learning the optimal policy $\pi^{\star}$ such that $Q^{\pi^{\star}}(s,a)\geq Q^{\pi}(s,a)$ for all $s\in\mathcal{S},a\in\mathcal{A}$ and all π. The corresponding Q-function of the policy satisfies the Bellman operator:

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

In the offline setting, the agent is not allowed to interact with the environment [18]. The objective of agents is to learn an optimal policy only from a fixed dataset $\mathcal{D}:=\{(s,a,r,s')\}$. For all states $s\in\mathcal{D}$, let $\hat{\pi}(a|s):=\frac{\sum_{s',a'\in\mathcal{D}}1_{(s'=s,a'=a)}}{\sum_{s'\in\mathcal{D}}1_{s'=s}}$ be the empirical behavior policy. In order to avoid the situation where the denominator of the fraction is 0 in the theoretical analysis, we assume that $\hat{\pi}(a|s)>0$. Broadly, actions that are not drawn from $\hat{\pi}$ (*i.e.*, those with low probability density) are the out-of-distribution (OOD).

2.2. Distributional Reinforcement Learning

Distributional RL directly models the full distribution of returns $Z^{\pi}(s,a)=\sum_{t=0}^{\infty}\gamma^{t}R(s_t,a_t)$, instead of merely learning its expected value [19]. The distributional Bellman operator for policy evaluation is $\mathcal{T}^{\pi}Z(s,a):\overset{D}{=}R(s,a)+\gamma Z(s',a')$, where $\overset{D}{=}$ indicates equality in distribution. Define the quantile function be the inverse of the cumulative density function $F_Z(z)=Pr(Z<z)$ as $F_Z^{-1}(\tau):=\inf\{z\in\mathbb{R}:\tau\leq F_Z(z)\}$ [20], where τ denotes the quantile fraction. For random variables U and V with quantile functions F_U^{-1} and F_V^{-1}, the p-Wasserstein distance $W_p(U,V)=\left(\int_0^1\left|F_U^{-1}(\tau)-F_V^{-1}(\tau)\right|^p\mathrm{d}\tau\right)^{1/p}$ is the L_p metric on quantile functions. [19] gave that the distributional Bellman operator $\mathcal{T}^{\pi}$ is a γ-contraction in the W_p, *i.e.*, let $\bar{d}_p(Z_1,Z_2):=\sup_{s,a}W_p(Z_1(s,a),Z_2(s,a))$ be the largest Wasserstein distance over (s,a), and $\mathcal{Z}=\left\{Z:\mathcal{S}\times\mathcal{A}\to\mathcal{P}(\mathbb{R})\,|\,\forall(s,a),\mathbb{E}\left[\left|Z(s,a)\right|^p\right]<\infty\right\}$ be the space of distributions over $\mathbb{R}$ with bounded p-th moment, then

$$\bar{d}_p\left(\mathcal{T}^\pi Z_1, \mathcal{T}^\pi Z_2\right) \le \gamma \cdot \bar{d}_p\left(Z_1, Z_2\right) \quad \forall Z_1, Z_2 \in \mathcal{Z}.$$

Fitted distributional evaluation (FDE) [15] approximates $\mathcal{T}^\pi$ by $\hat{\mathcal{T}}^\pi$ using $\mathcal{D}$, then Z^π can be estimated by starting from an arbitrary $\hat{Z}^0$ and iteratively computing

$$\hat{Z}^{k+1} = \arg\min_Z \mathcal{L}_p\left(Z, \hat{\mathcal{T}}^\pi \hat{Z}^k\right),$$

$$\text{where } \mathcal{L}_p\left(Z, Z'\right) = \mathbb{E}_{\mathcal{D}(s,a)}\left[W_p\left(Z(s,a), Z'(s,a)\right)^p\right].$$

In distributional RL, let risk measure function $\Phi : \mathcal{Z} \to \mathbb{R}$ be a map from the value distribution space to real numbers. Given a distorted function $g(\tau)$ over $[0,1]$, the distorted expectation of Z is

$$\Phi_g\left(Z(s,a)\right) = \int_0^1 F^{-1}_{Z(s,a)}(\tau) g(\tau) \mathrm{d}\tau,$$

and the corresponding policy is $\pi_g(s) \coloneqq \arg\max_a \Phi_g\left(Z(s,a)\right)$ [13]. Specially, if $g = \text{Uniform}\left([0,1]\right)$, then $Q^\pi(s,a) = \Phi_g\left(Z(s,a)\right)$. For other choices of $g(\tau)$, please refer to Section 5.2.

2.3. Robust Reinforcement Learning

Robust RL learns the policy by introducing worst-case adversarial noise to the system dynamics and formulating the noise distribution as the solution of a zero-sum minimax game. In order to learn robust policy π, SR^{2}L [21] obtains $\hat{s} = \arg\max_{\hat{s} \in \mathbb{B}_d(s,\epsilon)} D_{\mathrm{J}}\left(\pi(\cdot \mid s) \| \pi(\cdot \mid \hat{s})\right)$ by adding a perturbation to the state s, where $\mathbb{B}_d(s,\epsilon) = \left\{\hat{s} : d(s,\hat{s}) \le \epsilon\right\}$ and the Jeffrey's divergence D_{J} for two distributions P and Q is defined by $D_{\mathrm{J}}(P \| Q) = \frac{1}{2}\left[D_{\mathrm{KL}}(P \| Q) + D_{\mathrm{KL}}(Q \| P)\right]$, and then define a smoothness regularizer for policy as $\mathcal{R}_s^\pi = \mathbb{E}_{s \sim \rho^\pi} \max_{\hat{s} \in \mathbb{B}_d(s,\epsilon)} D_{\mathrm{J}}\left(\pi(\cdot \mid s) \| \pi(\cdot \mid \hat{s})\right)$. Analogously, $\mathcal{R}_s^Q = \mathbb{E}_{s \sim \rho^\pi} \max_{\hat{s} \in \mathbb{B}_d(s,\epsilon)} \left(Q(s,a) - Q(\hat{s},a)\right)^2$ is the smoothness regularizer for Q-function, where ρ^π is state distribution.

3. Offline Robustness of Distributional Actor-Critic Ensemble

In ORDER, we first obtain a state with adversarial perturbations, and then introduce the smoothness regularization to both the policy and the distribution action-value function for states with adversarial noises. The smooth regularization can be used to learn a smooth Z-function and generate a smooth policy, which makes the algorithm robust. However, the introduction of smoothness could potentially result in an overestimation of values at the boundaries of the supported dataset. To overcome this problem, we incorporate a penalty factor for OOD actions to reduce the quantile values of these actions. In addition, we strengthen the quantile network by increasing the number of quantile networks, which is also beneficial to the robustness of our algorithm. The overall architecture of ORDER is shown in **Figure 1**.

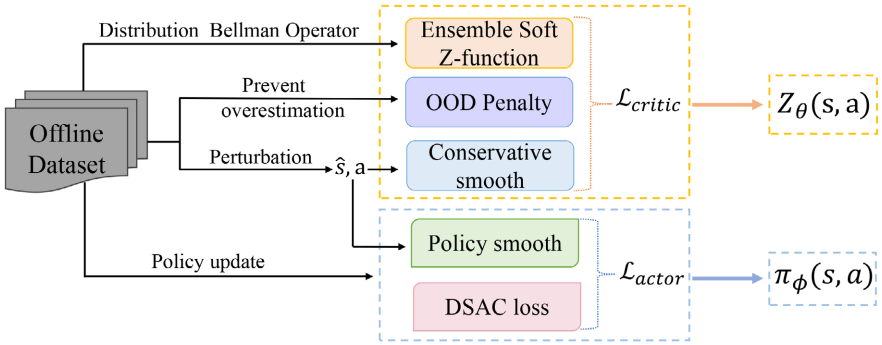

Figure 1. Architecture diagram for ORDER.

3.1. Robust Distributional Action-Value Function

In this part, we sample three sets of state-action pairs and form three different loss functions to obtain a conservative smooth policy. First of all, we construct a perturbation set $\mathbb{B}_d(s,\epsilon)$ to obtain $(\hat{s},a)$ pairs, where $\mathbb{B}_d(s,\epsilon)$ is an ϵ-radius ball measured in metric $d(\cdot,\cdot)$ and $\hat{s}\in\mathbb{B}_d(s,\epsilon)$. Then we sample $(s,\hat{a})$ pairs from the current policy π_ϕ, where $\hat{a}\sim\pi_\phi(s)$. ORDER contains M Z-function and denotes the parameters of the m-th Z-function and the target Z-function as θ_m and θ'_m, respectively. With the help of the constructions, we will give the different learning targets for (s,a), $(\hat{s},a)$ and $(s,\hat{a})$ pairs, respectively.

SAC [7] incorporates an entropy term in its objective function to optimize both cumulative rewards and policy stochasticity, enhancing policy robustness and generalization. We also introduce an entropy term, for a (s,a) pair sampled from $\mathcal{D}$, we obtain the target as

$$\hat{\mathcal{T}}^{\pi}_{\mathrm{DS}}Z_{\theta_m}(s,a) := R(s,a)+\gamma\left[\min_{j=1,\cdots,M} Z_{\theta'_j}(s',a')-c\cdot\log\pi(a'\mid s')\right],$$

where the next Z-function takes the minimum value among the target Z-functions. The loss function is defined as follows,

$$\mathcal{L}_Z(\theta_m)=\mathcal{L}_p\left(Z_{\theta_m},\hat{\mathcal{T}}^{\pi}_{\mathrm{DS}}Z_{\theta_m}\right).$$

Next, we introduce the smoothness regularizer term to the distribution action-value function that is designed to enhance the smoothness of the Z-function. Specifically, we minimize the difference between $Z_{\theta_m}(\hat{s},a)$ and $Z_{\theta_m}(s,a)$, where $(\hat{s},a)$ is a state-action pair with a perturbed state. Then we take an adversarial $\hat{s}\in\mathbb{B}_d(s,\epsilon)$ which maximizes $\mathcal{L}_p\left(Z_{\theta_m}(\hat{s},a),Z_{\theta_m}(s,a)\right)$. The final smooth term we introduced is shown below:

$$\mathcal{L}_{\mathrm{smooth}}(s,a;\theta_m)=\max_{\hat{s}\in\mathbb{B}_d(s,\epsilon)}\varrho\cdot\mathcal{L}_p\left(Z_{\theta_m}(\hat{s},a),Z_{\theta_m}(s,a)\right), \tag{1}$$

where $\varrho\in[0,1]$ is a factor that balances the learning of in-distribution and out-of-distribution values. Thus, for the selected $\hat{s}\in\mathbb{B}_d(s,\epsilon)$, we minimize Equation (1) to get a smooth Z-function. Since the actions should be near the offline data and close to the behavior actions in the dataset, we do not consider OOD action for smoothing in this part.

Finally, we consider the following loss function to prevent overestimation of OOD actions.

$$\mathcal{L}_{\text{OOD}}\left(\theta_m,\beta\right)=\beta\cdot\mathbb{E}_{U(\tau),\mathcal{D}(s,a)}\left[c_0\left(s,a\right)\cdot F^{-1}_{Z_{\theta_m}(s,a)}\left(\tau\right)\right],$$

for some state-action dependent scale factor c_0 and $U=\text{Uniform}\left(\left[0,1\right]\right)$.

Incorporating both the in-distribution target and OOD target, we conclude the loss function in ORDER as follows,

$$\min_{\theta_m}\mathbb{E}_{s,a,r,s'\sim\mathcal{D}}\left[\mathcal{L}_Z\left(\theta_m\right)+\alpha\mathcal{L}_{\text{smooth}}\left(s,a;\theta_m\right)+\mathcal{L}_{\text{OOD}}\left(\theta_m,\beta\right)\right]. \tag{2}$$

3.2. Robust Policy

With the above smooth limits, we can learn a robust policy with fewer policy changes under perturbations. We choose a state $\hat{s}\in\mathbb{B}_d\left(s,\epsilon\right)$ as mentioned above which is maximizing $D_{\text{J}}\left(\pi_\phi\left(\cdot\,|\,s\right)\|\,\pi_\phi\left(\cdot\,|\,\hat{s}\right)\right)$. Consequently, our loss function for policy is as follows:

$$\min_\phi\left[\mathbb{E}_{s,a,r,s'\sim\mathcal{D}}\left[-\min_{j=1,\cdots,M}\Phi_g^j\left(s,a\right)+\alpha_1\max_{\hat{s}\in\mathbb{B}_d(s,\epsilon)}D_{\text{J}}\left(\pi_\phi\left(\cdot\,|\,s\right)\|\,\pi_\phi\left(\cdot\,|\,\hat{s}\right)\right)+\log\pi_\phi\left(a\,|\,s\right)\right]\right], \tag{3}$$

where the first term is designed to get a conservative policy by maximizing the minimum of the distributional functions ensemble and the last term is a regularization term.

3.3. Implementation Details

In this subsection, we integrate the distributional evaluation and policy improvement algorithms introduced in Section 3.1 and Section 3.2 into the actor-critic framework. With the loss function introduced in Equation (2), it is natural to get the following iterative formula for Z^π by starting from an arbitrary $\tilde{Z}^0$,

$$\begin{aligned}\tilde{Z}^{k+1}=\arg\min_Z\Big\{&\mathcal{L}_p\left(Z\left(s,a\right),\hat{\mathcal{T}}^\pi_{\text{DS}}\hat{Z}^k\left(s,a\right)\right)+\max_{\hat{s}}\varrho\cdot\mathcal{L}_p\left(Z\left(\hat{s},a\right),Z\left(s,a\right)\right)\\&+\beta\cdot\mathbb{E}_{U(\tau),\mathcal{D}(s,a)}\left[c_0\left(s,a\right)\cdot F^{-1}_{Z(s,a)}\left(\tau\right)\right]\Big\}.\end{aligned} \tag{4}$$

Following [17], we suggest employing a min-max objective in which the inner loop selects the current policy to maximize the objective, while the outer loop is responsible for minimizing the objective with respect to this policy:

$$\begin{aligned}\tilde{Z}^{k+1}=\arg\min_Z\max_\mu\Big\{&\beta\cdot\mathbb{E}_{U(\tau)}\left[\mathbb{E}_{\mathcal{D}(s),\mu(\tilde{a}|s)}F^{-1}_{Z(s,\tilde{a})}\left(\tau\right)-\mathbb{E}_{\mathcal{D}(s,a)}F^{-1}_{Z(s,a)}\right]\\&+\mathcal{L}_p\left(Z\left(s,a\right),\hat{\mathcal{T}}^{\pi^k}_{\text{DS}}\hat{Z}^k\left(s,a\right)\right)+\max_{\hat{s}}\varrho\cdot\mathcal{L}_p\left(Z\left(\hat{s},a\right),Z\left(s,a\right)\right)\Big\},\end{aligned}$$

where μ is an actor policy. To establish a well-posed optimization problem, we introduce a regularization term in the original objective. Detailed analysis procedures are provided in Appendix A.1. The final optimization objective becomes

$$\begin{aligned}\tilde{Z}^{k+1}=\arg\min_Z\Big\{&\beta\cdot\mathbb{E}_{U(\tau)}\left[\mathbb{E}_{\mathcal{D}(s)}\log\sum_a\exp\left(F^{-1}_{Z(s,a)}\left(\tau\right)\right)-\mathbb{E}_{\mathcal{D}(s,a)}F^{-1}_{Z(s,a)}\left(\tau\right)\right]\\&+\mathcal{L}_p\left(Z\left(s,a\right),\hat{\mathcal{T}}^{\pi^k}_{\text{DS}}\hat{Z}^k\left(s,a\right)\right)+\max_{\hat{s}}\varrho\cdot\mathcal{L}_p\left(Z\left(\hat{s},a\right),Z\left(s,a\right)\right)\Big\},\end{aligned}$$

where $U = \text{Uniform}([0,1])$. To perform optimization with respect to the distribution Z, we express the quantile function using a neural network $G_\theta(\tau;s,a) \approx F^{-1}_{Z(s,a)}(\tau)$. In order to calculate $\mathcal{L}_p(Z,Z') = W_p(Z,Z')^p$ [22], we minimize the weighted pairwise Huber regression loss of various quantile fractions. τ-Huber quantile regression loss [23] with threshold κ is represented as

$$\mathcal{L}_\kappa(\delta;\tau) = \begin{cases} \left|\tau - 1\{\delta < 0\}\right| \cdot \delta^2/(2\kappa), & \text{if } |\delta| \le \kappa, \\ \left|\tau - 1\{\delta < 0\}\right| \cdot (|\delta| - \kappa/2), & \text{otherwise,} \end{cases}$$

where $\tau, \tau' \sim U$ are random quantiles, $\delta = r + \gamma G_{\theta'}(\tau';s',a') - G_\theta(\tau;s,a)$, $(s,a,r,s') \sim \mathcal{D}$, $a' \sim \pi(\cdot \mid s')$. More details for the algorithm ORDER are presented in Algorithm 1.

3.4. Theoretical Analysis

Before presenting our theorems, similar to [15], we first give some assumptions about the MDP and dataset. Next, we assume that the search space in Equation (4) includes all possible functions.

Assumption 1. *For all* $s \in \mathcal{S}$ *and* $a \in \mathcal{A}$, $F_{Z^\pi(s,a)}$ *is smooth. Furthermore, the search space of the minimum over Z in Equation* (4) *is overall smooth functions* $F_{Z(s,a)}$.

The assumption is given to guarantee the boundness of the p-th moments of $Z^\pi(s,a)$ as well as $Z^\pi \in \mathcal{Z}$.

Assumption 2. *For all* $s \in \mathcal{S}$ *and* $a \in \mathcal{A}$, *there exists* $\zeta \in \mathbb{R}_{>0}$, *such that* $F_{Z^\pi(s,a)}$ *is* ζ*-strongly monotone, i.e.,* $F'_{Z^\pi(s,a)}(x) \ge \zeta$.

This assumption is only designed to ensure the convergence of $F^{-1}_{Z^\pi(s,a)}(x)$ in our theoretical analysis. In algorithm implementation, we introduce a policy smoothing term and choose the minimum value of the *Z*-function in the ensemble to restrict the policy update amplitude, thus implicitly satisfying monotonicity. Next, we assume that the infinite norm of two quantiles with perturbation is restricted by a fixed constant.

Assumption 3. *For all* $s \in \mathcal{S}$, $a \in \mathcal{A}$ *and the selected* $\hat{s} \in \mathbb{B}_d(s,\epsilon)$, *we assume that* $\left\| F^{-1}_{Z(\hat{s},a)} - F^{-1}_{Z(s,a)} \right\|_\infty \le \sigma$, σ *is constant.*

In the above assumption, $\|\cdot\|_\infty$ represents the infinite norm. Since perturbed observation $\hat{s}$ is randomly sampling from an ℓ_∞ ball of norm ϵ, the assumption is reasonable.

Finally, we assume $p > 1$. Therefore, we derive the following lemma which characterizes the conservative distribution soft evaluation iterates $Z(s,a)$ with the distributional soft Bellman operator.

Lemma 1. *Suppose Assumptions 1 - 3 hold. For all* $s \in \mathcal{D}$, $a \in \mathcal{A}$, $k \in \mathbb{N}$, *and* $\tau \in [0,1]$, *we have* $F^{-1}_{Z(s,a)}(\tau) = F^{-1}_{\hat{\mathcal{T}}^\pi_{\text{DS}} \hat{Z}^k(s,a)}(\tau) - c(s,a)$, *where*

$$c(s,a) = \left| \beta p^{-1} c_0(s,a) \pm \sigma^{p-1} \right|^{\frac{1}{p-1}} \cdot \text{sign}(c_0(s,a)).$$

For detailed proof, please refer to Appendix B.2. Briefly, it is according to the result of a simple variational skill to handle that F is a function, and setting the

derivative of Equation (4) equal to zero.

Next, we define the conservative soft distributional evaluation operator $\tilde{\mathcal{T}}^{\pi} = \mathcal{O}_c \hat{\mathcal{T}}^{\pi}_{\mathrm{DS}}$ by compositing $\hat{\mathcal{T}}^{\pi}_{\mathrm{DS}}$ and the shift operator $\mathcal{O}_c : \mathcal{Z} \to \mathcal{Z}$, which is defined by $F^{-1}_{\mathcal{O}_c Z(s,a)}(\tau) = F^{-1}_{Z(s,a)}(\tau) - c(s,a)$. Consequently, it is following that $\tilde{Z}^{k+1} = \tilde{\mathcal{T}}^{\pi} \tilde{Z}^{k}$.

Now, the main theorem we came up with shows that the conservative distributional soft evaluation obtains a conservative quantile estimate of the true quantile at all quantiles τ.

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

$$F^{-1}_{Z^{\pi}(s,a)}(\tau) \geq F^{-1}_{\tilde{Z}^{\pi}(s,a)}(\tau) + (1-\gamma)^{-1} \min_{s',a'} \left\{ c(s',a') - \frac{1}{\zeta} \sqrt{\frac{5|\mathcal{S}|}{n(s',a')} \log \frac{4|\mathcal{S}||\mathcal{A}|}{\delta}} \right\},$$

for all $s \in \mathcal{D}, a \in \mathcal{A}$, and $\tau \in [0,1]$. Furthermore, for

$\beta \geq \max_{s,a} \left\{ \frac{p\left(\Delta(s,a)^{p-1} + \sigma^{p-1}\right)}{c_0(s,a)} \right\}$, we have $F^{-1}_{Z^{\pi}(s,a)}(\tau) \geq F^{-1}_{\tilde{Z}^{\pi}(s,a)}(\tau)$.

For detailed proof, please refer to Appendix B.2. As the theorem shows, the above inequality indicates that the quantile estimates obtained by $\tilde{\mathcal{T}}^{\pi}$ are a lower bound of the true quantiles. Furthermore, we give a sufficient condition to show that the result given in Theorem 1 is not a vacuous conclusion. Therefore, Theorem 1 theoretically illustrates that ORDER does not exacerbate the distribution shift problem, and the mitigation of the distribution shift problem will be demonstrated in the experimental section.

The performance of many RL algorithms will exhibit different behaviors under different distorted expectations. Consequently, we can acquire the same conservative estimates of these objectives, which is a kind of generalization of Theorem 1.

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

In particular, $Q^{\pi}(s,a) \geq \tilde{Q}^{\pi}(s,a)$ is obtained if we take $g = \mathrm{Uniform}([0,1])$. By choosing different risk measure functions, we can apply this conclusion to any risk-sensitive offline RL.

4. Related Works

Offline RL [1] [24]-[26] learns a policy from previously collected static datasets. As a subfield of RL [27] [28], it has achieved significant accomplishments in practice [4]-[6] [29]. However, the main two challenges of offline RL are the distribution shift problem and robustness [26] [30] [31], which require various techniques to improve the stability and performance of learned policies.

4.1. Distribution Shift

The cause of the distribution shift problem is that the distribution of collected

offline training data differs from the distribution of data in practice. BCQ [9] addresses this problem through policy regularization techniques, which formulates the policy as an adaptable deviation constrained by maximum values [32]. One solution to alleviate the distribution shift problem as mentioned in BEAR [33] is incorporating a weighted behavior-cloning loss achieved by minimizing maximum mean discrepancy (MMD) into the policy improvement step. Though learning a conservative Q-function caused by distribution shift, CQL [17] solves the overestimation of value functions, which theoretically proves that a lower bound of the true value is obtained. With the introduction of distributional reinforcement learning into offline RL, CODAC [15] learns a conservative return distribution by punishing the predicted quantiles returned for the OOD actions. From another perspective, continuous quantiles are used in MQN-CQR [16] to learn the quantile of return distribution with non-crossing guarantees. ORDER builds on these approaches, but considers the entropy regularizer of quantile function instead of an unchangeable constant for ensuring sufficient exploration and may relieve the training imbalance.

4.2. Robustness Issues

Owning to the distribution shift issues, current offline RL algorithms tend to prioritize caution in their approach to value estimation and action selection. Nevertheless, this selection can compromise the robustness of learned policies, making them highly sensitive to even minor perturbations in observations. As a groundbreaking piece of work, SR^{2}L [21] achieves a more robust policy by introducing a smoothness regularizer into both the value function and the policy. A robust Q-learning algorithm proposed in REM [30] is presented that integrates multiple Q-value networks in a random convex combination of multiple Q-value estimates, ensuring that the final Q-value estimate remains robust. With arbitrarily large state spaces, RFQI [34] learns the optimal policy by employing function approximation using only an offline dataset and addresses robust offline reinforcement learning problems. In ORDER, we import the smoothing regularizer to the distribution functions and policies instead of simple action-value functions.

Our method is related to the previous offline RL algorithms based on constraining the learned value function [17] [35]. What sets our method apart is that it can better capture uncertain information about OOD actions and learn more robust policies with the introduction of the smoothing technique into distributional reinforcement learning. In addition, we enhance the network by using the entropy regularizer of quantile networks and increasing the number of network ensembles, which also improves robustness.

5. Experiments

In the sequel, we first compare ORDER against some offline RL algorithms. Then how different risk measure functions impact our proposed algorithm is investigated in Section 5.2. Besides, Section 5.3 shows the ensemble size of the quantile

network in ORDER. And our approach significantly exceeds the baseline algorithm in most tasks.

We evaluate our experiments on the D4RL benchmark [18] with various continuous control tasks and datasets. Specifically, we employ three environments (HalfCheetah, Hopper and Walker2d) and four dataset types (random, medium, medium-replay, and medium-expert). The random or medium dataset is generated by a single random or medium policy. The medium-replay dataset contains experiences collected in training a medium-level policy, and the medium-expert dataset is a mixture of medium and expert datasets.

5.1. Comparison with Offline RL Algorithms

In all the aforementioned datasets, we compare our method against several previous popular offline RL algorithms, including 1) bootstrapping error accumulation reduction (BEAR) [33], 2) conservative q-learning (CQL) [17], 3) CODAC [15], 4) robust offline reinforcement learning (RORL) [36], 5) monotonic quantile network with conservative quantile regression (MQN-CQR) [16]. The results of BEAR and CQL are directly taken from [18]. For CODAC and MQN-CQR, their results are taken from the original paper. Since the RORL paper does not report scores with five random seeds, we run the RORL using the official code base. Implementation details are given in **Appendix D**. Table A1 and Table A2 list the hyperparameters of ORDER in different datasets. Without loss of generality, we employ the neutral risk measure in this subsection.

Table 1. Normalized average returns on D4RL benchmark, averaged over five random seeds. "r", "m", "m-r" and "m-e" indicate the abbreviations of random, medium, medium-replay and medium-expert, respectively. All methods are run for 1M gradient steps.

Datasets	BEAR	CQL	CODAC	RORL (Reproduced)	MQN-CQR	ORDER
hopper-r	3.9 ± 2.3	7.9 ± 0.4	11.0 ± 0.4	**22.7 ± 8.4**	13.2 ± 0.6	**24.8 ± 7.8**
hopper-m	51.8 ± 4.0	53.0 ± 28.5	70.8 ± 11.4	**104.8 ± 0.3**	94.7 ± 13.2	101.5 ± 0.2
hopper-m-r	52.2 ± 19.3	88.7 ± 12.9	100.2 ± 1.0	102.3 ± 0.5	95.6 ± 18.5	**106.4 ± 0.1**
hopper-m-e	50.6 ± 25.3	105.6 ± 12.9	**112.0 ± 1.7**	**112.8 ± 0.2**	**113.0 ± 0.5**	**114.6 ± 3.3**
walker2d-r	12.8 ± 10.2	5.1 ± 1.3	18.7 ± 4.5	21.5 ± 0.2	22.6 ± 6.1	**28.4 ± 6.2**
walker2d-m	-0.2 ± 0.1	73.3 ± 17.7	82.0 ± 0.5	**103.2 ± 1.7**	80.0 ± 0.5	86.0 ± 0.2
walker2d-m-r	7.0 ± 7.8	81.8 ± 2.7	33.2 ± 17.6	**90.1 ± 0.6**	52.3 ± 16.7	**87.9 ± 4.8**
walker2d-m-e	22.1 ± 44.9	107.9 ± 1.6	106.0 ± 4.6	**120.3 ± 1.8**	112.1 ± 8.9	115.1 ± 1.2
halfCheetah-r	2.3 ± 0.0	17.5 ± 1.5	**34.6 ± 1.3**	28.2 ± 0.7	32.6 ± 2.9	31.5 ± 1.0
halfCheetah-m	43.0 ± 0.2	47.0 ± 0.5	46.3 ± 1.0	**64.7 ± 1.1**	45.1 ± 1.5	**63.7 ± 0.4**
halfCheetah-m-r	36.3 ± 3.1	45.5 ± 0.7	44.0 ± 0.8	**61.1 ± 0.7**	45.3 ± 7.9	57.4 ± 1.7
halfCheetah-m-e	46.0 ± 4.7	75.6 ± 25.7	70.4 ± 19.4	**108.2 ± 0.8**	71.1 ± 4.9	93.2 ± 1.1

The performance of all these algorithms is exhibited in Table 1, which reports the average normalized scores along with their corresponding standard deviations. We observe that ORDER outperforms BEAR in all tasks and surpasses the performance of the CQL algorithm. Significantly, our algorithm surpasses the performance of current distributed offline RL methods (see CODAC and MQN-CQR in Table 1), which is attributed to the assurance of robustness in ORDER. Meanwhile, ORDER competes favorably with the current state-of-the-art algorithms, owing to the safety guaranteed by distributional RL.

5.2. Policy Training under Risk Measures Function

In this subsection, we investigate how the risk measure functions affect the performance of ORDER. We compare three risk-averse learned policies [14] in distribution RL with the risk-neutral measure function. Specifically, for different $g(\tau)$, three ways of distorted expectation are considered,

- CPW: $g(\tau)=\tau^{\lambda}\Big/\left(\tau^{\lambda}+(1-\tau)^{\lambda}\right)^{1/\lambda}$, and λ is set as 0.71.
- Wang: $g(\tau)=F_{\mathcal{N}}\left(F_{\mathcal{N}}^{-1}(\tau)+\lambda\right)$, where λ is set as 0.25 and $F_{\mathcal{N}}$ is the standard Gaussian CDF.
- CVaR: $g(\tau)=\min\{\tau/\lambda,1\}$, and λ is set as 0.25.

Besides, we evaluate three risk-seeking learned policies.

- The first risk-seeking method is mean-variance and λ is set as −0.1.
- The second risk-seeking method is Var, and λ is set as 0.75.
- The third risk-seeking method is Wang, and λ is set as −0.75.

The results are shown in Table 2 and Table 3, indicating that there is little difference between risk-averse methods and risk-seeking learned policies. This suggests that risk measure functions within the ORDER framework are not highly sensitive. At this point, we also empirically demonstrate the robustness of our approach.

Table 2. Performance of ORDER under various risk-averse methods in hopper-medium-replay-v2. Each method is run with five random seeds.

Risk measure	Neutral	CPW (0.71)	CVaR (0.25)	Wang (0.75)
Performance of ORDER	106.4 ± 0.1	105.3 ± 0.4	106.2 ± 0.4	106.0 ± 1.5

Table 3. Performance of ORDER under various risk-seeking methods in hopper-medium-replay-v2. Each method is run with five random seeds.

Risk measure	Neutral	Mean-Std (−0.1)	VaR (0.75)	Wang (−0.75)
Performance of ORDER	106.4 ± 0.1	107.5 ± 0.3	106.5 ± 0.6	106.4 ± 0.2

5.3. Ablations on Benchmark Results

Without loss of generality, we choose the hopper-medium-replay-v2 dataset as an

example to conduct the ablation study in this subsection. The performance of our ORDER algorithm under different Ms is visualized in Figure 2. We observe with the increase of M, the effect has a significant improvement in both computation efficiency and stability; as shown in the yellow and purple lines. However, M should not be too large, which is presumably attributed to the overfitting problem (see the blue line, the normalized score value fluctuates significantly around the training epoch of 700). In conclusion, M is set as four to balance between robustness enhancement and computational efficiency improvement.

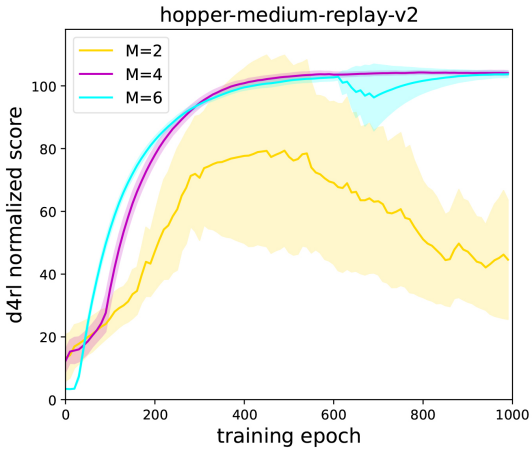

Figure 2. The normalized score under different ensemble sizes. Each method is run with five random seeds.

6. Conclusion

In this work, we introduce Offline Robustness of Distributional actor-critic Ensemble Reinforcement Learning (ORDER) to balance the conservatism and robustness in the offline setting. To achieve robustness, we first take into account the entropy regularizer that helps fully explore the dataset and alleviates training imbalance issues. Moreover, we consider the ensemble of multiple quantile networks to enhance robustness. Furthermore, a smoothing technique is introduced to the policies and the distributional functions for the perturbed states. In addition, we theoretically prove that ORDER converges to a conservative lower bound, which also shows that we improve the robustness without exacerbating the OOD problem. Finally, ORDER shows its advantage against the existing distributional offline RL methods in the D4RL benchmark. We also validate the effectiveness of ORDER through ablation studies.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Lange, S., Gabel, T. and Riedmiller, M. (2012) Batch Reinforcement Learning. In: Wiering, M. and van Otterlo, M., Eds., *Reinforcement Learning*, Springer, 45-73.

https://doi.org/10.1007/978-3-642-27645-3_2

[2] Wang, R., Wu, Y., Salakhutdinov, R. and Kakade, S. (2021) Instabilities of Offline RL with Pretrained Neural Representation. *Proceedings of the* 38*th International Conference on Machine Learning*, Virtual, 18-24 July 2021, 10948-10960.

[3] Nguyen-Tang, T., Yin, M., Gupta, S., Venkatesh, S. and Arora, R. (2023) On Instance-Dependent Bounds for Offline Reinforcement Learning with Linear Function Approximation. *Proceedings of the AAAI Conference on Artificial Intelligence*, **37**, 9310-9318. https://doi.org/10.1609/aaai.v37i8.26116

[4] Diehl, C., Sievernich, T.S., Kruger, M., Hoffmann, F. and Bertram, T. (2023) Uncertainty-Aware Model-Based Offline Reinforcement Learning for Automated Driving. *IEEE Robotics and Automation Letters*, **8**, 1167-1174. https://doi.org/10.1109/lra.2023.3236579

[5] Zhang, Z., Mei, H. and Xu, Y. (2023) Continuous-Time Decision Transformer for Healthcare Applications. *Proceedings of the* 26*th International Conference on Artificial Intelligence and Statistics*, Valencia, 25-27 April 2023, 6245-6262.

[6] Singh, B., Kumar, R. and Singh, V.P. (2022) Reinforcement Learning in Robotic Applications: A Comprehensive Survey. *Artificial Intelligence Review*, **55**, 945-990. https://doi.org/10.1007/s10462-021-09997-9

[7] Haarnoja, T., Zhou, A., Abbeel, P. and Levine, S. (2018) Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, 10-15 July 2018, 1861-1870.

[8] Ashvin, N., Murtaza, D., Abhishek, G. and Sergey, L. (2020) AWAC: Accelerating Online Reinforcement Learning with Offline Datasets. arXiv: 2006.09359. https://doi.org/10.48550/arXiv.2006.09359

[9] Fujimoto, S., Meger, D. and Precup, D. (2019) Off-Policy Deep Reinforcement Learning without Exploration. *Proceedings of the* 36*th International Conference on Machine Learning*, Long Beach, 9-15 June 2019, 2052-2062.

[10] Fujimoto, S. and Gu, S.S. (2021) A Minimalist Approach to Offline Reinforcement Learning. *Advances in Neural Information Processing Systems*, **34**, 20132-20145.

[11] Lyu, J., Ma, X., Li, X. and Lu, Z. (2022) Mildly Conservative Q-Learning for Offline Reinforcement Learning. *Advances in Neural Information Processing Systems*, **35**, 1711-1724.

[12] Dabney, W., Ostrovski, G., Silver, D. and Munos, R. (2018) Implicit Quantile Networks for Distributional Reinforcement Learning. *Proceedings of the 35th International Conference on Machine Learning*, Stockholm, 10-15 July 2018, 1096-1105.

[13] Dabney, W., Rowland, M., Bellemare, M. and Munos, R. (2018) Distributional Reinforcement Learning with Quantile Regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, **32**, 2892-2901. https://doi.org/10.1609/aaai.v32i1.11791

[14] Ma, X., Xia, L., Zhou, Z., Yang, J. and Zhao, Q. (2020) DSAC: Distributional Soft Actor Critic for Risk-Sensitive Reinforcement Learning. arXiv: 2004.14547. https://doi.org/10.48550/arXiv.2004.14547

[15] Ma, Y., Jayaraman, D. and Bastani, O. (2021) Conservative Offline Distributional Reinforcement Learning. *Advances in Neural Information Processing Systems*, **34**, 19235-19247.

[16] Bai, C., Xiao, T., Zhu, Z., Wang, L., Zhou, F., Garg, A., *et al.* (2022) Monotonic Quantile Network for Worst-Case Offline Reinforcement Learning. *IEEE Transactions on Neural Networks and Learning Systems*, **35**, 8954-8968.

https://doi.org/10.1109/TNNLS.2022.3217189

[17] Kumar, A., Zhou, A., Tucker, G. and Levine, S. (2020) Conservative Q-Learning for Offline Reinforcement Learning. *Advances in Neural Information Processing Systems*, **33**, 1179-1191.

[18] Fu, J., Kumar, A., Nachum, O., Tucker, G. and Levine, S. (2020) D4RL: Datasets for Deep Data-Driven Reinforcement Learning. arXiv: 2004.07219. https://doi.org/10.48550/arXiv.2004.07219

[19] Bellemare, M.G., Dabney, W. and Munos, R. (2017) A Distributional Perspective on Reinforcement Learning. *Proceedings of the* 34*th International Conference on Machine Learning*, Sydney, 6-11 August 2017, 449-458.

[20] Müller, A. (1997) Integral Probability Metrics and Their Generating Classes of Functions. *Advances in Applied Probability*, **29**, 429-443. https://doi.org/10.2307/1428011

[21] Shen, Q., Li, Y., Jiang, H., Wang, Z. and Zhao, T. (2020) Deep Reinforcement Learning with Robust and Smooth Policy. *Proceedings of the* 37*th International Conference on Machine Learning*, Virtual, 13-18 July 2020, 8707-8718.

[22] Koenker, R. and Hallock, K.F. (2001) Quantile Regression. *Journal of Economic Perspectives*, **15**, 143-156. https://doi.org/10.1257/jep.15.4.143

[23] Huber, P.J. (1992) Robust Estimation of a Location Parameter. In: Kotz, S. and Johnson, N.L., Eds., *Breakthroughs in Statistics*, Springer, 492-518. https://doi.org/10.1007/978-1-4612-4380-9_35

[24] Wu, Y., Tucker, G. and Nachum, O. (2019) Behavior Regularized Offline Reinforcement Learning. arXiv: 1911.11361. https://doi.org/10.48550/arXiv.1911.11361

[25] Kidambi, R., Rajeswaran, A., Netrapalli, P. and Joachims. T. (2020) MOReL: Model-Based Offline Reinforcement Learning. *Advances in Neural Information Processing Systems*, **33**, 21810-21823.

[26] Figueiredo Prudencio, R., Maximo, M.R.O.A. and Colombini, E.L. (2024) A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems. *IEEE Transactions on Neural Networks and Learning Systems*, **35**, 10237-10257. https://doi.org/10.1109/tnnls.2023.3250269

[27] Sutton, R.S. and Barto, A.G. (2018) Reinforcement Learning: An Introduction. MIT Press.

[28] Puterman, M.L. (2014) Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons.

[29] Singla, A., Rafferty, A.N., Radanovic, G. and Heffernan, N.T. (2021) Reinforcement Learning for Education: Opportunities and Challenges. arXiv:2107.08828. https://doi.org/10.48550/arXiv.2107.08828

[30] Agarwal, R., Schuurmans, D. and Norouzi, M. (2020) An Optimistic Perspective on Offline Reinforcement Learning. *Proceedings of the* 37*th International Conference on Machine Learning*, Vienna, 13-18 July 2020, 104-114.

[31] Bai, C., Wang, L., Yang, Z., Deng, Z., Garg, A., Liu, P. and Wang, Z. (2022) Pessimistic Bootstrapping for Uncertainty-Driven Offline Reinforcement Learning. arXiv: 2202.11566. https://doi.org/10.48550/arXiv.2202.11566

[32] Sohn, K., Lee, H. and Yan, X. (2015) Learning Structured Output Representation Using Deep Conditional Generative Models. *Advances in Neural Information Processing Systems*, **28**, 3483-3491.

[33] Kumar, A., Fu, J., Soh, M., Tucker, G. and Levine, S. (2019) Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. *Advances in Neural Information Processing Systems*, **32**, 11761-11771.

[34] Panaganti, K., Xu, Z., Kalathil, D. and Ghavamzadeh, M. (2022) Robust Reinforcement Learning Using Offline Data. *Advances in Neural Information Processing Systems*, **35**, 32211-32224.

[35] Kostrikov, I., Nair, A. and Levine, S. (2021) Offline Reinforcement Learning with Implicit Q-Learning. arXiv: 2110.06169. https://doi.org/10.48550/arXiv.2110.06169

[36] Yang, R., Bai, C., Ma, X., Wang, Z., Zhang, C. and Han, L. (2022) RORL: Robust Offline Reinforcement Learning via Conservative Smoothing. *Advances in Neural Information Processing Systems*, **35**, 23851-23866.

Appendix

A. Algorithm and Implementation Details

In this section, we provide a comprehensive account of our practical implementation of ORDER, offering a detailed explanation of the process.

A.1. ORDER Objective

To establish a well-defined optimization problem, we introduce a regularization term denoted as $\mathcal{R}(\mu)$ in the original objective:

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

Let $c_0(s,a) = \dfrac{\mu(a|s) - \hat{\pi}(a|s)}{\hat{\pi}(a|s)}$ and $\mathcal{R}(\mu)$ to be the entropy $\mathcal{H}(\mu)$, then $\mu(a|s) \propto \exp\big(Q(s,a)\big) = \exp\left(\int_0^1 F^{-1}_{Z(s,a)}(\tau)\mathrm{d}\tau\right)$ is the solution to the inner-maximization. Substituting this selection into the previously mentioned regularized objective function gives

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

As demonstrated in [15], we also introduce a parameter $\zeta \in \mathbb{R}_{>0}$ to threshold the quantile value difference between μ and $\hat{\pi}$, and give this difference a weight $\xi \in \mathbb{R}_{>0}$. Then we get a trainable expression of β through the process of dual gradient descent:

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

Since all our experiments take place in continuous-control domains, it is not feasible to list all possible actions as and directly compute $\log \sum_a \exp\Big(F^{-1}_{Z(s,a)}(\tau)\Big)$. In our implementation, we employ the importance sampling approximation method described in [17], and obtain

$$\log \sum_a \exp\Big(F^{-1}_{Z(s,a)}(\tau)\Big) \approx \log\left(\frac{1}{2C} \sum_{a_i \sim U(\mathcal{A})}^{N} \left[\frac{\exp\Big(F^{-1}_{Z(s,a)}(\tau)\Big)}{U(\mathcal{A})} \right] + \frac{1}{2C} \sum_{a_i \sim \pi(a|s)}^{N} \left[\frac{\exp\Big(F^{-1}_{Z(s,a)}(\tau)\Big)}{\pi(a_i|s)} \right] \right), \tag{5}$$

where $U(\mathcal{A}) = \mathrm{Uniform}(\mathcal{A})$ represents a uniform distribution of actions. Algorithm 7.1 summarizes a single step of the actor and critic updates used by ORDER.

Algorithm 1. ORDER

Hyperparameters: Number of generated quantiles N, number of quantile networks for value functions M, Huber loss threshold κ, discount rate γ, learning rates η_{actor}, η_Z, η_β, tuning parameter α, α_1, OOD penalty scale ξ, OOD penalty threshold ζ

Parameters: Critic parameters θ, Actor parameters ϕ, Penalty β

Compute distributional TD loss

Get the next action using current policy $a' \sim \pi(\cdot \mid s'; \phi)$

for $i = 1$ **to** N **do** (For the m-th quantile network)

  for $j = 1$ **to** N **do**

    $\delta^m_{\tau_i, \tau_j'} = r + \gamma F^{-1}_{Z(s',a'),\theta'_m}(\tau_j') - F^{-1}_{Z(s,a),\theta_m}(\tau_i)$

  end for

end for

Computer $\mathcal{L}_{critic}(\theta_m) = N^{-2} \sum_{i=1}^{N} \sum_{j=1}^{N} \mathcal{L}_\kappa(\delta^m_{\tau_i, \tau_j'}; \tau_i)$

Compute OOD penalty

Sample $i \sim U(\{1, \ldots, N\})$ and use quantile τ_i

Estimate $\log \sum_a \exp\left(F^{-1}_{Z(s,a),\theta_m}(\tau_i)\right)$ according to Equation (5)

Compute

$$\mathcal{L}_{OOD}(\theta_m, \beta) = \beta \cdot \left(\xi \cdot \left(\log \sum_a \exp\left(F^{-1}_{Z(s,a),\theta_m}(\tau_i)\right) - N^{-1} \sum_{j=1}^{N} F^{-1}_{Z(s,a),\theta_m}(\tau_j) \right) - \zeta \right).$$

Update quantile network

Use Equation (1) to add perturbations to the state to obtain $\hat{s}$.

Train θ using Equation (2) by SGD

Update $\theta \leftarrow \theta - \eta_Z \nabla \left(\mathcal{L}_Z(\theta_m) + \alpha \mathcal{L}_{smooth}(s, a; \theta_m) + \mathcal{L}_{OOD}(\theta_m, \beta) \right)$

Update policy network with Φ_j objective

Get new actions with re-parameterized samples $\tilde{a} \sim \pi(\cdot \mid s; \phi)$

Computer $\Phi_g(s, \tilde{a})$ using $F^{-1}_{Z(s,\tilde{a}),\theta}(\tau_i), i = 1, \cdots, N$

$\mathcal{L}_{\text{actor}}(\phi) = \log\left(\pi(\tilde{a} \mid s; \phi)\right) + \alpha_1 \max_{\hat{s}} D_J\left(\pi_\phi(\cdot \mid s) \| \pi_\phi(\cdot \mid \hat{s})\right) - \min_{j=1,\cdots,M} \Phi^j_g(s, \tilde{a})$

Update $\phi \leftarrow \phi + \eta_{actor} \nabla \mathcal{L}_{actor}(\phi)$

B. Proofs

B.1. Proof of Lemma 1

Proof. By the definition of p-Wasserstein distance, we can re-write Equation (4) as

$$\mathbb{E}_{\mathcal{D}(s,a)}\int_0^1\left|F^{-1}_{Z(s,a)}(\tau)-F^{-1}_{\hat{\mathcal{T}}^{\pi}_{\mathrm{DS}}\hat{Z}^k(s,a)}(\tau)\right|^p\mathrm{d}\tau+\beta\cdot\mathbb{E}_{U(\tau),\mathcal{D}(s,a)}\left[c_0(s,a)\cdot F^{-1}_{Z(s,a)}(\tau)\right]$$
$$+\mathbb{E}_{\mathcal{D}(s,a),\hat{s}\in\mathbb{B}_d(s,\epsilon)}\int_0^1\left|F^{-1}_{Z(\hat{s},a)}(\tau)-F^{-1}_{Z(s,a)}(\tau)\right|^p\mathrm{d}\tau$$
$$=\int_0^1\mathbb{E}_{\mathcal{D}(s,a)}\left[\left|F^{-1}_{Z(s,a)}(\tau)-F^{-1}_{\hat{\mathcal{T}}^{\pi}_{\mathrm{DS}}\hat{Z}^k(s,a)}(\tau)\right|^p+\beta\cdot c_0(s,a)\cdot F^{-1}_{Z(s,a)}(\tau)\right.$$
$$\left.+\left|F^{-1}_{Z(\hat{s},a)}(\tau)-F^{-1}_{Z(s,a)}(\tau)\right|^p\right]\mathrm{d}\tau.$$

For arbitrary smooth functions $\phi_{s,a}$ with compact support $[V_{\min},V_{\max}]$, we consider a perturbation $G^{\varepsilon}_{s,a}(\tau)=F^{-1}_{Z(s,a)}(\tau)+\varepsilon\cdot\phi_{s,a}(\tau)$ of $F^{-1}_{Z(s,a)}(\tau)$, then the above formula can be written as

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

Then the following equation is obtained considering the derivative of ε at $\varepsilon=0$:

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\int_0^1\mathbb{E}_{\mathcal{D}(s,a)}\left[\left|G^{\varepsilon}_{s,a}(\tau)-F^{-1}_{\hat{\mathcal{T}}^{\pi}_{\mathrm{DS}}\hat{Z}^k(s,a)}(\tau)\right|^p+\beta c_0(s,a)\cdot G^{\varepsilon}_{s,a}(\tau)+\left|G^{\varepsilon}_{\hat{s},a}(\tau)-G^{\varepsilon}_{s,a}(\tau)\right|^p\right]\mathrm{d}\tau\Big|_{\varepsilon=0}$$
$$=\mathbb{E}_{\mathcal{D}(s,a)}\int_0^1\left[p\left|F^{-1}_{Z(s,a)}(\tau)-F^{-1}_{\hat{\mathcal{T}}^{\pi}_{\mathrm{DS}}\hat{Z}^k(s,a)}(\tau)\right|^{p-1}\operatorname{sign}\left(F^{-1}_{Z(s,a)}(\tau)-F^{-1}_{\hat{\mathcal{T}}^{\pi}_{\mathrm{DS}}\hat{Z}^k(s,a)}(\tau)\right)\right.$$
$$\left.+\beta c_0(s,a)+p\left|F^{-1}_{Z(\hat{s},a)}(\tau)-F^{-1}_{Z(s,a)}(\tau)\right|^{p-1}\operatorname{sign}\left(F^{-1}_{Z(\hat{s},a)}(\tau)-F^{-1}_{Z(s,a)}(\tau)\right)\right]\phi_{s,a}(\tau)\mathrm{d}\tau.$$

Owning to some perturbation $G^{\varepsilon}_{s,a}$ will cause the objective value to decrease, so the above equation must be equal to 0 for $F^{-1}_{Z(s,a)}$. If $\phi(s,a)$ does not equal zero for each s,a, since $\phi(s,a)$ is arbitrary, it will also cause the above equation to be not equal to 0. Therefore, we obtain

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

for all s,a. According to the above term is zero for all $\phi(s,a)$, we have

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

According to Assumption 3, the above equation can be converted to

$$p\left|F^{-1}_{Z(s,a)}(\tau)-F^{-1}_{\hat{\mathcal{T}}^{\pi}_{\mathrm{DS}}\hat{Z}^k(s,a)}(\tau)\right|^{p-1}\operatorname{sign}\left(F^{-1}_{Z(s,a)}(\tau)-F^{-1}_{\hat{\mathcal{T}}^{\pi}_{\mathrm{DS}}\hat{Z}^k(s,a)}(\tau)\right)=\beta c_0(s,a)\pm\sigma^{p-1},$$

which holds if and only if

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

where $c(s,a)=\left|\beta p^{-1}c_0(s,a)\pm\sigma^{p-1}\right|^{\frac{1}{p-1}}\cdot\operatorname{sign}\left(c_0(s,a)\right)$. □

B.2. Proof of Theorem 1

Lemma 2. *For all* $\delta \in \mathbb{R}_{>0}$, *with probability at least* $1-\delta$, *for any* $Z \in \mathcal{Z}$ *and* $(s,a) \in \mathcal{D}$, *we have*

$$\left\| F_{\hat{\mathcal{T}}^{\pi} Z(s,a)} - F_{\mathcal{T}^{\pi} Z(s,a)} \right\|_{\infty} \leq \sqrt{\frac{5|\mathcal{S}|}{n(s,a)} \log \frac{4|\mathcal{S}||\mathcal{A}|}{\delta}}, \tag{6}$$

where $n(s,a)$ represents the number of occurrences of (s,a) in $\mathcal{D}$.

Proof. Applying the definition of distributional soft Bellman operator to the cumulative density function, we obtain that

$$\begin{aligned} & F_{\hat{\mathcal{T}}_{\mathrm{DS}}^{\pi} Z(s,a)}(x) - F_{\mathcal{T}_{\mathrm{DS}}^{\pi} Z(s,a)}(x) \\ &= \sum_{s',a'} \hat{P}(s' \mid s,a) \pi(a' \mid s') F_{\gamma\left[Z(s',a') - c \cdot \log \pi(a' \mid s')\right] + \hat{R}(s,a)}(x) \\ &\quad - \sum_{s',a'} P(s' \mid s,a) \pi(a' \mid s') F_{\gamma\left[Z(s',a') - c \cdot \log \pi(a' \mid s')\right] + R(s,a)}(x). \end{aligned}$$

Adding and subtracting $\sum_{s',a'} \hat{P}(s' \mid s,a) \pi(a' \mid s') F_{\gamma\left[Z(s',a') - c \cdot \log \pi(a' \mid s')\right] + R(s,a)}(x)$ from this expression gives

$$\begin{aligned} & \sum_{s',a'} \hat{P}(s' \mid s,a) \pi(a' \mid s') \left(F_{\gamma\left[Z(s',a') - c \cdot \log \pi(a' \mid s')\right] + \hat{R}(s,a)}(x) - F_{\gamma\left[Z(s',a') - c \cdot \log \pi(a' \mid s')\right] + R(s,a)}(x) \right) \\ & + \sum_{s',a'} \left(\hat{P}(s' \mid s,a) - P(s' \mid s,a) \right) \pi(a' \mid s') F_{\gamma\left[Z(s',a') - c \cdot \log \pi(a' \mid s')\right] + R(s,a)}(x). \end{aligned}$$

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

$$\begin{aligned} & F_{\gamma\left[Z(s',a') - c \cdot \log \pi(a' \mid s')\right] + \hat{R}(s,a)}(x) - F_{\gamma\left[Z(s',a') - c \cdot \log \pi(a' \mid s')\right] + R(s,a)}(x) \\ &= \int \left[F_{\hat{R}(s,a)}(r) - F_{R(s,a)}(r) \right] \mathrm{d} F_{\gamma\left[Z(s',a') - c \cdot \log \pi(a' \mid s')\right]}(x - r) \\ &\leq \int \left| F_{\hat{R}(s,a)}(r) - F_{R(s,a)}(r) \right| \mathrm{d} F_{\gamma\left[Z(s',a') - c \cdot \log \pi(a' \mid s')\right]}(x - r) \\ &\leq \sup_{r} \left| F_{\hat{R}(s,a)}(r) - F_{R(s,a)}(r) \right| \int \mathrm{d} F_{\gamma\left[Z(s',a') - c \cdot \log \pi(a' \mid s')\right]}(x - r) \\ &= \left\| F_{\hat{R}(s,a)}(r) - F_{R(s,a)}(r) \right\|_{\infty}. \end{aligned}$$

Therefore,

$$\begin{aligned} & \sum_{s',a'} \hat{P}(s' \mid s,a) \pi(a' \mid s') \left(F_{\gamma\left[Z(s',a') - c \cdot \log \pi(a' \mid s')\right] + \hat{R}(s,a)}(x) - F_{\gamma\left[Z(s',a') - c \cdot \log \pi(a' \mid s')\right] + R(s,a)}(x) \right) \\ & \leq \sum_{s',a'} \hat{P}(s' \mid s,a) \pi(a' \mid s') \left\| F_{\hat{R}(s,a)}(r) - F_{R(s,a)}(r) \right\|_{\infty} \\ & = \left\| F_{\hat{R}(s,a)}(r) - F_{R(s,a)}(r) \right\|_{\infty}. \end{aligned}$$

The following derivation process is similar to CODAC [15], so we can finally obtain the above conclusion.

It has been proved in CODAC that if $\|F - G\|_{\infty} \leq \epsilon$, then $\left\|F^{-1} - G^{-1}\right\|_{\infty} \leq \epsilon / \zeta$, where F and G are two cumulative distribution functions (CDFs) with support χ, and F is ζ-strongly monotone. Thus, according to Lemma 2, we have

Lemma 3. *For any return distributional* 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*

$$\left\| F^{-1}_{\hat{\mathcal{T}}^{\pi}_{\mathrm{DS}} Z(s,a)} - F^{-1}_{\mathcal{T}^{\pi} Z(s,a)} \right\|_{\infty} \leq \frac{1}{\zeta} \sqrt{\frac{5|\mathcal{S}|}{n(s,a)} \log \frac{4|\mathcal{S}||\mathcal{A}|}{\delta}}.$$

Let $\Delta(s,a) = \frac{1}{\zeta} \sqrt{\frac{5|\mathcal{S}|}{n(s,a)} \log \frac{4|\mathcal{S}||\mathcal{A}|}{\delta}}$ and followed by Lemma 1

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

the second step holds by Lemma 2 with probability at least $1-\delta$. For any $h \in \mathbb{R}$, if Z satisfies $F^{-1}_{Z(s,a)}(\tau) \geq F^{-1}_{\mathcal{T}^{\pi} Z(s,a)}(\tau) + h$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, then $F^{-1}_{Z(s,a)}(\tau) \geq F^{-1}_{\mathcal{T}^{\pi} Z(s,a)}(\tau) + (1-\gamma)^{-1} h \ (\forall \tau \in [0,1])$. Then,

$$\begin{aligned} F^{-1}_{Z^{\pi}(s,a)}(\tau) &\geq F^{-1}_{\tilde{\mathcal{T}}^{\pi} Z^{\pi}(s,a)}(\tau) + c(s,a) - \Delta(s,a) \\ &\geq F^{-1}_{\tilde{\mathcal{T}}^{\pi} Z^{\pi}(s,a)}(\tau) + \min_{s,a} \{ c(s,a) - \Delta(s,a) \} \\ &\geq F^{-1}_{\tilde{Z}^{\pi}(s,a)}(\tau) + (1-\gamma)^{-1} \min_{s,a} \{ c(s,a) - \Delta(s,a) \}. \end{aligned} \tag{7}$$

Notice that since for the last term in Equation (7) to be positive, we need

$$\beta p^{-1} c_0(s,a) + \sigma^{p-1} \geq \Delta(s,a)^{p-1} \quad (\forall s,a).$$

Owning to we have assumed that $c_0(s,a) > 0$, then it can be equivalent to

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

we thus prove the conclusion of Theorem 1.

C. Experimental Settings

Our experimental procedure largely adheres to [18], and the results of non-distributional methods are directly taken from [18]. For all experiments, we run algorithms for 1000 epochs (1000 training steps each epoch, *i.e.*, 1M gradient steps in total). Then we evaluate them using 10 test episodes in the original environment, which all last 1000 steps long. All benchmark results are averaged over 5 random seeds. The reported results are normalized to D4RL scores that measure how the performance compared with expert score and random score:

$$\text{normalized score} = 100 \times \frac{\text{score} - \text{random score}}{\text{expert score} - \text{random score}}.$$

D. Implementation Details

We implement ORDER based on DSAC and keep the DSAC-specific hyperparameters the same. These hyperparameters are detailed in **Table A1**. As with CODAC, we introduce hyperparameters β, ζ, ξ (See Appendix A.1). In most cases,

β is a learnable parameter initialized to 1 with learning rate $\eta_\beta = 3\times10^{-4}$. In a few cases, setting $\beta = 1$ throughout the entirety of training, which we indicate by setting $\zeta = -1$.

Table A1. ORDER backbone hyperparameters.

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

Table A2. Hyperparameters of ORDER for the benchmark results.

ss	ξ	ζ	η_{critic}	entropy tuning	α	α_1
hopper-random	1	10	3e−5	yes	0.0001	0.1
hopper-medium	10	10	3e−4	yes	0.0	0.0
hopper-med-rep	1	10	3e−5	yes	0.0	0.0
hopper-med-exp	10	10	3e−5	no	0.0001	0.1
walker2d-random	1	10	3e−5	yes	0.0001	1.0
walker2d-medium	10	10	3e−5	no	0.0001	1.0
walker2d-med-rep	1	10	3e−5	yes	0.0	0.0
walker2d-med-exp	10	10	3e−5	no	0.0001	1.0
halfCheetah-random	1	10	3e−5	yes	0.0001	0.1
halfCheetah-medium	10	10	3e−5	no	0.0001	0.1
halfCheetah-med-rep	1	10	3e−5	yes	0.0001	0.1
halfCheetah-med-exp	0.1	−1	3e−4	no	0.0	0.0

Since we introduce smoothing techniques to the policy and quantile networks, we also add other hyperparameters. In Equation (2), the weight α for the quan-

tile network smoothing loss $\mathcal{L}_{\text{smooth}}$ is searched in $\{0.0, 0.0001\}$. And beyond that, the weight α_1 of the policy smoothing loss in Equation (3) is searched in $\{0.0, 0.1, 1.0\}$. When training the policy and distribution action-value functions, we randomly sample $n = 10$ perturbed observations from a ℓ_∞ ball of norm ϵ and select the one to maximize $D_J\left(\pi_\theta(\cdot|s) \| \pi_\theta(\cdot|\hat{s})\right)$ and $\mathcal{L}_{\text{smooth}}$, respectively. For Z smoothing loss in Equation (1), set parameter ϱ to 0.2 for conservative value estimation. All the hyperparameters used in ORDER for the benchmark are listed in Table A2.