Journal of Artificial Intelligence Research 81 (2024) 481-509 Submitted 05/2024; published 11/2024

Towards Robust Offline-to-Online Reinforcement Learning via Uncertainty and Smoothness

Xiaoyu Wen WENXIAOYU@MAIL.NWPU.EDU.CN
Northwestern Polytechnical University
Xi'an, Shaanxi, China

Xudong Yu HIT20BYU@GMAIL.COM
Harbin Institute of Technology
Harbin, Heilongjiang, China

Rui Yang RYANGAM@CONNECT.UST.HK
The Hong Kong University of Science and Technology
HongKong, China

Haoyuan Chen CHEN_HY@MAIL.NWPU.EDU.CN
Northwestern Polytechnical University
Xi'an, Shaanxi, China

Chenjia Bai BAICHENJIA@PJLAB.ORG.CN
(Corresponding author)
Shanghai Artificial Intelligence Laboratory
Shanghai, China
Shenzhen Research Institute of Northwestern Polytechnical University
Shenzhen, GuangDong, China

Zhen Wang W-ZHEN@NWPU.EDU.CN
(Corresponding author)
Northwestern Polytechnical University
Xi'an, Shaanxi, China

Abstract

To obtain a near-optimal policy with fewer interactions in Reinforcement Learning (RL), a promising approach involves the combination of offline RL, which enhances sample efficiency by leveraging offline datasets, and online RL, which explores informative transitions by interacting with the environment. Offline-to-Online RL provides a paradigm for improving an offline-trained agent within limited online interactions. However, due to the significant distribution shift between online experiences and offline data, most offline RL algorithms suffer from performance drops and fail to achieve stable policy improvement in offline-to-online adaptation. To address this problem, we propose the Robust Offline-to-Online (RO2O) algorithm, designed to enhance offline policies through uncertainty and smoothness, and to mitigate the performance drop in online adaptation. Specifically, RO2O incorporates Q-ensemble for uncertainty penalty and adversarial samples for policy and value smoothness, which enable RO2O to maintain a consistent learning procedure in online adaptation without requiring special changes to the learning objective. Theoretical analyses in linear MDPs demonstrate that the uncertainty and smoothness lead to tighter optimality bound in offline-to-online against distribution shift. Experimental results illustrate the superiority of RO2O in facilitating stable offline-to-online learning and achieving significant improvement with limited online interactions.

1. Introduction

Reinforcement learning (RL) has demonstrated remarkable success in tackling complex tasks, such as playing games (Hessel et al., 2018; Silver et al., 2018; Berner et al., 2019) and controlling robots (Schulman et al., 2015, 2017; Haarnoja et al., 2018) in recent years. Nonetheless, persistent critiques point to its limited adaptability in real-world scenarios. The efficacy of RL critically hinges upon access to an unbiased interactive environment and millions of unrestricted trial-and-error attempts (Mnih et al., 2015). However, domains such as healthcare (Yu et al., 2021) and autonomous driving (Kiran et al., 2021) often present challenges in online data collection due to safety, feasibility, and financial reasons.

Offline RL presents a distinctive advantage over online RL, as it enables the learning of policies directly from a fixed dataset collected by a behavior policy (Lange et al., 2012; Fujimoto et al., 2019; Wu et al., 2019). These datasets can be sourced from historical logs, demonstrations, or expert knowledge, furnishing valuable information to facilitate learning without the need for costly online data collection. However, the performance of current offline RL methods heavily relies on the coverage of the state-action space and the quality of stored trajectories (Schweighofer et al., 2022). Furthermore, the lack of exploration hampers the agent's ability to discover superior policies (Lambert et al., 2022). To address this issue, numerous studies focus on enhancing pre-trained offline agents through limited online interactions, known as Offline-to-Online RL (Nair et al., 2020; Lee et al., 2022; Kostrikov et al., 2022). This paradigm aims to rectify estimation bias, which remains unaddressed during offline training, and leads to further policy improvement through several online fine-tuning steps.

Despite the potential to integrate offline datasets and online experiences to optimize the agent, existing offline-to-online learning methods suffer from performance drops and struggle to efficiently improve policies, which hinders their applicability in real-world scenarios. At the initial stage of online fine-tuning, the agent's performance may heavily decline due to the distributional shift between offline datasets and online transitions (Nair et al., 2020; Uchendu et al., 2023). Moreover, the inclusion of low-quality data can have detrimental effects on performance and lead to skewed optimization. Prior efforts to address this issue involve altering the policy extraction procedure (Nair et al., 2020; Kostrikov et al., 2022), incorporating behavior cloning regularization (Zhao et al., 2022), modifying data sampling methods (Lee et al., 2022; Swazinna et al., 2021), or proposing policy expansion sets (Zhang et al., 2023). While these methods have made progress in mitigating performance drops, they still suffer form limited performance improvement due to the lack of effective mechanisms to enhance performance during the fine-tuning phase.

In this paper, we propose the Robust Offline-to-Online (RO2O) algorithm for RL, designed to address the distribution shift in the offline-to-online process and achieve efficient policy improvement during the fine-tuning phase. To achieve this, RO2O utilizes Q-ensembles to learn robust value functions, resulting in no performance drop during the initial stage of online fine-tuning. Additionally, RO2O incorporates the smoothness regularization of policies and value functions on out-of-distribution (OOD) states and actions ensuring robust performance even when the interacting trajectories in the training buffer deviate significantly from offline policies. Notably, RO2O offers the advantage of not requiring the transformation of the learning algorithm (Zhao et al., 2023) or policy composition

(Zhang et al., 2023) throughout the process. From a theoretical perspective, we prove that under the linear MDP assumption, the uncertainty and smoothness lead to a tighter optimality bound in offline-to-online against distribution shift. Empirical results showcase the favorable performance of RO2O during both offline pre-training and online fine-tuning. Compared to baseline algorithms, RO2O achieves efficient policy improvement without the need for specific explorations or modifications to the learning architecture. The code is available in this repository (https://github.com/BattleWen/RO2O).

2. Related Work

Offline-to-Online RL A key challenge in offline-to-online process is the performance drop experienced at the initial stage, attributed to the distributional shift between offline data and online experiences. Previous approaches have attempted to address this issue by altering policy extraction (Nair et al., 2020; Kostrikov et al., 2022), adjusting sampling methods (Lee et al., 2022), expanding policy sets (Zhang et al., 2023), and modifying Q-function learning targets (Nakamoto et al., 2023). However, these methods cannot consistently achieve effective policy improvement within the limited fine-tuning steps. Recently, ensembles have been incorporated for both pessimistic learning during offline training and optimistic exploration during online learning (Zhao et al., 2023). While such an ensemble method improves the Offline-to-Online performance, it requires careful modifications of learning objectives when transferring the policy from offline to online. In contrast, our work handles offline training and online fine-tuning in a consistent manner without algorithmic modifications. The proposed approach not only achieves better offline performance but also enables efficient policy improvement during online fine-tuning.

Ensembles in RL Ensemble methods in RL have emerged as a powerful approach to improve the stability and performance of learning algorithms. In online RL, ensembles are utilized to capture epistemic uncertainty and improve exploration (Osband et al., 2016; Chen et al., 2017). Recent methods also employ ensembles to mitigate estimation bias during Bellman updates (Fujimoto et al., 2018; Lan et al., 2020) or enhance sample efficiency (Chen et al., 2021). In the context of offline RL, ensembles are employed in both model-free methods (Bai et al., 2022; An et al., 2021) and model-based methods (Yu et al., 2020; Swazinna et al., 2021) to characterize the uncertainty of Q-values or dynamics models. Notably, several works (An et al., 2021; Ghasemipour et al., 2022) estimate lower confidence bounds of Q-functions using ensembles, where EDAC (An et al., 2021) primarily focuses on improving sample efficiency with gradient diversity and MSG (Ghasemipour et al., 2022) mainly emphasizes the importance of ensemble independence for effectively estimating uncertainty. Our approach extends upon these methodologies by incorporating a perturbed sample set, which differs from solely estimating uncertainty within the existing state-action space of the dataset. We primarily use ensembles to penalize the Q-values of the OOD samples and apply smooth regularization to ensure that the policies and Q-values of in-sample data and perturbed samples do not deviate too much. In this way, we can smooth them out within a small range beyond the dataset's state space, resulting in more robust estimates.

Robustness in RL Robustness has gained paramount importance in RL to ensure the reliability and stability of RL agents in diverse and challenging environments. In online

RL, previous research has explored techniques such as domain randomization (Tobin et al., 2017), policy smoothing (Shen et al., 2020), and data augmentation methods (Sinha et al., 2022) to improve performance. Recently, an offline RL algorithm (Yang et al., 2022) incorporates policy and value smoothing for OOD states, highlighting the significance of robustness in offline RL agents. These approaches typically focus on enhancing robustness against adversarial perturbations on observations or actions and validate their effectiveness through the synthesis of noisy data. In contrast, our focus is on the robustness of models to handle the distributional shift specifically in the Offline-to-Online RL setting.

3. Preliminaries

Offline-to-Online RL The RL problem is typically formulated as Markov Decision Process (MDP), represented by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$. In this framework, the agent's decision-making process is guided by a policy denoted as π, which maps environmental states $s \in \mathcal{S}$ to actions $a \in \mathcal{A}$. The agent's objective is to find an optimal policy, denoted as π^*, that maximizes the expected cumulative reward over time. For a policy π, the state-action value function, denoted as $Q^\pi(s, a)$, represents the expected cumulative reward starting from state s, taking action a, and following policy π thereafter. The learning target for the value function in online RL, also referred to as the Bellman operator, can be expressed as:

$$\mathcal{T}Q(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a), a' \in \pi(\cdot|s')} Q(s', a').$$

In offline RL, learning is performed using a fixed dataset $\mathcal{D} = \{s_i, a_i, r_i, s'_i\}_{i=1}^n$ of historical interactions sampled from a behavior policy μ. A key challenge in offline RL is the bootstrapped error caused by the distributional shift between behavior policies and learned policies. To mitigate the distributional shift, previous methods (Schneegass et al., 2008; Bai et al., 2022; Yang et al., 2022) leverage Q-ensembles to capture epistemic uncertainty and penalize Q-values with large uncertainties. When we estimate the empirical expectation from the dataset $\mathcal{D}$, the Bellman operator becomes $\widehat{\mathcal{T}}Q(s, a) = r(s, a) + \gamma \widehat{\mathbb{E}}_{s' \sim P(\cdot|s,a), a' \sim \pi(\cdot|s')}(Q(s', a') - \alpha U(s', a'))$, where $U(s', a')$ denotes the estimated uncertainties, and α is used to adjust the degree of pessimism. Additionally, RORL (Yang et al., 2022) employs smooth regularization on the policy and the value function for states near the dataset.

Despite the advantage of leveraging large-scale offline data, the performance of pre-trained agents is often limited by the optimality and coverage of the datasets. Overestimation of value functions cannot be substantially corrected without interactions with the environment. To address this limitation, our work focuses on offline-to-online learning, aiming to improve agents by incorporating limited online interactions.

Linear MDPs Our theoretical derivations build on top of linear MDP assumptions. Least Squares Value Iteration (LSVI) (Jin et al., 2020) is a classic method frequently used in the linear MDPs to calculate the closed-form solution. In linear MDPs (Jin et al., 2020), the transition dynamics and reward function take the following form, as

$$\mathbb{P}_t(s_{t+1} \,|\, s_t, a_t) = \langle \varphi(s_{t+1}), \phi(s_t, a_t) \rangle, \quad r(s_t, a_t) = \upsilon^\top \phi(s_t, a_t), \quad \forall (s_{t+1}, a_t, s_t) \in \mathcal{S} \times \mathcal{A} \times \mathcal{S},$$

where the feature embedding $\phi : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}^d$ is known and φ is an unknown measures over $\mathcal{S}$. We further assume that the reward function $r : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$ is bounded, the feature

is bounded by $\|\phi\|_2 \leq 1$ and v is an unknown vector. We consider the settings of $\gamma = 1$ in the following. Then for any policy π, the state-action value function is also linear to ϕ, as

$$Q^\pi(s_t, a_t) = w^\top \phi(s_t, a_t).$$

Given data $\mathcal{D}_m = \{s_t^i, a_t^i, r_t^i, s_{t+1}^i\}_{i\in[m]}$, the parameter of the w can be solved via LSVI algorithm, as

$$\widehat{w}_t = \min_{w\in\mathbb{R}^d} \sum_{i=1}^{m} \big(\phi(s_t^i, a_t^i)^\top w - r(s_t^i, a_t^i) - V_{t+1}(s_{t+1}^i)\big)^2 \tag{1}$$

where V_{t+1} is the estimated value function in the $(t+1)$-th step. Following LSVI, the explicit solution to Equation (1) takes the form of

$$\widehat{w}_t = \Lambda_t^{-1} \sum_{i=1}^{m} \phi(s_t^i, a_t^i) y_t^i, \qquad \text{where } \Lambda_t = \sum_{i=1}^{m} \phi(s_t^i, a_t^i)\phi(s_t^i, a_t^i)^\top$$

is the feature covariance matrix of the state-action pairs in the offline dataset, and $y_t^i = r(s_t^i, a_t^i) + V_{t+1}(s_{t+1}^i)$ is the Bellman target in regression.

4. Methodology

In this section, we present our methodology for addressing the challenges posed by the Offline-to-Online RL setting. The most significant challenge in this context is effectively transferring knowledge from the static dataset to cope with distributional shift in the dynamic online environment. To tackle this crucial issue, we propose the RO2O algorithm, a novel approach that combines Q-ensembles and robustness regularization. We begin by providing a motivating example to illustrate that current methods struggle to handle a large distributional shift effectively. Subsequently, we introduce our algorithm, which maintains a consistent architecture in both offline and online learning phases. Furthermore, we establish theoretical support for our approach.

4.1 Motivating Example

Offline-to-Online RL methods face challenges arising from distribution shift not only between learned policies and behavior policies but also between offline data and online transitions during the fine-tuning process. Robust performance is expected from offline algorithms despite the presence of online trajectories that deviate from the learned offline policies. To investigate this, we evaluate two state-of-the-art offline methods, i.e., CQL and IQL, with distinct distribution shifts to simulate the offline-to-online process. Specifically, we pre-train the agents using the halfcheetah-expert dataset from D4RL (Fu et al., 2020) benchmark, and inject synthetic distributional shift similar to that in online fine-tuning process to assess their robustness. The synthetic distributional shift is incorporated by adding samples from a different offline dataset, such as the halfcheetah-medium dataset. As depicted in the left panel of Figure 1, noticeable discrepancies in the trajectory distribution exist between the two datasets, indicating the presence of distributional shift.

We compare the performance of CQL, IQL, and our method to figure out whether the state-of-the-art methods can handle the synthetic distributional shift during fine-tuning.

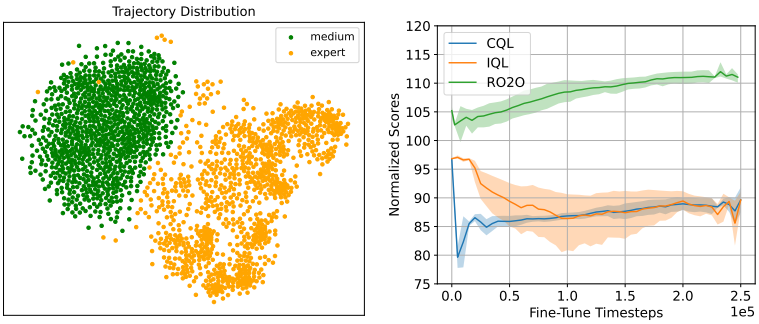

Figure 1: Illustration for the motivating example. In the left panel, we visualize the trajectory distribution of two datasets, by mapping the trajectories into two-dimensional points using T-SNE (Van der Maaten & Hinton, 2008). The right panel presents the fine-tuning performance.

The experimental results are shown in the right panel of Figure 1. Our findings reveal that all methods experience a performance drop at the initial stage due to the significant distributional shift. However, in comparison to CQL and IQL, our method exhibits a milder degradation in performance. Moreover, CQL and IQL fail to recover from the deviation during the fine-tuning phase with a new dataset. This inability is attributed to the presence of samples that deviate significantly from the region covered by the current policies, which affects the learning of policies and value functions. In contrast, our method demonstrates superior robustness, enabling effective policy improvement even in the presence of significant distributional shifts. As previously mentioned, it is expected to correct estimation bias and improve pre-trained policies within limited online interactions, while traditional methods struggle to accomplish this.

4.2 Algorithm

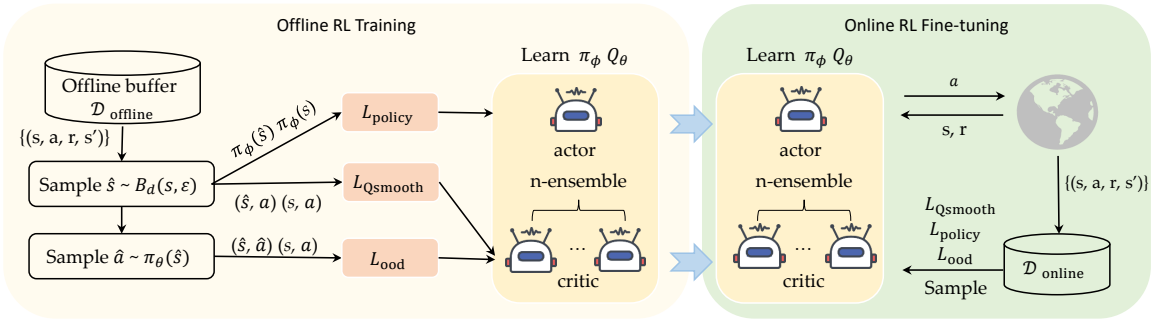

Figure 2: **Overall framework of RO2O.** RO2O employs the same off-policy RL algorithms during the offline-to-online training phase. By using OOD sampling, we incorporate $\mathcal{L}_{\text{ood}}$ and $\mathcal{L}_{\text{Qsmooth}}$ into the training process for the gradient update, while also calculating $\mathcal{L}_{\text{policy}}$ to constrain the policy $\pi_\zeta(\hat{s})$ as close as possible to the current policy $\pi_\zeta(s)$.

Based on the motivating example, we suggest that it is important to design a robust algorithm capable of ensuring stable policy improvement with online interactions. To this end, we propose the RO2O method, which incorporates ensembles and robustness regularization into the offline-to-online learning process. Notably, our method stands out from other existing approaches, such as PEX (Zhang et al., 2023) and E2O (Zhao et al., 2023), due to its unique characteristics. Unlike these methods, our approach does not require any changes to the learning algorithm or the need to conduct policy expansion when transitioning to the fine-tuning phase.

4.2.1 Ensemble-based Learning

In our method, we adopt Q-ensemble with N networks in both offline pre-training and online fine-tuning, employing the same update procedure. These ensemble networks possess identical architecture and are initialized independently. While prior studies (Chen et al., 2021; Zhao et al., 2023) suggest that randomly selecting two of the N ensembles is effective during the online learning phase, our findings indicate that choosing the *minimum* of ensemble Q-functions is sufficient to achieve favorable performance. Moreover, this choice remains consistent with the offline learning process, where pessimism is necessary to counteract overestimation bias. Formally, the TD target when using the minimum of ensemble Q-functions can be expressed as:

$$\widehat{\mathcal{T}}Q_{\theta_i}(s,a) = r(s,a) + \gamma\widehat{\mathbb{E}}_{s'} \min_i Q_{\theta_i^-}(s',a'), i \in [1,N], \tag{2}$$

where θ_i and θ_i^- are parameters for i-th Q-network and target Q-network, respectively, and $a' \sim \pi(s')$. Additionally, we notice that the results reported in (Ghasemipour et al., 2022; An et al., 2021) demonstrate that using shared targets has a prior performance in Mujoco tasks but fails in more challenging domains such as AntMaze. Because shared targets on AntMaze tend to be overly pessimistic, it's harder to explore new out-of-distribution samples, making it more difficult to learn better policies. Thus, for the challenging AntMaze environments, we adhere to previous work and utilize *independent* Bellman targets (refer to Appendix D for more details about the theoretical evaluation) without altering the network architecture. Similarly, independent targets can be formulated as:

$$\widehat{\mathcal{T}}Q_{\theta_i}(s,a) = r(s,a) + \gamma\widehat{\mathbb{E}}_{s'} Q_{\theta_i^-}(s',a'), i \in [1,N]. \tag{3}$$

During offline training, the Q-ensembles are utilized to learn the value function and update the policy. In online fine-tuning, the learned Q-ensembles and policies continue to be updated with online experiences, as shown in Figure 2. Several online RL methods (Chen et al., 2021; Lee et al., 2021) also employ Q-ensembles and suggest maximizing the average Q-values for policy optimization. In our study, we have found that maximizing the minimum Q-values, consistent with the objective in the offline phase, is also highly effective in obtaining the optimal policy during online fine-tuning.

4.2.2 Robustness Regularization

Similar to previous studies (Sinha et al., 2022; Shen et al., 2020) that consider robustness in RL, our goal is to enhance the robustness of offline-trained value function and policy,

which can have large estimation bias caused by the distribution shift in the fine-tuning phase. Different from previous offline RL approaches that mainly mitigate the effect of perturbed actions (Zhao et al., 2022; Nakamoto et al., 2023), we adopt a different robustness perspective by suggesting that the distributional shift in offline-to-online process brings both OOD states and actions.

Intuitively, the offline pre-trained policy inevitably encounters OOD samples during the offline-to-online process. Since the process of online exploration is based on the rollout of the offline policy, these OOD samples tend to be distributed around the offline data. In order to deal with the problem of distribution shift, we attempt to introduce additional adversarial samples for smoothness and uncertainty estimation. In this way, we can guarantee that the policies and Q-values of in-sample data and perturbed samples do not deviate too much and smooth out them within a small range beyond the dataset's state space, thereby leading to smooth value function and policy that are robust to distribution shift. To this end, we follow a similar approach as in RORL (Yang et al., 2022) to construct adversarial samples for regularization in the offline-to-online process.

Smoothness Regularization We employ regularization on both policy and Q-function by minimizing the difference between estimations obtained from in-sample data and perturbed samples. Specifically, we synthesize perturbed samples by constructing a perturbation set $\mathbb{B}_d(s,\epsilon) = \{\hat{s} : d(s,\hat{s}) \leq \epsilon\}$ for state s, which is an ϵ-radius ball with a distance metric $d(\cdot,\cdot)$. By sampling from this set $\hat{s} \in \mathbb{B}_d(s,\epsilon)$, the proposed RO2O minimizes the difference between $Q_{\theta_i}(s,a)$ and $Q_{\theta_i}(\hat{s},a)$, as

$$\mathcal{L}^i_{\text{Qsmooth}} = \mathcal{L}\left(Q_{\theta_i}(s,a), Q_{\theta_i}(\hat{s},a)\right),$$

which enforces value smoothness to adversarial state $\hat{s}$, and $\mathcal{L}$ can be a L_2 distance. The smooth loss should be applied to each network in the ensemble. To simplify the optimization, we choose to minimize the maximal smooth loss $\max_i \mathcal{L}^i_{\text{Qsmooth}}$ among the ensemble. Similarly, we can minimize the difference between $\pi(a|s)$ and $\pi(a|\hat{s})$, which is realized by minimizing the Jensen-Shannon divergence $D_{\text{JS}}(\pi(\cdot|s)\|\pi(\cdot|\hat{s}))$. The JS divergence $D_{\text{JS}}(\pi(\cdot|s)|\pi(\cdot|\hat{s}))$ is defined as: $D_{\text{JS}}(\pi(\cdot|s)||\pi(\cdot|\hat{s})) = \frac{1}{2}D_{\text{KL}}(\pi(\cdot|s)||M) + \frac{1}{2}D_{\text{KL}}(\pi(\cdot|\hat{s})||M)$, where M is the mixture distribution of $\pi(\cdot|s)$ and $\pi(\cdot|\hat{s})$, given by $M = \frac{1}{2}(\pi(\cdot|\hat{s}) + \pi(\cdot|s))$.

Overestimation Penalty Meanwhile, since the Q-values for OOD states and actions can be overestimated, we penalize their value estimation with uncertainty quantification following prior works (Bai et al., 2022; Yang et al., 2022). For OOD states $\hat{s} \in \mathbb{B}_d(s,\epsilon)$ and OOD actions $\hat{a} \sim \pi(\hat{s})$, their pseudo Bellman targets can be expressed as $\widehat{\mathcal{T}}^{\text{ood}}Q(\hat{s},\hat{a}) = Q(\hat{s},\hat{a}) - \alpha U(\hat{s},\hat{a})$. Here, θ_i denotes the parameters of the i-th Q-function. We define a loss function to constrain the value of OOD samples as:

$$\mathcal{L}^i_{\text{ood}} = \mathbb{E}_{\hat{s}\in\mathbb{B}_d(s,\epsilon),\hat{a}\sim\pi(\hat{s})}(\widehat{\mathcal{T}}^{\text{ood}}Q_{\theta_i}(\hat{s},\hat{a}) - Q_{\theta_i}(\hat{s},\hat{a}))^2.$$

Specifically, we define the uncertainty function $U(\hat{s},\hat{a})$ as follows:

$$U(\hat{s},\hat{a}) = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(Q_{\theta_i}(\hat{s},\hat{a}) - \bar{Q}_{\theta_i}(\hat{s},\hat{a}))^2},$$

where K is the number of the ensemble networks and $\bar{Q}_{\theta_i}(\hat{s},\hat{a})$ means the mean Q-value of the ensemble networks.

Algorithm 1 Robust Offline-to-Online RL algorithm

Require: ensemble Q-networks $\{Q_{\theta_i}\}_{i=1}^n$, target networks $\{Q_{\theta_i^-}\}_{i=1}^n$, and policy network π_ϕ

// *Offline Pre-training*
while $t \leq T_1$ **do**
  Sample mini batches from $\mathcal{D}$.
  Calculate robustness regularization $\mathcal{L}^i_{\text{Qsmooth}}, \mathcal{L}^i_{\text{ood}}$.
  Update Q-functions with Equation (4) and update target networks softly.
  Update the policy with Equation (5).
end while
// *Online Fine-tuning*
while $t \leq T_2$ **do**
  Interact with the online environment with π_ϕ.
  Collect transitions into new buffer $\mathcal{B}$.
  Sample batches from buffer $\mathcal{B}$.
  Update Q-functions and the policy with $\mathcal{L}^i_Q, \mathcal{L}_\pi$.
end while

4.2.3 Algorithm Description

As outlined in Algorithm 1, the learning process encompasses two phases: offline pre-training and online fine-tuning. We adopt the SAC (Haarnoja et al., 2018) algorithm as our backbone. For Q-value functions, RO2O has the following loss function:

$$\mathcal{L}^i_Q = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\mathcal{L}^i_{\text{TD}} + \eta_1 \mathcal{L}^i_{\text{Qsmooth}} + \eta_2 \mathcal{L}^i_{\text{ood}}\right], \tag{4}$$

where $\mathcal{D} = \mathcal{D}_{\text{offline}}$ in the offline training phase and $\mathcal{D} = \mathcal{D}_{\text{offline}} \cup \mathcal{D}_{\text{online}}$ in the online fine-tuning phase. $\mathcal{L}^i_{\text{TD}} = (\mathcal{T}Q_{\theta_i}(s,a) - Q_{\theta_i}(s,a))^2$ represents the TD error, where $\mathcal{T}Q_{\theta_i} = r + \gamma\left(\min Q_{\theta_i^-}(s',a') - \beta \log \pi(a'|s')\right)$ when taking shared targets in Equation (2) for Mujoco tasks, and $\mathcal{T}Q_{\theta_i} = r + \gamma\left(Q_{\theta_i^-}(s',a') - \beta \log \pi(a'|s')\right)$ when using independent targets in Equation (3) for AntMaze tasks. The policy is learned by optimizing the following loss function:

$$\mathcal{L}_\pi = \mathbb{E}_{(s,a)\sim\mathcal{D}}\big[\min_i Q_{\theta_i}(s,a) + \beta \log \pi_\zeta(a|s) + \eta_3 D_{\text{JS}}(\pi_\zeta(\cdot|s) \| \pi_\zeta(\cdot|\hat{s}))\big], \tag{5}$$

where ϕ represents the parameters of the policy network. In Equation (5), the first term maximizes the minimum of the ensemble Q-functions to obtain a conservative policy, the second term is the entropy regularization, and the third term is the smooth constraint. We remark that we maintain the same loss function throughout the offline-to-online process, which is more elegant than previous offline-to-online methods. The difference between the pre-training and the fine-tuning phase lies in the data sampled to estimate of the expectations in Equation (4) and Equation (5). In implementation, we also apply normalization to states, which is widely used in previous work (Fujimoto & Gu, 2021; Raffin et al., 2021). This also helps to deal with the variations of states in the fine-tuning phase.

4.3 Theoretical Analysis

Our analyses are conducted in linear MDP assumption (Jin et al., 2020, 2021), where the transition kernel and the reward function are linear in a given state-action feature $\phi(s,a)$. We estimate the value function by $Q(s,a) \approx \widehat{w}^\top \phi(s,a)$. See the appendix for the details.

We start by considering the offline training phase where the value function is learned from $\mathcal{D}_{\text{offline}}$. According to the loss in Equation (4), the parameter $\widehat{w}$ can be solved by

$$\begin{aligned}\widetilde{w}_{\text{offline}} = \min_{w\in\mathcal{R}^d}\Big[\sum_{i=1}^{m}\big(y_t^i - Q_w(s_t^i,a_t^i)\big)^2 + \sum_{(\hat{s},\hat{a},\hat{y})\sim\mathcal{D}_{\text{ood}}}\big(\hat{y} - Q_w(\hat{s},\hat{a})\big)^2 \\ + \sum_{i=1}^{m}\frac{1}{|\mathbb{B}_d(s_t^i,\epsilon)|}\sum_{(\hat{s}_t^i,s_t^i,a_t^i)\in\mathcal{D}_{\text{robust}}}\big(Q_w(s_t^i,a_t^i) - Q_w(\hat{s}_t^i,a_t^i)\big)^2\Big],\end{aligned} \tag{6}$$

where we denote $y = \widehat{\mathcal{T}}Q$ and $\hat{y} = \widehat{\mathcal{T}}^{\text{ood}}Q$ as the learning targets for simplicity. The three terms in Equation (6) correspond to TD-loss, OOD penalty, and smoothness constraints, respectively. For the clarity of notations, we explicitly define a dataset $\mathcal{D}_{\text{ood}}$ for OOD sampling, and an adversarial dataset $\mathcal{D}_{\text{robust}}$ for the smoothness term. Following Least-Squares Value Iteration (LSVI) (Jin et al., 2020), the solution of Equation (6) takes the following form as

$$\widetilde{w}_t = \widetilde{\Lambda}_t^{-1}\Big(\sum_{i=1}^{m}\phi(s_t^i,a_t^i)y_t^i + \sum_{(\hat{s},\hat{a},\hat{y})\sim\mathcal{D}_{\text{ood}}}\phi(\hat{s},\hat{a})\hat{y}\Big), \tag{7}$$

and the covariance matrix $\widetilde{\Lambda}_t$ is

$$\widetilde{\Lambda}_t = \widetilde{\Lambda}_t^{\text{in}} + \widetilde{\Lambda}_t^{\text{ood}} + \widetilde{\Lambda}_t^{\text{robust}}, \tag{8}$$

where the first term $\widetilde{\Lambda}_t^{\text{in}} = \sum_{i=1}^{m}\phi(s_t^i,a_t^i)\phi(s_t^i,a_t^i)^\top$ is calculated on in-distribution (i.e., in $\mathcal{D}_{\text{offline}}$) data, the second term is $\widetilde{\Lambda}_t^{\text{ood}} = \sum_{\mathcal{D}_{\text{ood}}}\phi(\hat{s}_t,\hat{a}_t)\phi(\hat{s}_t,\hat{a}_t)^\top$ is calculated on OOD samples (i.e., in $\mathcal{D}_{\text{ood}}$), and the third term is calculated on adversarial samples (i.e., in $\mathcal{D}_{\text{robust}}$), as $\widetilde{\Lambda}_t^{\text{robust}} = \sum_{i=1}^{m}\frac{1}{|\mathbb{B}_d|}\sum_{\mathcal{D}_{\text{robust}}}\big[\phi(\hat{s},a)-\phi(s,a)\big]\big[\phi(\hat{s},a)-\phi(s,a)\big]^\top$.

For comparison, we consider a variant of RO2O without smoothness regularization. Following LSVI, the solution of this variant takes a similar form as Equation (7), but with a different covariance matrix as $\widetilde{\Lambda}_t^{\text{in}} + \widetilde{\Lambda}_t^{\text{ood}}$. The difference in covariance matrices originates from the additional adversarial samples in RO2O. We denote the dataset for RO2O as $\mathcal{D}_{\text{RO2O}} = \mathcal{D}_{\text{offline}} \cup \mathcal{D}_{\text{ood}} \cup \mathcal{D}_{\text{robust}}$, and for this variant as $\mathcal{D}_{\text{variant}} = \mathcal{D}_{\text{offline}} \cup \mathcal{D}_{\text{ood}}$ without smoothness constraints.

Following the theoretical framework in PEVI (Jin et al., 2021), the sub-optimality gap of offline RL algorithms with uncertainty penalty is upper-bounded by the lower-confidence-bound (LCB) term, defined by

$$\Gamma^{\text{lcb}}(s_t,a_t;\mathcal{D}) = \beta_t\big[\phi(s_t,a_t)^\top\Lambda_t^{-1}\phi(s_t,a_t)\big]^{1/2},$$

where the form of Λ_t depends on the learned dataset (e.g., $\mathcal{D}_{\text{RO2O}}$ or $\mathcal{D}_{\text{variant}}$), and β_t is a factor. Then the following theorem shows our smoothness regularization leads to smaller uncertainties for arbitrary state-action pairs, especially for OOD samples (e.g., from online interactions).

Theorem 1. *Assuming that the size of adversarial samples $\mathbb{B}_d(s_t^i, \epsilon)$ is sufficient and the Jacobian matrix of $\phi(s,a)$ has full rank, the smoothness constraint leads to smaller uncertainty for $\forall(s^\star, a^\star) \in \mathcal{S} \times \mathcal{A}$, as*

$$\Gamma^{\text{lcb}}(s^\star, a^\star; \mathcal{D}_{\text{RO2O}}) < \Gamma^{\text{lcb}}(s^\star, a^\star; \mathcal{D}_{\text{variant}}),$$

where the covariance matrices for these two LCB terms are $\widetilde{\Lambda}_t$ in Equation (8) and $\widetilde{\Lambda}_t^{\text{in}} + \widetilde{\Lambda}_t^{\text{ood}}$, respectively.

According to Theorem 1, with the additional term $\widetilde{\Lambda}_t^{\text{robust}} = \sum_{i=1}^{m} \frac{1}{|\mathbb{B}_d|} \sum_{\mathcal{D}_{\text{robust}}} \big[\phi(\hat{s}, a) - \phi(s,a)\big]\big[\phi(\hat{s}, a) - \phi(s,a)\big]^\top$ in the covariance matrix $\widetilde{\Lambda}_t$ of $\mathcal{D}_{\text{RO2O}}$, the uncertainty of OOD samples measured by UCB will be reduced. As an extreme example in tabular case, the uncertainty for a purely OOD $(s^\star, a^\star)$ pair can be large as $\Gamma^{\text{lcb}}(s^\star, a^\star; \mathcal{D}_{\text{variant}}) \to \infty$ without the smoothness term, while $\Gamma^{\text{lcb}}(s^\star, a^\star; \mathcal{D}_{\text{RO2O}}) \leq \beta_t/\sqrt{\lambda}$ with $\lambda > 0$. As a result, RO2O is more robust to significant distribution shift theoretically. See appendix for the proof.

Then, for online fine-tuning with new data from $\mathcal{D}_{\text{online}}$, the following theorem shows RO2O can consistently reduce the sub-optimality gap with online fine-tuning, as

Theorem 2. *Under the same conditions as Theorem 1, with additional online experience in the fine-tuning phase, the sub-optimality gap holds for RO2O in linear MDPs, as*

$$\begin{aligned}\text{SubOpt}(\pi^*, \widetilde{\pi}) &\leq \sum_{t=1}^{T} \mathbb{E}_{\pi^*}\big[\Gamma_i^{\text{lcb}}(s_t, a_t; \mathcal{D}_{\text{RO2O}} \cup \mathcal{D}_{\text{online}})\big] \\ &\leq \sum_{t=1}^{T} \mathbb{E}_{\pi^*}\big[\Gamma_i^{\text{lcb}}(s_t, a_t; \mathcal{D}_{\text{RO2O}})\big],\end{aligned}$$

where $\widetilde{\pi}$ and π^* are the learned policy and the optimal policy in $\mathcal{D}_{\text{RO2O}} \cup \mathcal{D}_{\text{online}}$, respectively.

Theorem 2 shows that the optimality gap shrinks if the data coverage of π^* is better. See appendix for the proof. Considering a sub-optimal dataset is used in offline training, via interacting and learning in online fine-tuning, the agent is potential to obtain high-quality data to consistently reduce the sub-optimality and result in a near-optimal policy.

5. Experiments

We present a comprehensive evaluation of RO2O in the context of the Offline-to-Online RL setting. Specifically, we investigate whether RO2O can perform favorable offline training and further policy improvement given limited interactions. We compare RO2O to existing offline RL algorithms in offline pre-training and also compare it to offline-to-online algorithms in online adaptation. We also conduct ablation studies and visualizations to illustrate the effectiveness of our method.

5.1 Setups and Baselines

Our experiments are conducted on challenging environments from the D4RL (Fu et al., 2020) benchmark, specifically focusing on the Mujoco and AntMaze tasks. These environments are carefully selected to simulate real-world scenarios with limited data availability. We compare RO2O with the following RL algorithms: (i) PEX (Zhang et al., 2023): A recent offline-to-online method that performs policy expansion has shown promising results in longer online

interaction steps. (ii) AWAC (Nair et al., 2020): An efficient algorithm that employs an advantage-weighted actor-critic framework, which is one of the earlier methods to achieve policy improvement during the online fine-tuning phase. (iii) IQL (Kostrikov et al., 2022): A state-of-art offline algorithm that attempts to conduct in-sample learning and expected regression. (iv) Cal-QL (Nakamoto et al., 2023): An efficient algorithm calibrates Q-values within a reasonable range to improve policy performance. (v) SPOT (Wu et al., 2022): An algorithm utilizes density regularization to limit the difference between the learning policy and the current policy. (vi) SAC (Haarnoja et al., 2018): A SAC agent trained from scratch which highlights the benefit of Offline-to-Online RL, as opposed to fully online RL, in terms of learning efficiency.

5.2 Performance Comparisons

We conducted our comparisons using multiple offline datasets and tasks. In this study, we exclude the random datasets, as in typical real-world scenarios, we rarely use a random policy for system control. For the Mujoco locomotion tasks, we conducted 2.5M training steps over all datasets during the offline pre-training phase. Then, we proceeded with online fine-tuning, involving an additional 250K environment interactions. For the AntMaze navigation tasks, we performed 1M training steps on six types of datasets with different complexities and qualities, followed by 250K additional online interactions. It is worth noting that we all use an ensemble size of 10 for training in both tasks. More details about experiments and implementations are introduced in the Appendix B and C.

Offline performance on MuJoCo locomotion tasks First of all, we evaluate the performance of each method on Mujoco locomotion tasks, which include three environments: *Halfcheetah*, *Walker2d*, and *Hopper*. Different types of datasets are selected for offline pre-training, including medium, medium-replay, medium-expert, and expert datasets. Table 1 reports the offline performance of the average normalized score across five seeds. Compared to other algorithms, RO2O has certain superiority in the offline training phase.

Table 1: Offline performance on MuJoCo locomotion tasks.

Environment	PEX	AWAC	IQL	SPOT	Cal-QL	RO2O
halfcheetah-medium	48.67 ± 0.15	50.00 ± 0.27	48.33 ± 0.35	46.78 ± 0.50	47.75 ± 0.38	**66.08 ± 0.45**
halfcheetah-medium-replay	44.57 ± 0.47	45.28 ± 0.31	43.75 ± 0.97	43.29 ± 0.47	46.26 ± 0.57	**60.89 ± 1.01**
halfcheetah-medium-expert	78.9 ± 11.77	94.73 ± 0.64	94.19 ± 0.30	94.70 ± 1.02	67.14 ± 7.40	**104.73 ± 2.07**
halfcheetah-expert	91.2 ± 4.43	97.57 ± 0.94	97.11 ± 0.19	95.21 ± 0.93	96.5 ± 0.66	**104.08 ± 1.66**
walker2d-medium	61.87 ± 2.06	84.24 ± 1.15	83.96 ± 2.68	56.81 ± 3.96	64.07 ± 7.61	**103.25 ± 1.67**
walker2d-medium-replay	38.4 ± 13.36	80.92 ± 1.70	77.28 ± 7.45	70.49 ± 22.57	**94.48 ± 6.44**	93.05 ± 4.74
walker2d-medium-expert	98.8 ± 4.78	112.62 ± 0.66	111.24 ± 0.92	77.58 ± 8.49	108.26 ± 5.56	**120.01 ± 0.70**
walker2d-expert	103.13 ± 6.69	91.66 ± 35.78	112.67 ± 0.21	105.13 ± 13.03	111.93 ± 0.24	**112.84 ± 3.42**
hopper-medium	51.3 ± 5.07	71.33 ± 8.80	56.33 ± 2.83	82.25 ± 2.16	83.34 ± 0.91	**104.95 ± 0.03**
hopper-medium-replay	77.9 ± 5.77	96.56 ± 2.23	82.55 ± 17.57	70.37 ± 12.51	85.59 ± 1.85	**103.77 ± 0.47**
hopper-medium-expert	46.73 ± 48.88	108.36 ± 3.12	85.21 ± 39.43	96.52 ± 12.03	108.82 ± 0.21	**112.69 ± 0.02**
hopper-expert	102.27 ± 6.11	103.88 ± 8.98	100.36 ± 9.96	110.00 ± 0.41	107.29 ± 3.50	**112.31 ± 0.01**

Fine-tuning performance on Mujoco locomotion tasks Figure 3 illustrates the fine-tuning performance of different methods on Mujoco locomotion tasks. Compared with pure online RL methods such as SAC, other methods almostly reflect the advantages of offline pre-training. Within limited online interactions, IQL, AWAC and SPOT fail to achieve effective policy improvement, while PEX suffers from the performance drop. For PEX, we speculate that it is due to the randomness of strategies expanded in the online phase, which could lead to a poor initial strategy and requires lots of interactions to improve its

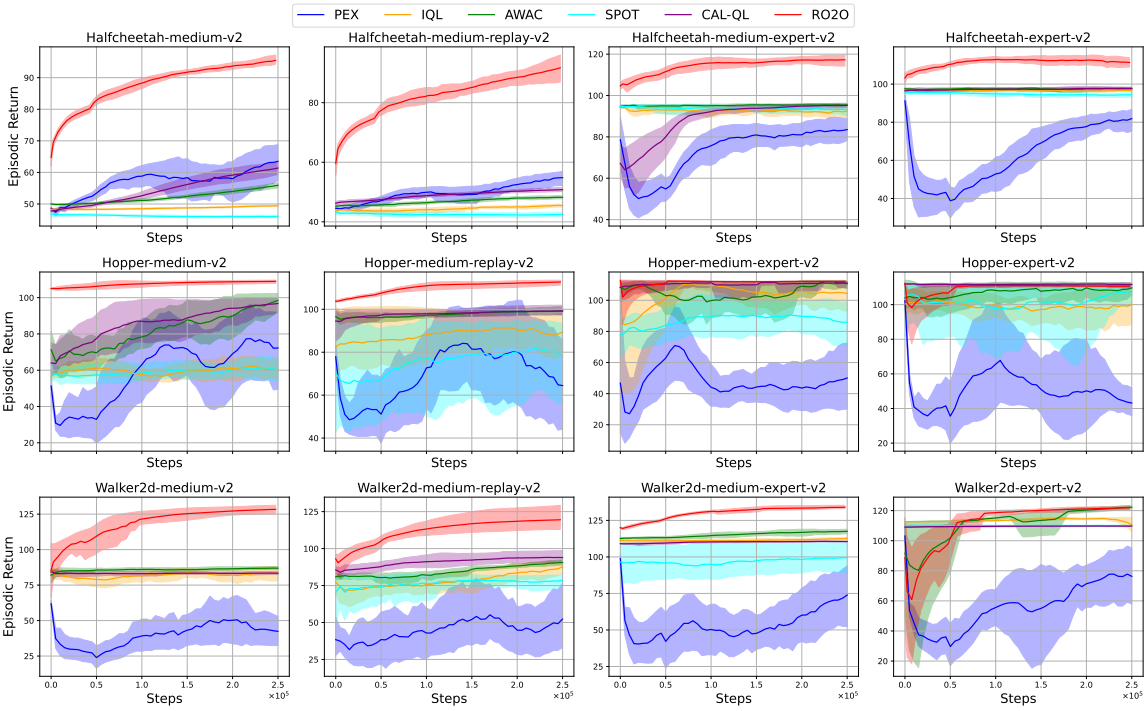

Figure 3: Fine-tuning performance curves of different methods across five seeds on MuJoCo locomotion tasks. The mean and standard deviation are shown by the solid lines and the shaded areas, respectively.

performance. For Cal-QL, it can achieve effective performance improvement in most tasks, but the improvement is still relatively limited. In contrast, RO2O exhibits a significant improvement in performance during the fine-tuning process and requires fewer steps to achieve the highest score. Compared with them, RO2O showcases comparable or better performance with 250K fine-tuning steps, indicating the efficiency and superiority.

Offline performance on AntMaze navigation tasks We also perform evaluations on the challenging AntMaze navigation tasks, where agents must learn to control the robot and stick trajectories together given sparse rewards. Agents are pre-trained on six types of datasets with different complexity and quality. Considering the poor performance of AWAC and SAC in this task, we have opted not to compare our approach with them. Table 2 reports the normalized scores using different methods across five seeds. We observe that RO2O achieves the best performance on almost all tasks.

Fine-tuning performance on AntMaze navigation tasks Figure 4 demonstrates the fine-tuning performance on AntMaze tasks. In the fine-tuning phase, IQL and SPOT achieve stable learning but limited improvement, while PEX suffers from a performance drop. Cal-QL rapidly enhances its performance from a poor initial policy, but cannot perform well in large scenarios. Different from these baselines, RO2O achieves significant improvement over all tasks within limited interactions. It provides good initial performance for online fine-tuning within limited interaction steps, demonstrating considerable advantages. In

Table 2: Offline performance on the challenging AntMaze environment.

Environments	PEX	IQL	SPOT	Cal-QL	RO2O
antmaze-umaze	87.33 ± 4.04	77.0 ± 6.38	89 ± 5.29	65.75 ± 4.03	**93.67 ± 5.77**
antmaze-umaze-diverse	58.67 ± 9.07	**65.24 ± 6.40**	42.75 ± 5.32	48.75 ± 4.43	63.67 ± 8.02
antmaze-medium-diverse	72.33 ± 5.13	73.75 ± 6.30	74.25 ± 4.99	1.25 ± 0.96	**91.67 ± 5.13**
antmaze-medium-play	68 ± 6.56	66 ± 7.55	71.5 ± 8.43	0.0 ± 0.0	**86.67 ± 3.06**
antmaze-large-diverse	45.67 ± 4.16	30.25 ± 4.20	36.5 ± 17.62	0.0 ± 0.0	**65.33 ± 5.71**
antmaze-large-play	51 ± 17.69	42.0 ± 5.23	30.25 ± 17.91	0.25 ± 0.5	**61.33 ± 9.82**

summary, RO2O achieves robust and state-of-the-art performance in comparison to all baseline methods in almost all tasks.

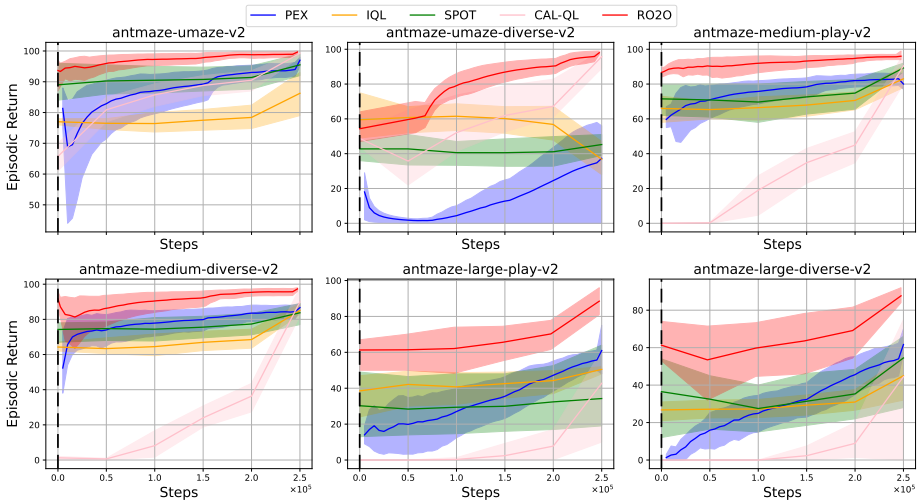

Figure 4: Fine-tuning performance curves of different methods across five seeds on Antmaze navigation tasks. The mean and standard deviation are shown by the solid lines and the shaded areas, respectively.

Robustness Analysis RO2O demonstrates robust performance in all environments except for some scenarios, such as AntMaze-large-play and AntMaze-large-diverse. The performance curves seem to exhibit more fluctuating behaviors, albeit with increasing steps, the performances are superior. We believe there are several reasons: (i) Antmaze tasks provide binary rewards, so that similar policies could obtain largely different returns, thus leading to fluctuated performance. While the regularization terms in RO2O aim to present smooth Q-functions and policies, their effects can be limited when the environment is complex, as in a large maze. (ii) In Antmaze tasks, RO2O utilizes independent TD targets instead of shared TD targets in the Bellman update of Q-values, which in some degree increases the diversity of Q-value estimation for OOD actions. During online fine-tuning phase, RO2O leverages the maximum of ensemble estimation of Q-values as the TD target to encourage the exploration of policies. This causes that agents tend to choose OOD actions, which may not perform well and also lead to fluctuated performance.

5.3 Ablation Study and Visualization

We analyze the effects of the smoothness regularization and OOD sampling terms on the learning process. Specifically, we consider variants of RO2O without policy smoothing, Q-smoothing, or OOD penalty. We conduct the ablation studies on walker2d-medium and hopper-medium tasks. Figure 5 demonstrates the experimental results over three random seeds. In the offline process, we observe that OOD penalty is indispensable to prevent divergence caused by OOD actions. However, it becomes trivial in the online phase since new states or actions may lead to high values and better policies. We also find that policy smoothing and Q-smoothing are useful in the offline-to-online process to obtain an effective improvement and mitigate the variance of performance.

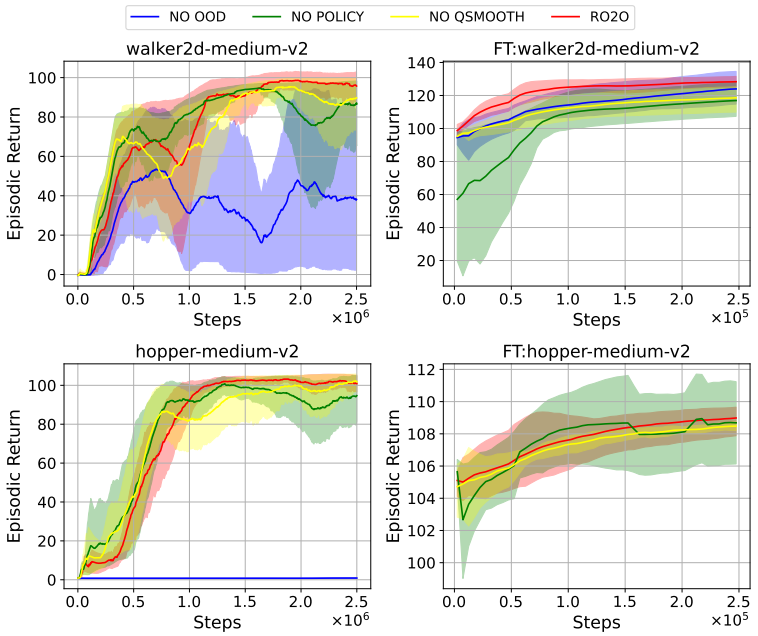

Figure 5: Offline (left column) and online performance (right column) when eliminating OOD penalty, policy smoothing, or Q-smoothing.

Additional, we employ various coefficients of the different losses η_1, η_2, η_3 and the radius of the perturbation set ϵ to investigate the algorithm's sensitivity to the hyperparameter, where η_1 maintains a constant value of 0.0001, η_2 is tuned within $\{0.0,\ 0.1,\ 0.5\}$, η_3 is searched in $\{0.1,\ 1.0\}$ and ϵ is tuned within $\{0.0,\ 0.005,\ 0.01\}$. The results shown in Figure 6 demonstrate that the coefficients of the different losses are a critical factor for both offline and online. We observed that the algorithm is relatively sensitive to the choice of parameters, especially η_3, where even small changes can lead to significant performance degradation. Constraints on the perturbation set behavior policy may have a greater impact compared to other coefficients. Moreover, in addition to adjusting the coefficients of the different loss functions, we can also modify the radius of the perturbation set to avoid excessive pessimism. Additionally, we notice that

To better understand the effectiveness of RO2O in the offline-to-online process, we compare the distribution of the offline states and the visited states in online interactions, as

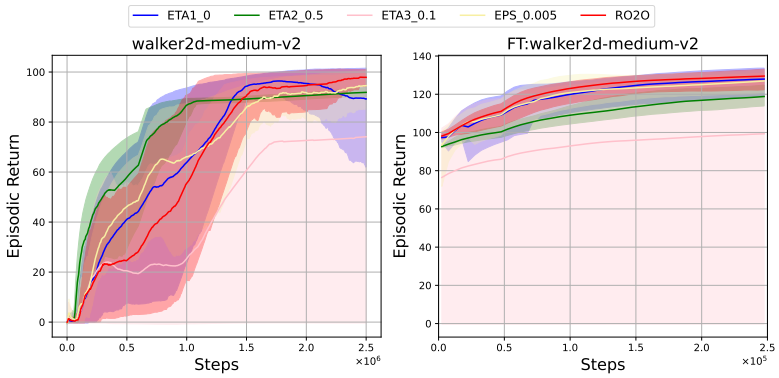

Figure 6: Sensitivity analysis of the various coefficients η_1, η_2, η_3 and the radius of the perturbation set ϵ.

shown in Figure 7 (left). The states are visualized via T-SNE. Meanwhile, we use brightness to represent the corresponding reward for each state, as shown in Figure 7 (right). We find that, with consistent policy improvement in online fine-tuning, the agent can obtain high-quality (i.e., with high reward) online experiences that are different from the offline data. Such a phenomenon verifies our theoretical analysis in Theorem 2, where RO2O can consistently reduce the sub-optimality gap and improve the policy via online fine-tuning.

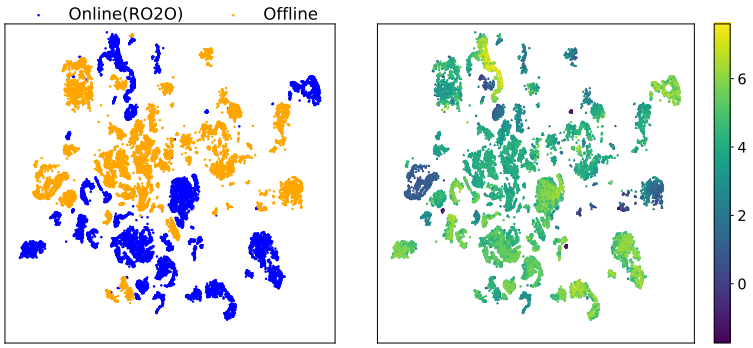

Figure 7: Visualization of the distribution of states (left) and rewards (right) from offline data and online experiences.

6. Computational Cost Comparison

We compare the computational cost of RO2O against baselines. All methods are run on a single machine with one GPU (NVIDIA GeForce RTX 3090). For each method, we measure the average epoch time (i.e., 1×10^3 training steps) and the GPU memory usage on the walker2d-medium-v2 task. The results in Table 3 show that although RO2O includes the OOD state-action sampling and the robust training procedure, it does not significantly lag behind other baselines in terms of runtime. And we implemented these procedures efficiently based on the parallelization of Q networks.

Table 3: The computational cost of various algorithms on walker2d-medium-v2.

	Runtime (s/epoch)	GPU Memory (GB)
PEX	5.18	2.17
AWAC	8.44	5.20
IQL	6.31	2.38
SAC-10	7.82	2.23
SPOT	5.24	5.20
Cal-QL	15.7	2.69
RO2O	21.8	2.85

7. Conclusion

In this paper, we propose RO2O for Offline-to-Online RL by incorporating Q-ensembles and smoothness regularization. By regularizing the smoothness of value and policy, RO2O achieves stable offline learning and effective policy improvement in online fine-tuning. Moreover, RO2O maintains the same architecture in the offline-to-online process without specific modifications. Empirical results on Mujoco and AntMaze tasks demonstrate the effectiveness and superiority of RO2O. Future work may explore ways to perform offline-to-online learning with domain gaps, including dynamics or reward differences.

Acknowledgments

This work was completed jointly by Xiaoyu Wen and Xudong Yu as co-first authors and was supported by the National Science Fund for Distinguished Young Scholars (Grant No.62025602), the National Natural Science Foundation of China (Grant Nos. 62306242, U22B2036, 11931915), Fok Ying-Tong Education Foundation China (No.171105), the Tencent Foundation, and XPLORER PRIZE.

Appendix A. Theoretical Analysis

A.1 RO2O Algorithm in Linear MDPs

We consider the loss function of RO2O algorithm in offline learning, which contains temporal-difference (TD) error, smoothness loss, and OOD penalty. Converting the loss function in linear MDPs, the parameter $\widehat{w}$ in RO2O can be solved by

$$\begin{aligned}\widetilde{w}_{\text{offline}} = \min_{w\in\mathcal{R}^d}\Big[\sum_{i=1}^{m}\big(y_t^i - Q_w(s_t^i,a_t^i)\big)^2 + \sum_{(\hat{s},\hat{a},\hat{y})\sim\mathcal{D}_{\text{ood}}}\big(\hat{y} - Q_w(\hat{s},\hat{a})\big)^2 \\ + \sum_{i=1}^{m}\frac{1}{|\mathbb{B}_d(s_t^i,\epsilon)|}\sum_{(\hat{s}_t^i,s_t^i,a_t^i)\in\mathcal{D}_{\text{robust}}}\big(Q_w(s_t^i,a_t^i) - Q_w(\hat{s}_t^i,a_t^i)\big)^2\Big],\end{aligned} \tag{9}$$

where $y = \widehat{\mathcal{T}}Q$ and $\hat{y} = \widehat{\mathcal{T}}^{\text{ood}}Q$ denote the learning targets for simplicity and $|\mathbb{B}_d(s_t^i,\epsilon)|$ means the size of adversarial samples. The three terms in Equation (9) correspond to TD-loss, OOD penalty, and smoothness constraints, respectively. For the clarity of notations, we explicitly define a dataset $\mathcal{D}_{\text{ood}}$ for OOD sampling, and an adversarial dataset $\mathcal{D}_{\text{robust}}$ for

the smoothness constraint. Following LSVI (Jin et al., 2020), the solution of Equation (9) takes the following form as

$$\widetilde{w}_t = \widetilde{\Lambda}_t^{-1}\Big(\sum_{i=1}^{m}\phi(s_t^i,a_t^i)y_t^i + \sum_{(\hat{s},\hat{a},\hat{y})\sim\mathcal{D}_{\text{ood}}}\phi(\hat{s},\hat{a})\hat{y}\Big), \tag{10}$$

and the covariance matrix $\widetilde{\Lambda}_t$ is

$$\begin{aligned}\widetilde{\Lambda}_t &= \widetilde{\Lambda}_t^{\text{in}} + \widetilde{\Lambda}_t^{\text{ood}} + \widetilde{\Lambda}_t^{\text{robust}} \\ &= \sum_{i=1}^{m}\phi(s_t^i,a_t^i)\phi(s_t^i,a_t^i)^\top + \sum_{\mathcal{D}_{\text{ood}}}\phi(\hat{s}_t,\hat{a}_t)\phi(\hat{s}_t,\hat{a}_t)^\top \\ &+ \sum_{i=1}^{m}\frac{1}{|\mathbb{B}_d(s_t^i,\epsilon)|}\sum_{\mathcal{D}_{\text{robust}}}\big[\phi(\hat{s}_t^i,a_t^i)-\phi(s_t^i,a_t^i)\big]\big[\phi(\hat{s}_t^i,a_t^i)-\phi(s_t^i,a_t^i)\big]^\top,\end{aligned} \tag{11}$$

where the first term $\widetilde{\Lambda}_t^{\text{in}}$ is calculated on in-distribution (i.e., in $\mathcal{D}_{\text{offline}}$) data, the second term is $\widetilde{\Lambda}_t^{\text{ood}}$ is calculated on OOD samples (i.e., in $\mathcal{D}_{\text{ood}}$), and the third term is calculated on adversarial samples (i.e., in $\mathcal{D}_{\text{robust}}$), .

For comparison, we consider a variant of RO2O without smoothness regularization, and denote it as 'variant'. The parameter of this variant can be solved by

$$\widetilde{w}_{\text{variant}} = \min_{w\in\mathcal{R}^d}\Big[\sum_{i=1}^{m}\big(y_t^i - Q_w(s_t^i,a_t^i)\big)^2 + \sum_{(\hat{s},\hat{a},\hat{y})\sim\mathcal{D}_{\text{ood}}}\big(\hat{y}-Q_w(\hat{s},\hat{a})\big)^2\Big], \tag{12}$$

Following LSVI, the solution of this variant takes a similar form as Equation (10), but with a different covariance matrix as

$$\widetilde{\Lambda}_t^{\text{variant}} = \widetilde{\Lambda}_t^{\text{in}} + \widetilde{\Lambda}_t^{\text{ood}}.$$

We remark that the difference in covariance matrices between RO2O and this variant originates from the additional adversarial samples from $\mathcal{D}_{\text{robust}}$ used in RO2O. We denote the dataset for RO2O as

$$\mathcal{D}_{\text{RO2O}} = \mathcal{D}_{\text{offline}} \cup \mathcal{D}_{\text{ood}} \cup \mathcal{D}_{\text{robust}},$$

and for this variant as

$$\mathcal{D}_{\text{variant}} = \mathcal{D}_{\text{offline}} \cup \mathcal{D}_{\text{ood}}$$

without smoothness constraints.

A.2 Effective of Smoothness with LCB

Following the theoretical framework in PEVI (Jin et al., 2021), the sub-optimality gap of offline RL algorithms with uncertainty penalty is upper-bounded by the lower-confidence-bound (LCB) term, defined by

$$\Gamma^{\text{lcb}}(s_t,a_t;\mathcal{D}) = \beta_t\big[\phi(s_t,a_t)^\top\Lambda_t^{-1}\phi(s_t,a_t)\big]^{1/2},$$

which forms an uncertainty quantification with the covariance matrix Λ_i^{-1} given the dataset $\mathcal{D}_i$ (Jin et al., 2020, 2021), and the form of Λ_t depends on the learned dataset (e.g., $\mathcal{D}_{\text{RO2O}}$ or $\mathcal{D}_{\text{variant}}$). β_t is a factor. LCB measures the confidence interval of Q-function learned by the given dataset. Intuitively, $\Gamma_i^{\text{lcb}}(s,a)$ can be considered as a reciprocal pseudo-count of the state-action pair in the representation space.

In the following, we aim to show the smoothness regularization leads to smaller uncertainties for arbitrary state-action pairs, especially for OOD samples (e.g., from online interactions). We start by building a Lemma to show the covariance matrix $\widetilde{\Lambda}_t^{\text{robust}}$ introduced by smoothness regularization calculated in $\mathcal{D}_{\text{robust}}$ is positive-definite.

Lemma 1. *Assuming that the size of adversarial samples $\mathbb{B}_d(s_t^i,\epsilon)$ is sufficient and the Jacobian matrix of $\phi(s,a)$ has full rank, then the covariance matrix $\widetilde{\Lambda}_t^{\text{robust}}$ is positive-definite: $\widetilde{\Lambda}_t^{\text{robust}} \succeq \lambda \cdot \mathrm{I}$ where $\lambda > 0$.*

Proof. For the $\widetilde{\Lambda}_t^{\text{robust}}$ matrix (i.e., the third part in Eq. (8)), we denote the covariance matrix for a specific i as Φ_t^i. Then we have $\widetilde{\Lambda}_t^{\text{ood\_diff}} = \sum_{i=1}^m \Phi_t^i$. In the following, we discuss the condition of positive-definiteness of Φ_t^i. For the simplicity of notation, we omit the superscript and subscript of s_t^i and a_t^i for given i and t. Specifically, we define

$$\Phi_t^i = \frac{1}{|\mathbb{B}_d(s_t^i,\epsilon)|} \sum_{\hat{s}_j \sim \mathcal{D}_{\text{ood}}(s)} \big[\phi(\hat{s}_j,a) - \phi(s,a)\big]\big[\phi(\hat{s}_j,a) - \phi(s,a)\big]^\top,$$

where $j \in \{1,\dots,N\}$ indicates we sample $|\mathbb{B}_d(s_t^i,\epsilon)| = N$ perturbed states for each s. For a nonzero vector $y \in \mathbb{R}^d$, we have

$$\begin{aligned} y^\top \Phi_t^i y &= y^\top \left(\frac{1}{N} \sum_{j=1}^N \big(\phi(\hat{s}_j,a) - \phi(s,a)\big)\big(\phi(\hat{s}_j,a) - \phi(s,a)\big)^\top \right) y \\ &= \frac{1}{N} \sum_{j=1}^N y^\top \big(\phi(\hat{s}_j,a) - \phi(s,a)\big)\big(\phi(\hat{s}_j,a) - \phi(s,a)\big)^\top y \\ &= \frac{1}{N} \sum_{j=1}^N \Big(\big(\phi(\hat{s}_j,a) - \phi(s,a)\big)^\top y\Big)^2 \geq 0, \end{aligned} \tag{13}$$

where the last inequality follows from the observation that $\big(\phi(\hat{s}_j,a) - \phi(s,a)\big)^\top y$ is a scalar. Then Φ_t^i is always positive semi-definite. In the following, we denote $z_j = \phi(\hat{s}_j,a) - \phi(s,a)$. Then we need to prove that the condition to make Φ_t^i positive definite is $\text{rank}[z_1,\dots,z_N] = d$, where d is the feature dimension. Our proof follows contradiction.

In Equation (13), when $y^\top \Phi_t^i y = 0$ with a nonzero vector y, we have $z_j^\top y = 0$ for all $j = 1,\dots,N$. Suppose the set $\{z_1,\dots,z_N\}$ spans $\mathbb{R}^d$, then there exist real numbers $\{\alpha_1,\dots,\alpha_N\}$ such that $y = \alpha_1 z_1 + \cdots + \alpha_N z_N$. But we have $y^\top y = \alpha_1 z_1^\top y + \cdots + \alpha_N z_N^\top y = \alpha_1 \times 0 + \ldots + \alpha_N \times 0 = 0$, yielding that $y = \mathbf{0}$, which forms a contradiction. Hence, if the set $\{z_1,\dots,z_N\}$ spans $\mathbb{R}^d$, which is equivalent to $\text{rank}[z_1,\dots,z_N] = d$, then Φ_t^i is positive definite.

Under the given conditions, since the size of samples $\mathbb{B}_d(s_t^i,\epsilon)$ is sufficient and the neural network maintains useful variability to make the Jacobian matrix of $\phi(s,a)$ have full rank, it

ensures that $\exists k \in [1, m]$, for any nonzero vector $y \in \mathbb{R}^d$, $y^\top \Phi_t^k y > 0$. We have $y^\top \widetilde{\Lambda}_t^{\text{robust}} y = \sum_{i=1}^m y^\top \Phi_t^i y \geq y^\top \Phi_t^k y > 0$. Therefore, $\widetilde{\Lambda}_t^{\text{robust}}$ is positive definite, which concludes our proof. □

Recall the covariance matrix of the variant algorithm without smoothness constraint is $\widetilde{\Lambda}_t^{\text{variant}} = \widetilde{\Lambda}_t^{\text{in}} + \widetilde{\Lambda}_t^{\text{ood}}$, and RO2O has a covariance matrix as $\widetilde{\Lambda}_t = \widetilde{\Lambda}_t^{\text{variant}} + \widetilde{\Lambda}_t^{\text{robust}}$, we have the following corollary based on Lemma 1.

Theorem 1 (restate)**.** *Assuming that the size of adversarial samples $\mathbb{B}_d(s_t^i, \epsilon)$ is sufficient and the Jacobian matrix of $\phi(s, a)$ has full rank, the smoothness constraint leads to smaller uncertainty for $\forall (s^\star, a^\star) \in \mathcal{S} \times \mathcal{A}$, as*

$$\Gamma^{\text{lcb}}(s^\star, a^\star; \mathcal{D}_{\text{RO2O}}) < \Gamma^{\text{lcb}}(s^\star, a^\star; \mathcal{D}_{\text{variant}}),$$

where the covariance matrices for these two LCB terms are $\widetilde{\Lambda}_t$ in Equation (11) *and $\widetilde{\Lambda}_t^{\text{in}} + \widetilde{\Lambda}_t^{\text{ood}}$, respectively.*

Proof. According to Lemma 1, since $\Lambda_t^{\text{robust}}$ is positive-definite, we have $\Lambda_t^{\text{robust}} \succeq \lambda I$ with a factor $\lambda > 0$. Meanwhile, the factor λ can be large if we have sufficient adversarial samples and also with large variability in adversarial samples. By assuming $\widetilde{\Lambda}_t^{\text{in}} + \widetilde{\Lambda}_t^{\text{ood}}$ is positive definite and leveraging the properties of generalized Rayleigh quotient, we have

$$\begin{aligned}
\frac{\phi^\top (\widetilde{\Lambda}_t^{\text{variant}})^{-1} \phi}{\phi^\top (\widetilde{\Lambda}_t^{\text{variant}} + \widetilde{\Lambda}_t^{\text{robust}})^{-1} \phi} &\geq \lambda_{\min}\big((\widetilde{\Lambda}_t^{\text{variant}} + \widetilde{\Lambda}_t^{\text{robust}})(\widetilde{\Lambda}_t^{\text{variant}})^{-1}\big) \\
&= \lambda_{\min}\big(\mathrm{I} + (\widetilde{\Lambda}_t^{\text{robust}})(\widetilde{\Lambda}_t^{\text{variant}})^{-1}\big) \\
&= 1 + \lambda_{\min}\big((\widetilde{\Lambda}_t^{\text{robust}})(\widetilde{\Lambda}_t^{\text{variant}})^{-1}\big).
\end{aligned}$$

Since $\widetilde{\Lambda}_t^{\text{robust}}$ and $(\widetilde{\Lambda}_t^{\text{variant}})^{-1}$ are both positive definite, the eigenvalues of $\widetilde{\Lambda}_t^{\text{robust}}(\widetilde{\Lambda}_t^{\text{variant}})^{-1}$ are all positive: $\lambda_{\min}\big(\widetilde{\Lambda}_t^{\text{robust}}(\widetilde{\Lambda}_t^{\text{variant}})^{-1}\big) > 0$, where $\lambda_{\min}(\cdot)$ is the minimum eigenvalue.

Recall the uncertainty is calculated as $\Gamma^{\text{lcb}}(s_t, a_t; \mathcal{D}) = \beta_t\big[\phi(s_t, a_t)^\top \Lambda_t^{-1} \phi(s_t, a_t)\big]^{1/2}$. Then for $\forall \phi(s^\star, a^\star)$, we have

$$\begin{aligned}
\phi(s^\star, a^\star)^\top (\widetilde{\Lambda}_t^{\text{variant}})^{-1} \phi(s^\star, a^\star) &> \phi(s^\star, a^\star)^\top (\widetilde{\Lambda}_t^{\text{variant}} + \widetilde{\Lambda}_t^{\text{robust}})^{-1} \phi(s^\star, a^\star) \\
&= \phi(s^\star, a^\star)^\top (\widetilde{\Lambda}_t)^{-1} \phi(s^\star, a^\star),
\end{aligned}$$

which means that $\Gamma^{\text{lcb}}(s^\star, a^\star; \mathcal{D}_{\text{variant}}) > \Gamma^{\text{lcb}}(s^\star, a^\star; \mathcal{D}_{\text{RO2O}})$ and concludes our proof. □

As an extreme case in tabular MDPs where the states and actions are finite, the LCB-penalty takes a simpler form. Specifically, we consider the joint state-action space $D = |\mathcal{S}| \times |\mathcal{A}|$. Then j-th state-action pair can be encoded as a one-hot vector as $\phi(s, a) \in \mathbb{R}^D$, where $j \in [0, D-1]$. By considering the tabular MDP as a special case of the linear MDP (Yang & Wang, 2019; Jin et al., 2020), we define

$$\phi(s_j, a_j) = \begin{bmatrix} 0 \\ \vdots \\ \mathrm{i} \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^D, \qquad \phi(s_j, a_j)\phi(s_j, a_j)^\top = \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & & & \vdots \\ 0 & & 1 & & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{D \times D},$$

where the value of $\phi(s_j, a_j)$ is 1 at the j-th entry and 0 elsewhere. Then the matrix $\Lambda_j = \sum_{i=0}^{m} \phi(s_j^i, a_j^i)\phi(s_j^i, a_j^i)^\top$ is a specific covariance matrix based on the learned datasets. It takes the form of

$$\Lambda_j = \begin{bmatrix} n_0 & 0 & & \cdots & 0 \\ 0 & n_1 & & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & & n_j & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & \cdots & & \cdots & n_{d-1} \end{bmatrix},$$

where the j-th diagonal element of Λ_j is the corresponding counts for state-action (s_j, a_j), i.e.,

$$n_j = N_{s_j, a_j}.$$

It thus holds that

$$\left[\phi(s_j, a_j)^\top \Lambda_j^{-1} \phi(s_j, a_j)\right]^{1/2} = \frac{1}{\sqrt{N_{s_j, a_j}}}. \tag{14}$$

For the variant algorithm of RO2O in Equation (12), since the value function is learned from $\mathcal{D}_{\text{variant}}$, the counting function also counts from this dataset. However, without any constraints, the count for a purely OOD state-action pair $(s^\star, a^\star)$ can approach zero, and thus $\Gamma^{\text{lcb}}(s^\star, a^\star; \mathcal{D}_{\text{variant}}) \to \infty$ according to Equation (14). In contrast, as we proved in Lemma 1, the covariance matrix $\widetilde{\Lambda}_t^{\text{robust}}$ for smoothness constraints is positive-definite as $\widetilde{\Lambda}_t^{\text{robust}} \succeq \lambda \cdot \mathrm{I}$ where $\lambda > 0$. Then the covariance matrix for RO2O as $\widetilde{\Lambda}_t \succeq \lambda \cdot \mathrm{I}$ since $\widetilde{\Lambda}_t = \widetilde{\Lambda}_t^{\text{variant}} + \widetilde{\Lambda}_t^{\text{robust}}$. Then, we have $\left[\phi(s_j, a_j)^\top \Lambda_j^{-1} \phi(s_j, a_j)\right]^{1/2} < 1/\sqrt{\lambda}$ and thus $\Gamma^{\text{lcb}}(s^\star, a^\star; \mathcal{D}_{\text{RO2O}}) \leq \beta_t/\sqrt{\lambda}$ with $\lambda > 0$. As a result, RO2O is more robust to significant distribution shift theoretically.

A.3 Sub-optimality Gap of RO2O

To quantify the sub-optimality gap, we start by the following lemma to show the ensemble Q-networks used in RO2O can recover the LCB term in linear MDPs.

Lemma 2 (Equivalence between LCB-penalty and Ensemble Uncertainty)**.** *We assume that the noise in linear regression follows the standard Gaussian, then it holds for the posterior of w given $\mathcal{D}_i$ that*

$$\mathbb{V}_{\widehat{w}}[Q_i(s, a)] = \mathbb{V}_{\widehat{w}}\big(\phi(s, a)^\top \widehat{w}\big) = \phi(s, a)^\top \Lambda^{-1} \phi(s, a), \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}.$$

Proof. We refer to the proof in Lemma 1 of (Bai et al., 2022). □

In RO2O, we choose the minimum value among ensemble Q-networks (i.e., $\min Q_i$) as the learning target, which is equivalent to the uncertainty penalty as *i.e.*, $\bar{Q} - \alpha\sqrt{\mathbb{V}(Q_i)}$ with a specific α (An et al., 2021). The following theorem shows RO2O can consistently reduce the sub-optimality gap with online fine-tuning.

Theorem 2. *Under the same conditions as Theorem 1, with additional online experience in the fine-tuning phase, the sub-optimality gap holds for RO2O in linear MDPs, as*

$$\begin{aligned} \text{SubOpt}(\pi^*, \widetilde{\pi}) &\leq \sum\nolimits_{t=1}^{T} \mathbb{E}_{\pi^*}\left[\Gamma^{\text{lcb}}(s_t, a_t; \mathcal{D}_{\text{RO2O}} \cup \mathcal{D}_{\text{online}})\right] \\ &\leq \sum\nolimits_{t=1}^{T} \mathbb{E}_{\pi^*}\left[\Gamma^{\text{lcb}}(s_t, a_t; \mathcal{D}_{\text{RO2O}})\right], \end{aligned} \tag{15}$$

where $\widetilde{\pi}$ and π^* are the learned policy and the optimal policy in $\mathcal{D}_{\text{RO2O}} \cup \mathcal{D}_{\text{online}}$, respectively.

Proof. Based on the LSVI solution of $\widetilde{w}_{\text{offline}}$, we consider importing additional dataset $\mathcal{D}_{\text{finetune}}$ in online interactions. Following a similar solution procedure as in Equation (9) via LSVI, we obtain the solution of RO2O with online dataset as

$$\widetilde{w}_t^{\text{R2O2}} = (\widetilde{\Lambda}_t^{\text{RO2O}})^{-1}\Big(\sum_{(s,a,y)\sim\mathcal{D}_{\text{offline}}\cup\mathcal{D}_{\text{finetune}}} \phi(s,a)y + \sum_{(\hat{s},\hat{a},\hat{y})\sim\widehat{\mathcal{D}}_{\text{ood}}} \phi(\hat{s},\hat{a})\hat{y}\Big),$$

where $\widehat{\mathcal{D}}_{\text{ood}}$ is a new OOD dataset that contains OOD samples of both the offline and online data. The new covariance matrix $\widetilde{\Lambda}_t^{\text{RO2O}}$ is calculated on samples in both online and offline data,

$$\begin{aligned}
\widetilde{\Lambda}_t^{\text{RO2O}} &= \widetilde{\Lambda}_t + \widetilde{\Lambda}_t^{\text{online}} \\
&= \sum_{\mathcal{D}_{\text{offline}}\cup\mathcal{D}_{\text{finetune}}} \phi(s_t^i, a_t^i)\phi(s_t^i, a_t^i)^\top + \sum_{\widehat{\mathcal{D}}_{\text{ood}}} \phi(\hat{s}_t, \hat{a}_t)\phi(\hat{s}_t, \hat{a}_t)^\top \\
&\quad + \sum_{i=1}^{m} \frac{1}{|\mathbb{B}_d(s_t^i, \epsilon)|} \sum_{\widehat{\mathcal{D}}_{\text{robust}}} \big[\phi(\hat{s}_t^i, a_t^i) - \phi(s_t^i, a_t^i)\big]\big[\phi(\hat{s}_t^i, a_t^i) - \phi(s_t^i, a_t^i)\big]^\top,
\end{aligned} \tag{16}$$

where each term is calculated on both the offline dataset and online fine-tuning dataset since the proposed RO2O algorithm does not change the learning objective in the offline-to-online process. We denote the total data used in online fine-tuning as $\mathcal{D}_{\text{online}}$, which contains the $\mathcal{D}_{\text{finetune}}$ collected in interacting with the environment, the additional adversarial samples, and the OOD samples that are constructed based on $\mathcal{D}_{\text{finetune}}$. Then, $\widetilde{\Lambda}_t^{\text{RO2O}}$ is the covariance matrix of samples from both offline and online datasets, i.e., $\mathcal{D}_{\text{RO2O}} \cup \mathcal{D}_{\text{online}}$.

According to the theoretical framework of pessimistic value-iteration (Jin et al., 2021), value iteration with LCB-based uncertainty penalty is provable efficient in offline RL. Based on the covariance matrix of RO2O, the LCB-term of RO2O learning in offline pre-training and online-fine-tuning are

$$\Gamma^{\text{lcb}}(s_t, a_t; \mathcal{D}_{\text{RO2O}}) = \beta_t[\phi(s_t, a_t)^\top(\widetilde{\Lambda}_t)^{-1}\phi(s_t, a_t)]^{1/2}, \tag{17}$$

$$\text{and} \quad \Gamma^{\text{lcb}}(s_t, a_t; \mathcal{D}_{\text{RO2O}} \cup \mathcal{D}_{\text{online}}) = \beta_t[\phi(s_t, a_t)^\top(\widetilde{\Lambda}_t^{\text{RO2O}})^{-1}\phi(s_t, a_t)]^{1/2}, \tag{18}$$

respectively, where $\widetilde{\Lambda}_t^{\text{RO2O}}$ is defined in Equation (16). According to the definition of ξ-uncertainty quantifier (Jin et al., 2020), $\Gamma^{\text{lcb}}(s_t, a_t; \mathcal{D}_{\text{RO2O}} \cup \mathcal{D}_{\text{online}})$ also forms a valid ξ-uncertainty quantifier under mild assumptions (Yang et al., 2022). According to (Jin et al., 2021), since $\Gamma^{\text{lcb}}(s_t, a_t; \mathcal{D}_{\text{RO2O}})$ is a valid ξ-uncertainty quantifier, the first inequality of Equation (15) holds in quantifying the sub-optimality gap. Further, since $\widetilde{\Lambda}_t^{\text{RO2O}} \succeq \widetilde{\Lambda}_t$ according to Equation 16, we have $\Gamma^{\text{lcb}}(s_t, a_t; \mathcal{D}_{\text{RO2O}} \cup \mathcal{D}_{\text{online}}) \leq \Gamma^{\text{lcb}}(s_t, a_t; \mathcal{D}_{\text{RO2O}})$ by following Equation 18 and 17, which concludes our proof. □

Definition 1 (ξ-Uncertainty Quantifier)**.** *The set of penalization $\{\Gamma_t\}_{t\in[T]}$ forms a ξ-Uncertainty Quantifier if it holds with probability at least $1-\xi$ that*

$$|\widehat{\mathcal{T}}V_{t+1}(s,a) - \mathcal{T}V_{t+1}(s,a)| \leq \Gamma_t(s,a)$$

for all $(s,a) \in \mathcal{S} \times \mathcal{A}$*, where* $\mathcal{T}$ *is the Bellman equation and* $\widehat{\mathcal{T}}$ *is the empirical Bellman equation that estimates* $\mathcal{T}$ *based on the offline data.*

Following PBRL (Bai et al., 2022) and RORL (Yang et al., 2022) that adopt ensemble disagreement as the uncertainty quantifier, in linear MDPs, the proposed ensemble uncertainty $\beta_t \cdot \mathcal{U}(s_t, a_t)$ is an estimation to the LCB-penalty $\Gamma^{\text{lcb}}(s_t, a_t)$ for an appropriately selected tuning parameter β_t. As a result, our method enjoys a similar form of optimality gap in PEVI.

Further, since our method adopts additional OOD sampling and smooth constraints, the covariance matrix in calculating $\Gamma^{\text{lcb}}(s_t, a_t; \mathcal{D}_{\text{RO2O}})$ for our method becomes

$$\Gamma^{\text{lcb}}(s_t, a_t; \mathcal{D}_{\text{RO2O}}) = \beta_t [\phi(s_t, a_t)^\top (\widetilde{\Lambda}_t)^{-1} \phi(s_t, a_t)]^{\frac{1}{2}},$$

where

$$\tilde{\Lambda}_t = \tilde{\Lambda}_t^{\text{in}} + \tilde{\Lambda}_t^{\text{ood}} + \tilde{\Lambda}_t^{\text{robust}},$$

which also serves as a ξ-uncertainty quantifier. Then the uncertainty term for RO2O is $\Gamma_i^{\text{lcb}}(s_t, a_t; \mathcal{D}_{\text{RO2O}})$ in offline setting, and in online exploration it becomes $\Gamma_i^{\text{LCB}}(s_t, a_t; \mathcal{D}_{\text{RO2O}} \cup \mathcal{D}_{\text{online}})$. Further, our theoretical analysis in Theorem 2 shows that

$$\text{SubOpt}(\pi^*, \widetilde{\pi}) \leq \sum\nolimits_{t=1}^{T} \mathbb{E}_{\pi^*} \left[\Gamma_i^{\text{lcb}}(s_t, a_t; \mathcal{D}_{\text{RO2O}} \cup \mathcal{D}_{\text{online}}) \right] \leq \sum\nolimits_{t=1}^{T} \mathbb{E}_{\pi^*} \left[\Gamma_i^{\text{lcb}}(s_t, a_t; \mathcal{D}_{\text{RO2O}}) \right],$$

which signifies the online exploration data can consistently reduce the sub-optimality gap of our method with ξ-uncertainty quantification.

Appendix B. Environmental Settings

In this section, we introduce more details of the experimental environments.

MuJoCo Locomotion We conduct experiments on three MuJoCo locomotion tasks from D4RL(Fu et al., 2020), namely HalfCheetah, Walker2d, and Hopper. The goal of each task is to move forward as far as possible without falling, while keeping the control cost minimal. For each task, we consider three types of datasets. The medium datasets contain trajectories collected by medium-level policies. The medium-replay datasets encompass all samples collected during the training of a medium-level agent from scratch. In the case of the medium-expert datasets, half of the data comprises rollouts from medium-level policies, while the other half consists of rollouts from expert-level policies. In this study, we exclude the random datasets, as in typical real-world scenarios, we rarely use a random policy for system control. We utilize the v2 version of each dataset. For offline phase, We train agents for 2.5M gradient steps over all datasets with an ensemble size of $N = 10$. Then we run online fine-tuning for an additional 250K environment interactions.

Antmaze Navigation We also evaluate our method on the Antmaze navigation tasks that involve controlling an 8-DoF ant quadruped robot to navigate through mazes and reach a desired goal. The agent receives binary rewards based on whether it successfully reaches the goal or not. We study each method using the following datasets from D4RL (Fu et al., 2020): large-diverse, large-play, medium-diverse, medium-play, umaze-diverse, and umaze. The difference between diverse and play datasets is the optimality of the trajectories they

contain. The diverse datasets consist of trajectories directed towards random goals from random starting points, whereas the play datasets comprise trajectories directed towards specific locations that may not necessarily correspond to the goal. We use the v2 version of each dataset. For offline phase, We train agents for 1M gradient steps over all datasets with an ensemble size of $N = 10$. Then we run online fine-tuning for an additional 250K environment interactions.

Appendix C. Implementation Details

In this section, we introduce implementation details and hyper-parameters for each task.

MuJoCo Locomotion We select PEX, AWAC, SAC, SPOT, Cal-QL and IQL as our baselines in mujoco locomotion tasks. For SAC, AWAC, SPOT, Cal-QL and IQL, we use the implementation from CORL[1] with default hyperparameters. For PEX, we use the open-source code of the original paper[2]. To compare the fine-tuning performance of the algorithms under limited online interactions, we reduce the number of online interaction steps from the previous 1M to 250K. All the hyper-parameters used in RO2O for the benchmark experiments are listed in Table 4. η_1, η_2, η_3 indicate the coefficient of the Q-network smoothing loss $\mathcal{L}_{\text{Qsmooth}}$, ood loss $\mathcal{L}_{\text{ood}}$ and the policy smoothing loss $\mathcal{L}_{\text{policy}}$, respectively, where η_1 maintains a constant value of 0.0001, η_2 is tuned within $\{0.0, 0.1, 0.5\}$ and η_3 is searched in $\{0.1, 1.0\}$. Additionally, for the above three losses, we construct a perturbation set $\mathbb{B}_d(s, \epsilon) = \{\hat{s} : d(s, \hat{s}) \leq \epsilon\}$ by setting different epsilons ϵ. We denote the perturbation scales for the Q value functions, the policy, and the OOD loss as ϵ_Q, ϵ_P, ϵ_{ood}. τ is set to control the weight of $\mathcal{L}_{\text{Qsmooth}}$ which maintains a constant value of 0.2. The number of sampled perturbed observations n is set for tuning within $\{10, 20\}$. And α is set to control the pessimistic degree of $\mathcal{L}_{\text{ood}}$ during the pre-trained phase. Moreover, discarding offline data buffer is adopted in RO2O, which exhibits benefits for stable transfer in our experiments and mitigates the distributional shift.

Table 4: Hyperparameters of RO2O for the MuJoCo domains.

Task Name	η_1	η_2	η_3	ϵ_{Q}	ϵ_{P}	ϵ_{ood}	τ	n	α
halfcheetah-medium	0.0001	0.0	0.1	0.001	0.001	0.00	0.2	10	0
halfcheetah-medium-replay				0.001	0.001			10	
halfcheetah-medium-expert				0.001	0.001			10	
halfcheetah-expert				0.005	0.005			10	
hopper-medium	0.0001	0.5	0.1	0.005	0.005	0.01	0.2	20	$2.0 \rightarrow 0.1$ $(1e^{-6})$
hopper-medium-replay									$0.1 \rightarrow 0.0$ $(1e^{-6})$
hopper-medium-expert									$3.0 \rightarrow 1.0$ $(1e^{-6})$
hopper-expert									$4.0 \rightarrow 1.0$ $(1e^{-6})$
walker2d-medium	0.0001	0.1	1.0	0.01	0.01	0.01	0.2	20	$1.0 \rightarrow 0.1$ $(5e^{-7})$
walker2d-medium-replay		0.1		0.01	0.01				$0.1 \rightarrow 0.1$ (0.0)
walker2d-medium-expert		0.1		0.01	0.01				$0.1 \rightarrow 0.1$ (0.0)
walker2d-expert		0.5		0.005	0.005				$1.0 \rightarrow 0.5$ $(1e^{-6})$

Antmaze Navigation We select PEX, SPOT and Cal-QL as our baselines in antmaze navigation tasks. For SPOT and Cal-QL, we use the implementation provided by CORL with default hyperparameters. We directly used the experimental results provided by CORL

1. https://github.com/tinkoff-ai/CORL
2. https://github.com/Haichao-Zhang/PEX

in weight & bias for comparison. For PEX, we use the open-source code of the original paper. To compare the fine-tuning performance of the algorithms under limited online interactions, we reduce the number of online interaction steps from the previous 1M to 250K. We found that incorporating behavior cloning (BC) during the offline pre-training phase of the AntMaze task can effectively improve model performance. Additionally, making appropriate adjustments to BC during the online fine-tuning phase for certain tasks can also enhance the algorithm's performance and stability. And we transform AntMaze rewards according to $4(r-0.5)$ as per MSG(Ghasemipour et al., 2022) or CQL(Kumar et al., 2020). All the hyper-parameters used in RO2O for the benchmark experiments are listed in Table 5. $\beta_{\text{BC, off}}$ and $\beta_{\text{BC, on}}$ indicate the weight of BC regularization on policy networks during offline pre-training and online fine-tuning, respectively. The LCB policy objective and 'Min' policy objective represent optimizing the policy network using $\text{Mean}(\text{Q}_{\theta_\text{i}}(\text{s},\text{a})) - \text{Std}(\text{Q}_{\theta_\text{i}}(\text{s},\text{a}))$ or $\min_i Q_{\theta_i}(s,a)$, respectively. And the meanings of other parameters remain consistent with Table 4 under the Mujoco tasks.

Table 5: Hyper-parameters of RO2O for the AntMaze domains.

Task Name	η_2	η_3	ϵ_P	ϵ_ood	n	policy objective	$\beta_{\text{BC, off}}$	$\beta_{\text{BC, on}}$	α
umaze	1.0	0.3	0.005	0.01	20	LCB	5	5	$1.0 \rightarrow 1.0$ (0.0)
umaze-diverse		0.3				LCB	10	20	$2.0 \rightarrow 2.0$ (0.0)
medium-play		0.3				LCB	2	2	$1.0 \rightarrow 1.0$ (0.0)
medium-diverse		0.3				LCB	4	4	$2.0 \rightarrow 1.0$ ($1e^{-6}$)
large-play		0.5				Min	2	8	$2.0 \rightarrow 1.0$ ($1e^{-6}$)
large-diverse		0.3				Min	2	8	$1.0 \rightarrow 1.0$ (0.0)

Appendix D. More Discussion

Using different learning target for different tasks Most of the ensemble-based RL algorithms use shared pessimistic target values when computing each ensemble member's Bellman error. However, the results reported in the reference (Ghasemipour et al., 2022; Yang et al., 2022) and our experiments demonstrate that using independent target surpasses highly well-tuned state-of-the-art methods by a wide margin on challenging domains such as Antmaze. We believe there are several reasons: (i) Antmaze navigation tasks are more complex than Mujoco locomotion tasks. Since there is significant distribution shift between online interactions and offline data, it will be challenging to learn effective policies by relying solely on policies derived from offline data. Due to shared TD target is too pessimistic, agents tend to avoid accessing a significant number of ODD samples, thereby limiting exploration to some extent. This also results in methods like PBRL (Bai et al., 2022) and EDAC (An et al., 2021), which utilize shared TD targets, performing poorly on tasks such as Antmaze. (ii) In contrast, the disparity between in-distribution and OOD policies is not obvious in Mujoco tasks. Therefore, directly applying shared targets to achieve pessimistic updates in Mujoco tasks ensures pessimism while also capturing the uncertainty near the distribution. Therefore, we refer to the independent target used in the Q-value Bellman update by MSG (Ghasemipour et al., 2022).

Comparison to RORL Here, we discuss the differences between RORL (Yang et al., 2022) and our method from several perspectives. (i) **Motivation.** The motivation of robustness constraints in RORL is to improve the smoothness of policy and Q-functions in

facing adversarial attacks. In contrast, our method focuses on offline-to-online settings, where robustness regularization is used to prevent the distribution shift of OOD data in online exploration. We highlight that both RORL and our method adopt the same smooth value function/policy originally proposed in online exploration, while the motivations for introducing robustness in our method and RORL are quite different. (ii) From a **theoretical** perspective, we provide new theoretical results in Theorem 2, which analyzes the optimality gap of the learned policy in online exploration with additional online datasets. With uncertainty quantification and smoothness constraints, our method benefits from more online exploration data without suffering from distribution shifts, which is crucial for offline-to-online RL. Our theoretical result shows the optimality gap of our method shrinks if the online exploration data increases the data coverage of the optimal policy, which is significantly different from previous offline-to-online methods that should penalize OOD data in online exploration. (iii) **Empirically**, our method obtains strong performance without a specially designed online adaptation process. The offline-to-online performance does not drop when interacting with the online environment, which is consistent with our theoretical analysis. Benefiting from the theoretical result, our method can perform efficient policy improvement without specific modifications to the learning architecture in the offline-to-online process.

References

An, G., Moon, S., Kim, J.-H., & Song, H. O. (2021). Uncertainty-based offline reinforcement learning with diversified Q-ensemble. *Advances in neural information processing systems*, *34*, 7436–7447.

Bai, C., Wang, L., Yang, Z., Deng, Z.-H., Garg, A., Liu, P., & Wang, Z. (2022). Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. In *International Conference on Learning Representations*.

Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J., Petrov, M., de Oliveira Pinto, H. P., Raiman, J., Salimans, T., Schlatter, J., Schneider, J., Sidor, S., Sutskever, I., Tang, J., Wolski, F., & Zhang, S. (2019). Dota 2 with large scale deep reinforcement learning. *CoRR*, *abs/1912.06680*.

Chen, R. Y., Sidor, S., Abbeel, P., & Schulman, J. (2017). Ucb exploration via Q-ensembles. *CoRR*, *abs/1706.01502*.

Chen, X., Wang, C., Zhou, Z., & Ross, K. W. (2021). Randomized ensembled double Q-learning: Learning fast without a model. In *International Conference on Learning Representations*.

Fu, J., Kumar, A., Nachum, O., Tucker, G., & Levine, S. (2020). D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, *abs/2004.07219*.

Fujimoto, S., & Gu, S. S. (2021). A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, *34*, 20132–20145.

Fujimoto, S., Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR.

Fujimoto, S., Meger, D., & Precup, D. (2019). Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR.

Ghasemipour, K., Gu, S. S., & Nachum, O. (2022). Why so pessimistic? estimating uncertainties for offline RL through ensembles, and why their independence matters. *Advances in Neural Information Processing Systems*, *35*, 18267–18281.

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., & Silver, D. (2018). Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

Jin, C., Yang, Z., Wang, Z., & Jordan, M. I. (2020). Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pp. 2137–2143. PMLR.

Jin, Y., Yang, Z., & Wang, Z. (2021). Is pessimism provably efficient for offline RL?. In *International Conference on Machine Learning*, pp. 5084–5096. PMLR.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., & Pérez, P. (2021). Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, *23*(6), 4909–4926.

Kostrikov, I., Nair, A., & Levine, S. (2022). Offline reinforcement learning with implicit Q-learning. In *ICLR*. OpenReview.net.

Kumar, A., Zhou, A., Tucker, G., & Levine, S. (2020). Conservative Q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, *33*, 1179–1191.

Lambert, N., Wulfmeier, M., Whitney, W., Byravan, A., Bloesch, M., Dasagi, V., Hertweck, T., & Riedmiller, M. (2022). The challenges of exploration for offline reinforcement learning. *CoRR*, *abs/2201.11861*.

Lan, Q., Pan, Y., Fyshe, A., & White, M. (2020). Maxmin Q-learning: Controlling the estimation bias of Q-learning. *CoRR*, *abs/2002.06487*.

Lange, S., Gabel, T., & Riedmiller, M. (2012). Batch reinforcement learning. In *Reinforcement learning: State-of-the-art*, pp. 45–73. Springer.

Lee, K., Laskin, M., Srinivas, A., & Abbeel, P. (2021). Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 6131–6141. PMLR.

Lee, S., Seo, Y., Lee, K., Abbeel, P., & Shin, J. (2022). Offline-to-online reinforcement learning via balanced replay and pessimistic Q-ensemble. In *Conference on Robot Learning*, pp. 1702–1712. PMLR.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529–533.

Nair, A., Gupta, A., Dalal, M., & Levine, S. (2020). AWAC: Accelerating online reinforcement learning with offline datasets. *CoRR*, *abs/2006.09359*.

Nakamoto, M., Zhai, Y., Singh, A., Mark, M. S., Ma, Y., Finn, C., Kumar, A., & Levine, S. (2023). Cal-QL: Calibrated offline RL pre-training for efficient online fine-tuning. *CoRR*, *abs/2303.05479*.

Osband, I., Blundell, C., Pritzel, A., & Van Roy, B. (2016). Deep exploration via bootstrapped DQN. *Advances in neural information processing systems*, *29*.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., & Dormann, N. (2021). Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, *22*(268), 1–8.

Schneegass, D., Udluft, S., & Martinetz, T. (2008). Uncertainty propagation for quality assurance in reinforcement learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 2588–2595. IEEE.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *CoRR*, *abs/1707.06347*.

Schweighofer, K., Dinu, M.-c., Radler, A., Hofmarcher, M., Patil, V. P., Bitto-Nemling, A., Eghbal-zadeh, H., & Hochreiter, S. (2022). A dataset perspective on offline reinforcement learning. In *Conference on Lifelong Learning Agents*, pp. 470–517. PMLR.

Shen, Q., Li, Y., Jiang, H., Wang, Z., & Zhao, T. (2020). Deep reinforcement learning with robust and smooth policy. In *International Conference on Machine Learning*, pp. 8707–8718. PMLR.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, *362*(6419), 1140–1144.

Sinha, S., Mandlekar, A., & Garg, A. (2022). S4rl: Surprisingly simple self-supervision for offline reinforcement learning in robotics. In *Conference on Robot Learning*, pp. 907–917. PMLR.

Swazinna, P., Udluft, S., & Runkler, T. (2021). Overcoming model bias for robust offline deep reinforcement learning. *Engineering Applications of Artificial Intelligence*, *104*, 104366.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE.

Uchendu, I., Xiao, T., Lu, Y., Zhu, B., Yan, M., Simon, J., Bennice, M., Fu, C., Ma, C., Jiao, J., et al. (2023). Jump-start reinforcement learning. In *International Conference on Machine Learning*, pp. 34556–34583. PMLR.

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE.. *Journal of machine learning research*, *9*(11).

Wu, J., Wu, H., Qiu, Z., Wang, J., & Long, M. (2022). Supported policy optimization for offline reinforcement learning. *Advances in Neural Information Processing Systems*, *35*, 31278–31291.

Wu, Y., Tucker, G., & Nachum, O. (2019). Behavior regularized offline reinforcement learning. *CoRR*, *abs/1911.11361*.

Yang, L., & Wang, M. (2019). Sample-optimal parametric Q-learning using linearly additive features. In *International Conference on Machine Learning*, pp. 6995–7004. PMLR.

Yang, R., Bai, C., Ma, X., Wang, Z., Zhang, C., & Han, L. (2022). RORL: Robust offline reinforcement learning via conservative smoothing. *Advances in Neural Information Processing Systems*, *35*, 23851–23866.

Yu, C., Liu, J., Nemati, S., & Yin, G. (2021). Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, *55*(1), 1–36.

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., & Ma, T. (2020). MOPO: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, *33*, 14129–14142.

Zhang, H., Xu, W., & Yu, H. (2023). Policy expansion for bridging offline-to-online reinforcement learning. In *The Eleventh International Conference on Learning Representations*.

Zhao, K., Ma, Y., Liu, J., Jianye, H., Zheng, Y., & Meng, Z. (2023). Improving offline-to-online reinforcement learning with Q-ensembles. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*.

Zhao, Y., Boney, R., Ilin, A., Kannala, J., & Pajarinen, J. (2022). Adaptive behavior cloning regularization for stable offline-to-online reinforcement learning. In *ESANN*.