A Perspective of Q-value Estimation on Offline-to-Online Reinforcement Learning

Yinmin Zhang[1, 2 *], **Jie Liu**[2, 3 *], **Chuming Li**[1, 2 *], **Yazhe Niu**[2], **Yaodong Yang**[4†], **Yu Liu**[2†], **Wanli Ouyang**[2]

[1] The University of Sydney, SenseTime Computer Vision Group, Australia
[2] Shanghai Artificial Intelligence Laboratory
[3] Multimedia Laboratory, The Chinese University of Hong Kong
[4] Institute for AI, Peking University
yinmin.zhang@sydney.edu.au, jieliu@ie.cuhk.edu.hk, chli3951@uni.sydney.edu.au, yaodong.yang@pku.edu.cn, liuyuisanai@gmail.com, ouyangwanli@pjlab.org.cn

Abstract

Offline-to-online Reinforcement Learning (O2O RL) aims to improve the performance of offline pretrained policy using only a few online samples. Built on offline RL algorithms, most O2O methods focus on the balance between RL objective and pessimism, or the utilization of offline and online samples. In this paper, from a novel perspective, we systematically study the challenges that remain in O2O RL and identify that the reason behind the slow improvement of the performance and the instability of online finetuning lies in the inaccurate Q-value estimation inherited from offline pretraining. Specifically, we demonstrate that the estimation bias and the inaccurate rank of Q-value cause a misleading signal for the policy update, making the standard offline RL algorithms, such as CQL and TD3-BC, ineffective in the online finetuning. Based on this observation, we address the problem of Q-value estimation by two techniques: (1) perturbed value update and (2) increased frequency of Q-value updates. The first technique smooths out biased Q-value estimation with sharp peaks, preventing early-stage policy exploitation of sub-optimal actions. The second one alleviates the estimation bias inherited from offline pretraining by accelerating learning. Extensive experiments on the MuJoco and Adroit environments demonstrate that the proposed method, named SO2, significantly alleviates Q-value estimation issues, and consistently improves the performance against the state-of-the-art methods by up to 83.1%.

Introduction

In recent years, deep offline Reinforcement Learning (RL) (Yu et al. 2020; Agarwal, Schuurmans, and Norouzi 2020; Bai et al. 2021) has received increasing attention due to its potential for leveraging abundant logged data or expert knowledge (*e.g.*, human demonstrations) to learn high-quality policies. Analogous to trends in computer vision (He et al. 2021; Radford et al. 2021; Bommasani et al. 2021) and natural language processing (Devlin et al. 2019; Brown et al. 2020), where powerful models pretrained on large, diverse datasets generalize to task-specific data through finetuning, *offline-to-online reinforcement learning* (O2O RL) aims to enhance the performance of a pretrained offline RL policy via finetuning on only a few online collected samples. In O2O RL, *the central challenge is to maximize the efficient utilization of additional online samples for further performance improvement.*

We find that directly finetuning offline RL policies with online interactions remains inefficient in many cases, as shown in Figure 2a. Most existing works (Zhao et al. 2022; Lee et al. 2022) assume that the above issue results from the state-action distribution shift between the offline and online samples. To handle this distribution shift, they propose different methods to progressively transfer from offline to online finetuning settings, including the balance between RL objective and pessimism (Zhao et al. 2022) and between the usage of offline and online samples (Lee et al. 2022).

The main contribution of this paper is to offer a comprehensive understanding of the problem in O2O RL, focusing on **Q-value estimation**, a perspective that has been underexplored. While prior research acknowledges this issue, our work delves deeper into it, unveiling two key challenges: (1) *biased Q-value estimation* and (2) *inaccurate rank of Q-value estimation*, which represents the distinguishability in the quality of different state-action pairs. Despite most offline RL policies trained in pessimistic manners such as conservative Q-learning (Kumar et al. 2020), action constraints (Fujimoto and Gu 2021), and Q-ensemble (An et al. 2021), we still observe severe overestimation (CQL and TD3-BC) or underestimation (EDAC) compared to an online RL policy trained from scratch with similar performance. These disparities result in in a severe bootstrap error, as shown in Figure 2b. Moreover, it is well-known that the inaccurate rank of Q-value estimation, such as inaccurate advantage, results in misleading signals to policy updates. To quantitatively assess the accuracy of Q-value estimation rank, we compute Kendall's τ coefficient between the estimated and true Q-values, separately for offline RL and online RL policies. Figure 2c shows that offline RL methods (CQL, TD3-BC, and EDAC) have lower Kendall's τ coefficient compared to the online RL, SAC. This indicates that the offline RL is less accurate in ranking estimated Q-values than the online RL. Consequently, the signal from the inaccurate Q-value estimation used in pol-

*Equal contribution. Author ordering is determined by coin flip.
†Corresponding author

icy updates is misleading, which causes instability and slow improvement of performance during online finetuning.

To mitigate the Q-value estimation problems stated above, we identify two widely used techniques: (1) perturbed value update and (2) increased frequency of Q-value updates. Specifically, we perturb the target action with an extra noise to smooth out the biased Q-value estimation with sharp peaks. This prevents the agent from overfitting to a specific action that might have worked in the past but might not generalize well to new situations. By incorporating noisy actions, the agent is encouraged to explore different actions in the next state, reducing overestimation bias. This encourages the agent to consider a range of plausible actions rather than fixating on a single seemingly optimal action. Consequently, it leads to more accurate value estimates, mitigating value estimation bias in online RL. Additionally, we remark that the biased Q-value estimation requires an increased frequency of updates to converge rapidly to a normal level. Increasing the update frequency for Q-values and policies makes the learning process more responsive to new experiences, resulting in more accurate value estimates and faster convergence towards the optimal policy. As illustrated in Figure 1, both of them provide more accuracy of Q-value function estimation to further improve performance during online finetuning.

The proposed **S**imple method for **O2O** RL, named SO2, is evaluated on (1) MuJoCo locomotion tasks, and (2) dexterous manipulation tasks, using an offline policy pretrained on the D4RL. Our experiments highlight that SO2 consistently outperforms prior state-of-the-art methods in sample efficiency and asymptotic performance on various tasks and datasets.

Related Work

Offline RL Offline RL employs static datasets to train agents but faces Q-value overestimation challenges, especially with unseen state-action pairs (Fujimoto, Meger, and Precup 2019). Solutions can be categorized as follows: (1) Policy constraint methods (Nair et al. 2020; Fujimoto and Gu 2021; Wu, Tucker, and Nachum 2019) enforce policy alignment with the behavior policy in logged datasets, minimizing policy distribution shift through behavior cloning and KL-divergence; (2) Pessimistic value methods (Kumar et al. 2020) regularize the Q-value to mitigate the overestimation by adding a conservative penalty to the RL object; (3) Uncertainty-based methods (An et al. 2021; Kumar et al. 2019; Bai et al. 2021) measure the distribution shift using Q-value uncertainty to guarantee a robust policy update.

Online RL with offline datasets. Several works (Ijspeert, Nakanishi, and Schaal 2002; Kim et al. 2013; Zhu et al. 2019; Nair et al. 2017; Theodorou, Buchli, and Schaal 2010; Gupta et al. 2020) improve online RL by utilizing offline datasets, assuming these datasets contain optimal demonstrations specific to the current environment. However, real-world offline datasets, sourced from various sources, contain numerous sub-optimal demonstrations. Consequently, these methods may not perform well in such scenarios due to estimation bias from sub-optimal datasets. An alternative approach (Nair et al. 2020; Lee et al. 2022) takes a pessimistic view, avoiding the assumption of optimality. These methods focus on the challenge of significant distribution gaps between offline datasets and online samples during O2O finetuning. Particularly, AWAC (Nair et al. 2020) enforces an implicit constraint on policy updates to align the learned policy with policies in both offline and online samples. (Lee et al. 2022) employs Q-ensemble and balanced replay, encouraging the use of near-on-policy samples from offline demonstrations using neural network predictions. AdaptiveBC (Zhao et al. 2022) proposes a randomized Q-function ensemble, dynamically balancing the RL objective with behavior cloning based on agent performance and training stability. IQL (Kostrikov, Nair, and Levine 2021) approximates each state's upper bound of Q-values and extracts the policy from the Q-value estimation. ODT (Zheng, Zhang, and Grover 2022) proposes an O2O RL algorithm using sequence modeling and entropy regularizers to unify pertaining and finetuning. PEX (Zhang, Xu, and Yu 2022) introduces a policy expansion approach by incorporating offline policies for additional learning. QDagger (Agarwal et al. 2022) focuses on optimizing knowledge transfer efficiency within the reincarnating RL paradigm. While some prior works do acknowledge inaccurate Q-value estimation, they may not have explicitly dissected the various types of inaccuracies and their specific effects on policy updates and online finetuning. Our research extends the existing literature by providing a more comprehensive understanding of these issues within O2O RL. Specifically, we focus on uncovering the challenges related to biased Q-value estimation and inaccurate Q-value ranking inherited from offline RL. These issues result in unreliable signals that exacerbate bootstrap errors during finetuning, leading to instability and suboptimal policy updates. Our investigation and proposed solutions address problems not explored in-depth in previous research.

Background

RL seeks to learn a policy maximizing the cumulative discounted reward in the Markov Decision Process (MDP) defined by $(\mathcal{S}, \mathcal{A}, \mathcal{R}, p, \gamma)$, with the state space $\mathcal{S}$, the action space $\mathcal{A}$, reward function $\mathcal{R}$, transition dynamics p, and discount factor $\gamma \in (0, 1)$. Actor-critic algorithms maintain the Q-value function $Q(\mathbf{s}, \mathbf{a})$ and the learned policy $\pi(\mathbf{a} \mid \mathbf{s})$.

Offline RL. Offline RL assumes the agent leverages the logged datasets without any interaction with the environment. The agent is trained in a logged dataset $\mathcal{D}$ which is sampled from the environment by a behavior policy $\pi_\beta(\mathbf{a} \mid \mathbf{s})$. For the actor-critic algorithm given dataset $\mathcal{D} = \{(\mathbf{s}, \mathbf{a}, r, \mathbf{s}')\}$, we can represent value iteration and policy improvement by:

$$Q^{k+1} \leftarrow \arg\min_Q \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}} \left[\left(\mathcal{B}^{\pi^k} Q^k(\mathbf{s}, \mathbf{a}) - Q(\mathbf{s}, \mathbf{a}) \right)^2 \right], \tag{1}$$

$$\pi^{k+1} \leftarrow \arg\max_\pi \mathbb{E}_{\mathbf{s}\sim\mathcal{D}, \mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})} \left[Q^{k+1}(\mathbf{s}, \mathbf{a}) \right], \tag{2}$$

where the $\mathcal{B}^\pi$ is the bellman operator on the fixed dataset following the learned policy $\pi(\mathbf{a} \mid \mathbf{s})$, $\mathcal{B}^\pi Q(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\mathbf{s}'\sim p(\mathbf{s},\mathbf{a})} \left[r + \gamma \mathbb{E}_{\mathbf{a}'\sim\pi(\mathbf{a}'|\mathbf{s}')} \left[\hat{Q}(\mathbf{s}', \mathbf{a}') \right] \right]$.

Online RL. Following a given policy $\pi(\mathbf{a} \mid \mathbf{s})$, the objective of online RL is the expected cumula-

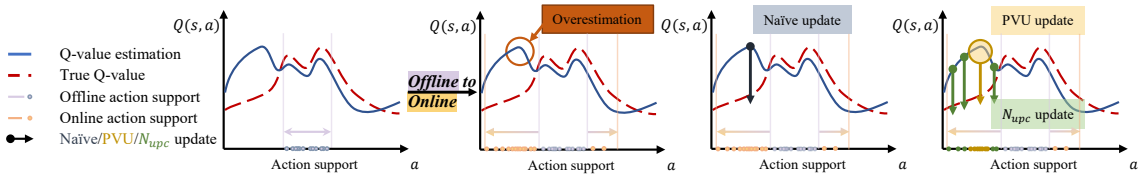

Figure 1: Comparison of SO2 against baseline methods. The orange and purple spots represent offline and online dataset samples, respectively. The two figures on the left exhibit that from offline to online the expanding action support introduces unseen samples with the biased Q-value. On the right side, we illustrate the basic idea of SO2 compared with the naive update. Specifically, Perturbed Value Update (PVU) (yellow) provides indirect range-based regularization leading to smoothed Q-value estimation. The N_{upc} green provides more frequency of Q-value updates after each online collection. Both of them further improve the accuracy of Q-value estimation during online finetuning.

tive discounted reward following the policy in the MDP formulated by $\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\sum_{t=0}^{\infty}\gamma^t r_t$. Q-Learning methods maintain a Q-function $Q^\pi(\mathbf{s},\mathbf{a})$ to estimate $\mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r_t \mid \mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a}\right]$, which measures the expected discounted return following a given policy $\pi(\mathbf{a}\mid\mathbf{s})$.

O2O RL. In O2O RL, agents aim to enhance performance by finetuning with online-collected samples. This process involves two phases: (1) offline training, where logged datasets are used to pretrain the policy, and (2) online finetuning, where additional samples refine the pretrained policy.

Q-Value Estimation Issues in O2O RL

In this section, we systematically study the issues in O2O RL by focusing on Q-value estimation. We begin by assessing the performance of offline RL methods in O2O scenarios and subsequently pinpoint the root cause as Q-value estimation.

Performance. We evaluate the performance of both standard online RL methods and offline RL methods with online finetuning. The online RL methods includes SAC and the online variant of EDAC, which are trained without any offline dataset. Their curves are labeled with star marks in Figure 2a. The offline RL methods include policy constraints (TD3-BC), conservative Q-value learning (CQL), and Q-ensemble (EDAC) in the O2O RL setting. Additionally, we evaluate the loose variant of those methods by removing the pessimistic constraints during online finetuning after offline pretraining. The performance of standard online RL methods and offline RL methods is averaged on three environments (HalfCheetah, Hopper, and Walker2D). Particularly, for offline RL, each environment has four datasets with different qualities in D4RL benchmark and generates four pretrained policies, resulting total 12 policies to average on. It is observed in Figure 2a that the improvement of CQL, EDAC, TD3-BC and the loose variant of CQL and EDAC is slow in online finetuning; and the asymptotic performance of the loose variant of TD3-BC is significantly worse than online RL and its initial performance. In summary, while offline algorithms (CQL, TD3-BC and EDAC) all achieve excellent performance in offine settings, all of them achieve slow performance improvement or even performance degradation during online finetuning.

Normalized difference of the Q-value. We use normalized difference, also called percent difference in (Fujimoto and Gu 2021), to measure the difference between the estimated Q-value and the true Q-value, formalized by $\frac{Q^{estimated}-Q^{true}}{Q^{true}}$. Q^{true} is computed based on actual returns obtained along an adequately extended trajectory collected by the current policy, providing an accurate reflection of the true Q-value. To highlight the difference between online RL and offline RL algorithms, the normalized differences of offline RL algorithms are subtracted by the normalized difference of the online RL baseline, SAC (Haarnoja et al. 2018). Therefore, a positive normalized difference means that the Q-value estimation of the offline RL algorithm tends to overestimate, compared to the online baseline, and vice versa. Figure 2b demonstrates the subtracted normalized difference of CQL, TD3-BC and EDAC. While offline algorithms deal with the out-of-distribution problem via a pessimistic perspective, TD3-BC and CQL usually overestimate the Q-value for the unseen state-action pairs collected online. Conversely, EDAC faces the issue of Q-value estimation bias, notably leading to significant Q-value underestimation when compared to online reinforcement learning algorithms. This Q-value estimation bias propagates through the Bellman Equation, resulting in suboptimal policy updates or, in some cases, divergence. Specifically, excessively high Q-value estimates can lead to the collapse of Q-value, while overly low Q-value estimates can impede progress, causing slow improvement.

Kendall's τ coefficient over Q-value. Kendall's τ coefficient measures the rank correlation between two sets of variables. To assess the precision of the rank-ordering of estimated Q-values, we employ Kendall's τ coefficient to compare them with true Q-values. We evaluate this metric on policies pretrained by CQL, TD3-BC, and EDAC, as well as online SAC, all of which exhibit comparable performance. This evaluation follows a structured procedure: (1) Roll out with each pretrained policy to collect multiple episodes of state-action pairs; (2) Select collections of pairs within each episode using a sliding window approach, denoting each collection as P_i, where $i \in [1, ..., M]$ and M represents the window number (10, as used in reported results); (3) Compute both the estimated Q-value and true Q-value of all state-action pairs within each P_i; (4) Compute Kendall's ratio K_i for each collection, and derive the final metric K as the average, given by $K = \frac{1}{M}\sum K_i$. Figure 2c exhibits the K for each RL algorithm. The results reveal that offline RL

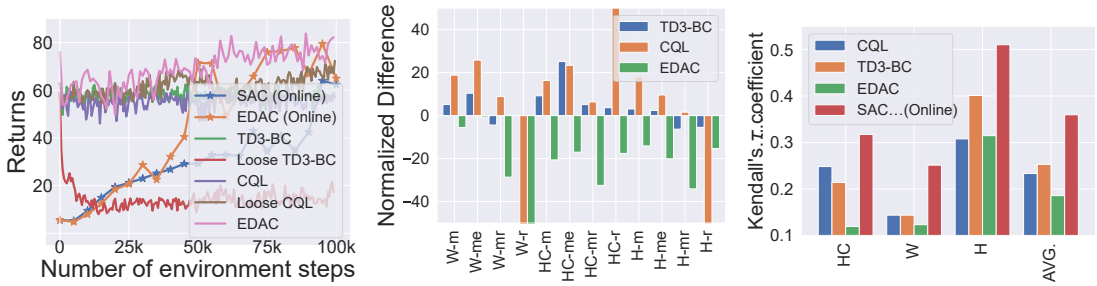

Figure 2: (a) Learning curve. (b) Normalized difference in the Q-value estimation between offline RL algorithms and an online RL baseline. A positive normalized difference means that the Q-value estimation of the offline RL algorithm tends to overestimate, compared to online RL, and vice versa. While offline algorithms deal with the out-of-distribution problem via a pessimistic perspective, TD3-BC and CQL still overestimate the Q-value for the state-action pairs collected by the corresponding trained policy with noise. (c) Kendall's τ coefficient measures the correlation between the Q-value estimation and the true Q-value on the same batch of state-action pairs. The lower coefficient shows that the rank of Q-value estimation from offline RL algorithms is severely worse than online RL algorithms, which leads to unstable training and slow improvement. HC = HalfCheetah, H = Hopper, W = Walker, r = random, m = medium, mr = medium-replay, me = medium-expert, AVG.=average.

algorithms, CQL, TD3-BC, and EDAC, exhibit significantly lower K values compared to the online RL algorithm SAC. This suggests that the estimated Q-values are not accurate in assessing the relative qualities of different state-action pairs.

Summary. While these offline algorithms (CQL, TD3-BC, and EDAC) achieve outstanding performance after pretraining and provide great initial actors for online finetuning, all of them improve slowly with high variances in different environments with different settings, due to the biased estimation and inaccurate rank of Q-value. Thus, how to obtain an accurate Q-value estimation is the bottleneck of O2O RL.

A Simple but Effective Approach to O2O RL

Algorithmic Details

To provide proper guidance for online policy updates, we present an O2O RL approach that improves the accuracy of Q-value estimation. Our algorithm builds on top of the SAC with Q-ensemble, with two modifications, Perturbed Value Update, and Increased Frequency of Q-value Update.

Perturbed Value Update (PVU). We modify the update step of the ensemble Q-value, as follows:

$$\begin{gathered}\mathcal{T}Q\phi_i(\mathbf{s},\mathbf{a}) \leftarrow r+\gamma\left(\hat{Q}_{\phi_i}(\mathbf{s}',\mathbf{a}'+\epsilon)-\beta\log\pi\left(\mathbf{a}'\mid\mathbf{s}'\right)\right), \\ \mathbf{a}'\sim\pi\left(\cdot\mid\mathbf{s}'\right), \\ \epsilon\sim\operatorname{clip}(\mathcal{N}(0,\sigma),-c,c),\ i=1,\ldots,N_{ensemble},\end{gathered} \tag{3}$$

where ϵ represents action noise with variance σ and bounded by c. ϕ_i is the parameter of the i^{th} Q network in the ensemble method, with a total of $N_{ensemble}$ networks. This perturbation enlarges the action distribution used for estimating the target Q-value, resulting in a smoother Q-value estimate. Such smooth estimation prevents extremely exploiting the state on which the action has biased Q-value estimation, often seen as sharp peaks. Hence, this perturbation encourages policy exploration of state-action pairs, even when they have low estimated Q-values inherited from the offline RL pretraining, and further updates the Q-value estimation of these pairs to find potential strategies with high returns.

Increased Frequency of Q-value Update. To alleviate the inaccurate Q-value estimation in offline RL, as discussed in Sec. , we increase the update frequency of the Q-value function after each online collection. Specifically, we denote the update frequency after each online sample collection as N_{upc}, where upc means update per collection.

Implementation Details

We describe the implementation of SO2 based on SAC with Q-ensemble in DI-engine, a DRL framework utilized by various methods (Li et al. 2023b,a; Liu et al. 2023; Wang et al. 2023). The pseudocode is shown in Algorithm 1, with differences from Q-ensemble SAC in Lines 5-8. After each online collect iteration, we sample a mini-batch $\mathcal{B}$ from the replay buffer. This buffer contains both online and extensive offline dataset samples, enhancing training stability. Subsequently, we perform N_{upc} updates to the Q-value network.

Experiments

Our experiments aim to investigate the following concerns:

- **Performance:** Can our method improve the performance compared to existing O2O RL approaches and online RL approaches trained from scratch (see Figure 3)?
- $\boldsymbol{N_{upc}}$ **:** Does increasing update frequency per collection effectively enhance performance (see Table 3)?
- **PVU:** Does the proposed Perturbed Value Update stabilize the O2O training (see Table 2)?
- **Q-value estimation:** Whether the proposed method can effectively address Q-value estimation issues including estimation bias and inaccurate rank (see Figure 5b)?
- **Extension:** Does our method generalize to more challenging robotic manipulation tasks (see Table 4)?

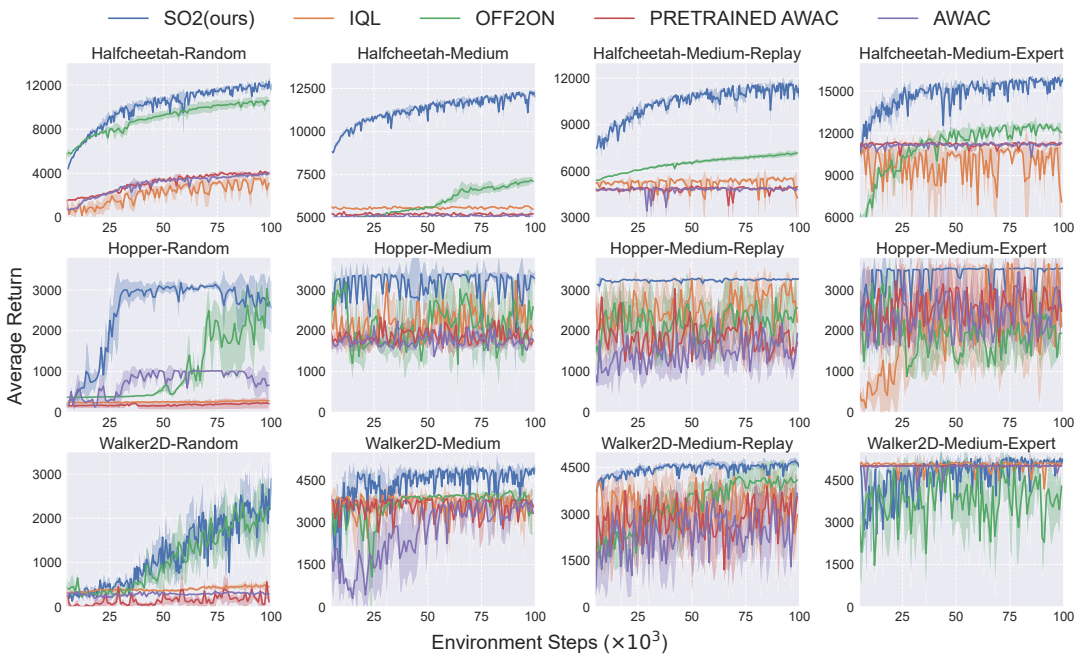

Figure 3: Comparison of learning curves between our proposed method and O2O RL baselines pretrained from D4RL datasets over 100k environment steps. Averaged across 4 seeds with shaded areas indicating standard deviation. Our approach significantly outperforms all baselines in terms of sample efficiency, demonstrating superior asymptotic performance and reduced variance.

- **Compatibility:** Is our method compatible with other O2O RL algorithms (see Table 5)?

Evaluation on MuJoCo Tasks

Setup. We first evaluate SO2 and baselines O2O RL algorithms on MuJoCo (Todorov, Erez, and Tassa 2012) tasks trained from the D4RL-v2 dataset consisting of three environments including HalfCheetah, Walker2d, and Hopper, each with four level datasets collected by policies with different levels, including Random, Medium, Medium-Replay, and Medium-Expert. We report the performance on the standard normalized return metric in D4RL, averaged over 4 seeds.

Comparison. We take the following methods as baselines:

- **OFF2ON:** a pessimistic Q-ensemble RL method that proposes a balanced replay to encourage the use of near-on-policy samples from the offline dataset.
- **IQL-ft:** an offline method with a strong finetuning performance by approximating the upper bound of the Q-value distribution and extracting the policy from the Q-value estimation. We use the implementation in rlkit for IQL.
- **AWAC:** an O2O RL method that trains the policy to imitate actions with high advantage estimates.
- **ODT (Zheng, Zhang, and Grover 2022):** an O2O RL algorithm based on sequence modeling that considers offline and online finetuning in a unified framework.
- **PEX (Zhang, Xu, and Yu 2022):** an RL algorithm based on a policy expansion where the policy set includes offline policy and another policy used for further learning.

We train policies for 100K environment steps and evaluate every 1K step. Our method uses perturbation noise with $\sigma = 0.3$, $c = 0.6$, and $N_{upc} = 10$ as the default setup.

Figure 3 shows the performance of our proposed SO2 against state-of-the-art O2O algorithms during the online finetuning. We re-run the baseline algorithms to ensure a fair comparison. SO2 demonstrates three key advantages: (1) Significantly outperforms all baselines in terms of both sample efficiency and asymptotic performance; (2) Exhibits notably lower variances per environment and per offline dataset compared to baselines, as shown in Figure 4; (3) Even when the offline training data is collected by random policy, SO2 still achieves expert performance with only a few online interactions. Moreover, we compare SO2 and SAC, a standard off-policy algorithm, as shown in Figure 5a. Although the pretrained baseline using the 1M offline data in the HalfCheetah-Random dataset has quite limited performance, SO2 achieves outstanding performance with an average episodic return of 13K, requiring only 0.17M online steps, totaling 1.17M. In comparison, standard SAC requires more than 3M steps to achieve the same performance. It means we fully release the potential of O2O RL in sample efficiency, even when most data is randomly collected and with poor quality –an achievement not observed in any baseline algorithms.

Figure 3 shows that the differences in starting performance across different methods arise from each O2O algorithm utilizing distinct baselines. Moreover, the hyperparameters employed in O2O algorithms are meticulously tailored to their respective baselines, rendering them inherently sensitive. Ensuring consistent starting performance across all algorithms

Dataset		AWAC	ODT	IQL	Off2ON	PEX	SO2 (Ours)
Random	HalfCheetah	6.5 →35.0	10.1 → 18.3	13.4 → 27.3	27.7 → 87.2	15.6 → 55.3	37.7 → **95.6**
	Hopper	6.7 →21.1	6.9 → 31.2	7.6 → 9.3	10.5 → **80.4**	11.0 → 47.0	9.2 → **79.9**
	Walker2d	5.9 →6.3	6.4 → 12.3	6.8 → 9.9	10.3 → 49.4	8.9 → 15.4	6.9 → **62.9**
Medium Replay	HalfCheetah	40.5 → 41.2	32.4 → 39.7	42.7 → 36.7	42.1 → 60.0	45.5 → 51.3	62.5 → **89.4**
	Hopper	37.7 → 60.1	60.4 → 78.5	75.8 → 68.5	28.2 → 79.5	31.5 → 97.1	97.0 → **101.0**
	Walker2d	24.5 → 79.8	44.2 → 71.8	75.6 → 64.9	17.7 → 89.2	80.1 → 92.3	80.9 → **98.2**
Medium	HalfCheetah	43.0 → 42.4	42.7 → 42.1	46.7 → 46.5	39.3 → 59.6	50.8 → 60.9	73.3 → **98.9**
	Hopper	57.8 → 55.1	47.2 → 67.0	73.4 → 61.9	97.5 → 80.2	56.5 → 87.5	77.6 → **101.2**
	Walker2d	35.9 → 72.1	72.0 → 72.2	84.7 → 78.8	66.2 → 72.4	80.1 → 92.3	76.4 → **107.6**
Medium Expert	HalfCheetah	65.2 → 93.0	47.1 → 94.7	88.2 → 59.6	56.7 → 99.3	37.3 → 76.2	87.1 → **130.2**
	Hopper	55.5 → 81.3	64.9 → 111.7	52.8 → 65.0	112.0 → 60.2	91.3 → 99.3	67.2 → **109.1**
	Walker2d	108.3 → 108.7	78.9 → 107.8	110.8 → 110.4	102.1 → 93.3	83.3 → **114.2**	109.5 → **112.9**
	Average	40.6 → 58.0	42.8 → 62.3	56.6 → 53.2	48.4 → 75.9	46.0 → 75.6	76.1 → **98.9**

Table 1: Normalized average returns on D4RL Gym tasks, averaged over 4 random seeds. SO2 significantly improves the performance compared to the state-of-the-art OFF2ON, despite being much simpler to implement.

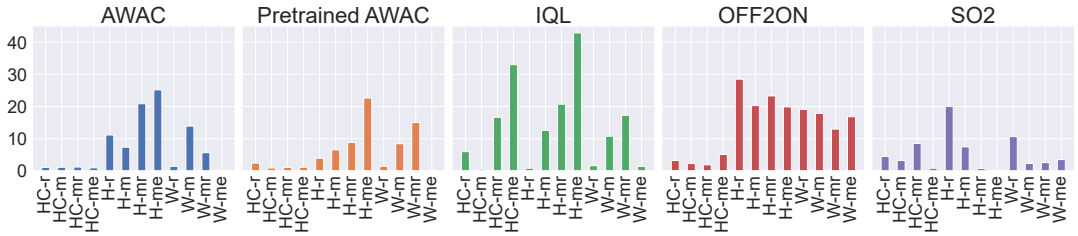

Figure 4: Comparing the standard deviation of ours against O2O RL baselines pretrained from D4RL datasets with 100k environment steps averaged over 4 seeds. Our proposed method SO2 has a smaller standard deviation compared with baselines.

Dataset	$\sigma = 0.15$	$\sigma = 0.3$	$\sigma = 0.45$
W-R	24.6 ±12.9	**62.9 ±10.6**	22.9 ± 12.8
W-M	93.5 ±16.0	**107.6 ±2.3**	105.5± 3.3
W-MR	94.5 ±7.7	**98.2 ±2.6**	91.4 ± 3.1
W-ME	110.6 ±6.9	**112.9** ±3.5	110.2± **1.9**
Average	80.8 ±10.9	**95.4 ±4.7**	82.5 ± 5.3

Table 2: SO2 Performance across various σ on the Walker2d. W=Walker2d, R=Random, M=Medium, E=Expert.

Dataset	$N_{upc} = 1$	$N_{upc} = 5$	$N_{upc} = 10$
W-R	29.5 ±12.6	30.8 ± 14.9	**62.8 ± 10.6**
W-M	97.5 ± 5.0	97.1 ± 17.4	**107.6 ± 2.2**
W-MR	94.4 ± 2.2	**99.0 ± 1.3**	**98.2** ± 2.5
W-ME	103.7 ± 3.4	111.2 ± **1.4**	**112.9** ± 3.5
Average	81.3 ± 5.8	84.5 ± 8.7	**95.4** ± 4.7

Table 3: Performance of SO2 over various N_{upc} pretrained on Walker2d. N_{upc} denotes the update frequency per collection. W=Walker2d, R=Random, M=Medium, E=Expert.

would require an extensive search for suitable parameters. Consequently, most O2O algorithms (Nair et al. 2020; Lee et al. 2022; Zheng, Zhang, and Grover 2022; Zhang, Xu, and Yu 2022) present results from different starting points to accommodate these algorithm-specific sensitivities.

Analysis on Perturbed Value Update. In this paper, we argue that in the O2O task, adding appropriate noise to the target action can make training more stable and result in better policy performance. To verify the effectiveness of target noise, we conduct an ablation over σ. As shown in Table 2, the baseline without the target noise ($\sigma = 0$) has significant variances in all offline datasets, up to 1/3 of the average performance. The high variance reveals that even if the learned policies have the same initial performance, some of them may still perform poorly on some episodes, and the learned policy may fluctuate dramatically among each evaluation. On the contrary, the volatile policy, once equipped with the Perturbed Value Update, achieves consistently impressive performance with low variance across various PVUs.

Analysis on N_{upc}. To verify the effectiveness of N_{upc}, we report the performance of SO2 on Walker2D with different N_{upc}. Table 3 shows that the performance consistently improves with the increase of N_{upc}, in all environments on Walker2D. It is worth noting that increasing N_{upc} on random data has the most obvious effect. Intuitively, the N_{upc} term induces the value network to favor frequent updates after each interaction with environments. Consequently, height-

Algorithm 1: SO2

Initialized policy parameters θ, Q-function parameters $\{\phi_j\}_{j=1}^N$, target Q-function parameters $\{\phi'_j\}_{j=1}^N$, offline replay buffer $\mathcal{D}^{\text{off}}$, online replay buffer $\mathcal{D}^{\text{on}}$, epoch number E and frequency of Q-value updates N_{upc}, and batch size B.
for epoch e in $1, 2, ..., E$ **do**
  Collect a transition $\tau = (s, a, r, s')$ via environment interaction with π_θ. ▷ Collect samples
  Update online replay buffer $\mathcal{D}^{\text{on}} \leftarrow \mathcal{D}^{\text{on}} \cup \tau$.
  for N_{upc} iterations **do**
    Sample a mini-batch $\mathbf{b}$ of transition $(\mathbf{s}, \mathbf{a}, r, \mathbf{s}')$ from $\mathcal{D}^{\text{off}} \cup \mathcal{D}^{\text{on}}$.
    Compute target Q-values $\mathcal{T}Q_{\phi_i}(\mathbf{s}, \mathbf{a})$ with the objective in Equation 3.
    Update each Q-function Q_{ϕ_i} by gradient descent with: ▷ Value Phase

$$\nabla_{\phi_i} \frac{1}{B} \sum_{(\mathbf{s},\mathbf{a},r,\mathbf{s}')\in\mathbf{b}} \left(Q_{\phi_i}(\mathbf{s}, \mathbf{a}) - \mathcal{T}Q_{\phi_i}(\mathbf{s}, \mathbf{a}) \right)^2.$$

    Update policy by gradient ascent with: ▷ Policy Phase

$$\nabla_\theta \frac{1}{B} \sum_{\mathbf{s}\in\mathbf{b}} \left(\min_{j=1,...,N} Q_{\phi_j}(\mathbf{s}, \tilde{\mathbf{a}}_\theta(\mathbf{s})) - \beta \log \pi_\theta(\tilde{\mathbf{a}}_\theta(\mathbf{s}) \mid \mathbf{s}) \right),$$

    where $\tilde{\mathbf{a}}_\theta(\mathbf{s})$ is a sample from $\pi_\theta(\cdot \mid \mathbf{s})$ which is differentiable w.r.t. θ via the reparametrization trick.
    Update target networks with $\phi'_i \leftarrow \rho\phi'_i + (1-\rho)\phi_i$.
  end for
end for

Dataset	AWAC	IQL	Ours(400k)
pen-human	44.6→70.3	37.4→60.7	51.4→**70.6**
door-human	1.3→30.1	0.7→32.3	16.5→**66.2**
Total	46.7→103.1	38.1→124.0	67.9→ **136.8**

Table 4: Normalized average scores on the Adroit. We report the finetuning results of baselines with 1M environment steps, while the online finetuning results of ours with 400k steps.

ened N_{upc} yields more precise Q-value estimations, particularly beneficial when initial estimations are suboptimal.

Analysis on Q-value estimation. Figure 5b shows an accuracy comparison of the rank of the estimated Q-value during online finetuning, between the EDAC and variants that integrate individual components of SO2. It is shown that SO2 outperforms the baseline consistently in all environments, which further validates our idea of target noise and frequent updates to boost Q-value estimation to advance finetuning.

Compatibility. We report the results of the combination of SO2 with OFF2ON and PEX (Zhang, Xu, and Yu 2022) in Table 5. Our results demonstrate significant performance improvements when applying our proposed SO2 method to

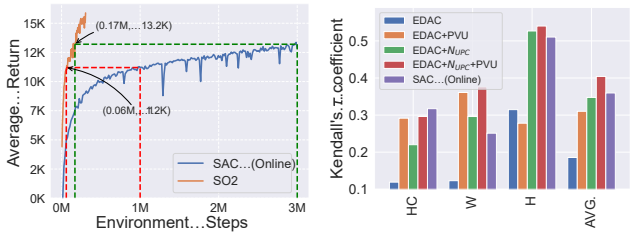

Figure 5: (a): Performance comparing our proposed method against online SAC in HalfCheetah. Although the baseline is initialized from the HalfCheetah-Random dataset with limited performance, our proposed method boosts the baseline algorithm effectively and significantly improves the sample efficiency by up to 17 compared to SAC. (b): Kendall's τ coefficient comparing ablation over Perturbed Value Update (PVU), and update per collection ($N_{upc} = 10$) in the MuJoco. This measures the rank correlation between true Q-value and estimated Q-value for similar state-action pairs.

	100k	Algo. + SO2
OFF2ON	75.89 ± 14.26	88.64 ± 4.33
PEX	67.68 ± 8.83	74.26 ± 4.26

Table 5: Compatibility of SO2. SO2 Integration with OFF2ON and PEX, Reported with 100K Online Samples.

OFF2ON (Lee et al. 2022), PEX (Zhang, Xu, and Yu 2022), underscoring the effectiveness of our approach.

Evaluation on Adroit Tasks

Setup and Results. We also conduct experiments on Adroit. Specifically, the offline policy is trained with limited human demonstrations, and the online finetuning policy is trained with **1M** environment steps for baseline, and with only **400k** environment steps for ours. Results for AWAC and IQL are quoted from IQL paper. Table 4 shows that our method also outperforms all baselines by a large margin.

Conclusion

In this paper, we have delved into O2O reinforcement learning and systematically studied why this setting is challenging. Different from most existing works, we in-depth analyze the Q-value estimation issues in offline-to-online including the biased estimation and inaccurate rank of the Q-value, besides the bootstrap error resulting from state-action distribution shift. Based on this argument, we propose smoothed offline-to-online (SO2). It effectively and efficiently improves the Q-value estimation by perturbing the target action and improving the frequency of Q-value updates. The proposed method, without any explicit estimation of the state-action distribution shift and complex components to balance offline and online replay buffers, remarkably improves the performance of the state-of-the-art methods by up to 83.1% on the MuJoco and Adroit environments.

Acknowledgments

This research is funded by Shanghai AI Laboratory. This work is partially supported by the National Key R&D Program of China (NO.2022ZD0160100), (NO.2022ZD0160101), and National Natural Science Foundation of China (62376013). This work was done during Yinmin's internship at Shanghai Artificial Intelligence Laboratory. We would like to thank many colleagues for useful discussions, suggestions, feedback, and advice, including: Zilin Wang and Ruoyu Gao.

References

Agarwal, R.; Schuurmans, D.; and Norouzi, M. 2020. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, 104–114. PMLR.

Agarwal, R.; Schwarzer, M.; Castro, P. S.; Courville, A. C.; and Bellemare, M. 2022. Reincarnating reinforcement learning: Reusing prior computation to accelerate progress. *Advances in Neural Information Processing Systems*, 35: 28955–28971.

An, G.; Moon, S.; Kim, J.-H.; and 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.; and Wang, Z. 2021. Pessimistic Bootstrapping for Uncertainty-Driven Offline Reinforcement Learning. In *International Conference on Learning Representations*.

Bommasani, R.; Hudson, D. A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M. S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics.

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

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

Gupta, A.; Kumar, V.; Lynch, C.; Levine, S.; and Hausman, K. 2020. Relay Policy Learning: Solving Long-Horizon Tasks via Imitation and Reinforcement Learning. In *Conference on Robot Learning*, 1025–1037. PMLR.

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

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2021. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*.

Ijspeert, A.; Nakanishi, J.; and Schaal, S. 2002. Learning attractor landscapes for learning motor primitives. *Advances in neural information processing systems*, 15.

Kim, B.; Farahmand, A.-m.; Pineau, J.; and Precup, D. 2013. Learning from limited demonstrations. *Advances in Neural Information Processing Systems*, 26.

Kostrikov, I.; Nair, A.; and Levine, S. 2021. Offline Reinforcement Learning with Implicit Q-Learning. In *International Conference on Learning Representations*.

Kumar, A.; Fu, J.; Soh, M.; Tucker, G.; and Levine, S. 2019. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32.

Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33: 1179–1191.

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

Li, C.; Jia, R.; Yao, J.; Liu, J.; Zhang, Y.; Niu, Y.; Yang, Y.; Liu, Y.; and Ouyang, W. 2023a. Theoretically Guaranteed Policy Improvement Distilled from Model-Based Planning. In *PRL Workshop at IJCAI*.

Li, C.; Liu, J.; Zhang, Y.; Wei, Y.; Niu, Y.; Yang, Y.; Liu, Y.; and Ouyang, W. 2023b. Ace: Cooperative multi-agent q-learning with bidirectional action-dependency. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 8536–8544.

Liu, J.; Zhang, Y.; Li, C.; Yang, C.; Yang, Y.; Liu, Y.; and Ouyang, W. 2023. Masked Pretraining for Multi-Agent Decision Making. *arXiv preprint arXiv:2310.11846*.

Nair, A.; Chen, D.; Agrawal, P.; Isola, P.; Abbeel, P.; Malik, J.; and Levine, S. 2017. Combining self-supervised learning and imitation for vision-based rope manipulation. In *2017 IEEE international conference on robotics and automation (ICRA)*, 2146–2153. IEEE.

Nair, A.; Gupta, A.; Dalal, M.; and Levine, S. 2020. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 8748–8763. PMLR.

Theodorou, E.; Buchli, J.; and Schaal, S. 2010. A generalized path integral control approach to reinforcement learning. *The Journal of Machine Learning Research*, 11: 3137–3181.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5026–5033. IEEE.

Wang, L.; Liu, J.; Shao, H.; Wang, W.; Chen, R.; Liu, Y.; and Waslander, S. L. 2023. Efficient Reinforcement Learning for Autonomous Driving with Parameterized Skills and Priors. *Science and Systems (RSS)*.

Wu, Y.; Tucker, G.; and Nachum, O. 2019. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*.

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

Zhang, H.; Xu, W.; and Yu, H. 2022. Policy Expansion for Bridging Offline-to-Online Reinforcement Learning. In *The Eleventh International Conference on Learning Representations*.

Zhao, Y.; Boney, R.; Ilin, A.; Kannala, J.; and Pajarinen, J. 2022. Adaptive behavior cloning regularization for stable offline-to-online reinforcement learning. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*.

Zheng, Q.; Zhang, A.; and Grover, A. 2022. Online decision transformer. In *international conference on machine learning*, 27042–27059. PMLR.

Zhu, H.; Gupta, A.; Rajeswaran, A.; Levine, S.; and Kumar, V. 2019. Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. In *2019 International Conference on Robotics and Automation (ICRA)*, 3651–3657. IEEE.