Optimistic Q-value Adaptation for Offline-to-Online Reinforcement Learning

Hao Wu[1,2], **Shunhao Zhang**[1,2], **Shuai Lü**[1,2*]
[1]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Ministry of Education, Changchun 130012, China
{haowu24, shunhao24}@mails.jlu.edu.cn, lus@jlu.edu.cn

Abstract

Offline-to-online reinforcement learning (O2O RL) aims to improve offline RL agents with limited online interaction. Prior work has shown that, in conservative value-regularized methods such as CQL, pessimistic offline critics can underestimate values and hinder online fine-tuning. We show that this issue is more general: critic underestimation also appears in explicit policy-constraint methods such as TD3+BC, and can persist or become more severe through bootstrapped Bellman updates during fine-tuning. To address this problem, we propose Optimistic Q-value Adaptation (OQA), a framework that corrects underestimation at the Bellman-backup level. OQA constructs an additional offline-pretrained actor–critic pair and uses it together with the original pair to form a controlled optimistic backup that mitigates inherited pessimism during fine-tuning. Unlike trajectory-return calibration methods, OQA does not rely on return lower bounds and is not tied to CQL-style regularization. Across D4RL MuJoCo locomotion, Maze2D, AntMaze, and Adroit benchmarks, OQA instantiations based on TD3+BC and CQL consistently outperform the corresponding backbones and strong O2O RL baselines.

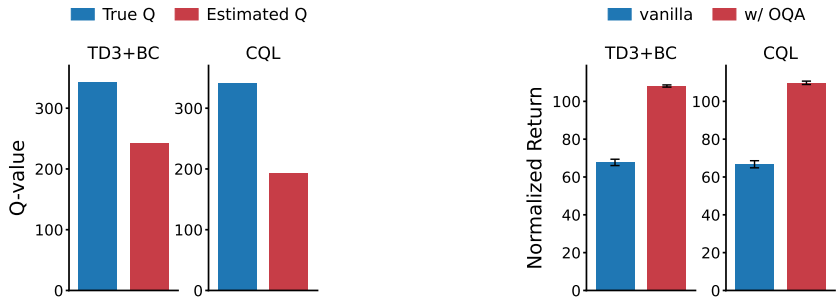
1 Introduction

Reinforcement learning (RL) [6, 5] can learn high-performing policies through online interaction, but it often requires millions of environment steps. This requirement is difficult to meet when data collection is costly, risky, or slow. Offline RL [10, 4] reduces this cost by learning from fixed datasets, but its performance depends heavily on data quality and coverage. When offline data is suboptimal or narrow, the learned policy often remains far from optimal. Offline-to-online (O2O) RL [9, 14, 17] addresses this limitation by using an offline policy as the starting point for online improvement. However, fine-tuning from an offline initialization can still be slow or unstable, suggesting that errors inherited from offline training may hinder adaptation.

Previous work [14] has shown that, in conservative value-regularized methods such as Conservative Q-learning (CQL), critics can become overly pessimistic. The resulting Q-value underestimation can be a major bottleneck for online fine-tuning because it weakens the value signal used for policy improvement. We show that this issue is more general than previously recognized. Critic underestimation is not restricted to conservative value-regularized methods, but also appears in explicit policy-constraint methods such as TD3+BC, where slow online improvement is often attributed mainly to the behavior-cloning constraint. As shown in Figure 1(a), the offline critics of both CQL and TD3+BC substantially underestimate Monte Carlo returns on locomotion tasks. We further observe that this inherited underestimation is not naturally corrected by online interaction. Because

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Critic Bias (b) OQA Gains

Figure 1: Offline critic underestimation and its impact on O2O fine-tuning. (a) Underestimation of offline critics on locomotion tasks for TD3+BC and CQL, where estimated Q values are compared with Monte Carlo returns. (b) Aggregated normalized scores on locomotion tasks, comparing fine-tuning with the vanilla backbone and fine-tuning with OQA.

critic learning relies on bootstrapped Bellman updates, underestimated next-state values can induce underestimated current targets. Without explicit correction, this pessimistic bias can propagate through fine-tuning and become more severe over time.

These observations motivate correcting underestimation at the Bellman-backup level during online fine-tuning. If the backup target remains pessimistic, critic learning can continue to reinforce underestimation even after online data becomes available. Instead of modifying the offline training objective or discarding the pretrained critic, we focus on the online fine-tuning stage and aim to reduce the propagation of pessimistic targets. This leads to a backup-level correction that counteracts inherited bias while preserving the stability provided by conservative value estimates.

We propose Optimistic Q-value Adaptation (OQA), a backbone-compatible framework that introduces controlled optimism into the Bellman backup. OQA trains an additional offline-pretrained actor–critic pair on a perturbed version of the offline dataset, producing two conservative but non-identical value priors together with the original pair. During fine-tuning, OQA uses these priors to select a less underestimated backup and interpolates it with the original conservative target. This mechanism reduces compounding underestimation without causing uncontrolled overestimation. Compared with Cal-QL, which calibrates CQL targets using trajectory returns, OQA modifies the Bellman backup directly during online fine-tuning. It does not rely on trajectory-return lower bounds and is not tied to CQL-style conservative regularization, making it compatible with representative offline backbones such as CQL and TD3+BC. As shown in Figure 1(b), OQA improves fine-tuning performance when integrated with different offline backbones on locomotion tasks.

Our contributions are threefold: (1) we show that critic underestimation during online fine-tuning occurs in both conservative value-regularized methods such as CQL and explicit policy-constraint methods such as TD3+BC, and identify it as a key factor behind slow improvement through Q-value bias analysis; (2) we propose OQA, an efficient O2O RL framework that introduces controlled optimism to accelerate value adaptation and can be combined with representative offline RL algorithms across goal-conditioned and dense-reward settings; (3) we instantiate OQA with TD3+BC and CQL, evaluate the resulting methods on D4RL MuJoCo locomotion, Maze2D, AntMaze, and Adroit benchmarks, and show consistent gains over the corresponding backbones and strong O2O RL baselines, with ablations validating the effectiveness of each component.

2 Background and Related Work

RL is typically modeled as a Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, d_0, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $P(s' \mid s, a)$ denotes the transition kernel, $r(s, a)$ is the reward function, d_0 is the initial-state distribution, and $\gamma \in (0, 1]$ is the discount factor. A policy $\pi(a \mid s)$ induces a trajectory $\tau = (s_0, a_0, s_1, a_1, \dots)$ with return $R(\tau) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$. The aim is to find an optimal policy $\pi^\star \in \arg\max_\pi J(\pi)$, where $J(\pi) = \mathbb{E}_{\tau\sim\pi}[R(\tau)]$.

2.1 Offline RL Backbone Algorithms

Offline RL learns a policy from a static dataset $\mathcal{D} = \{(s, a, r, s')\}$ collected by a behavior policy, without further environment interaction. In O2O RL, such an offline algorithm typically provides the pre-trained initialization for subsequent online fine-tuning. We therefore consider two representative offline backbones: CQL for conservative value learning and TD3+BC for explicit policy constraints.

CQL [10] learns a pessimistic critic by penalizing Q-values for OOD actions. Given the dataset $\mathcal{D}$, the critic is trained by minimizing:

$$\alpha\,\mathbb{E}_{s\sim\mathcal{D}}\Big[\log\sum_{a}\exp\big(Q(s,a)\big)-\mathbb{E}_{a\sim\hat{\pi}_\beta(\cdot|s)}\big[Q(s,a)\big]\Big]+\frac{1}{2}\,\mathbb{E}_{(s,a)\sim\mathcal{D}}\Big[\big(Q(s,a)-\mathcal{T}^{\pi^k}\hat{Q}^k(s,a)\big)^2\Big] \quad (1)$$

where $\alpha > 0$ controls the strength of the conservative regularizer, and $\mathcal{T}^{\pi^k}$ denotes the empirical Bellman backup in the temporal-difference (TD) term. The policy is then updated with an entropy-regularized objective using the learned conservative critic.

TD3+BC [4] uses the standard TD3 critic update and augments the actor objective with an explicit behavior-cloning regularizer:

$$\max_{\pi}\ \mathbb{E}_{(s,a)\sim\mathcal{D}}\big[\lambda_Q\,Q(s,\pi(s)) - \|\pi(s)-a\|_2^2\big], \quad (2)$$

where λ_Q denotes the adaptive value-scaling coefficient used in the original TD3+BC objective. We use the subscript Q to distinguish this adaptive coefficient from the fixed policy-constraint coefficient λ_{BC} used in our online fine-tuning objectives.

2.2 Offline-to-Online RL

Existing O2O RL methods can be broadly categorized into critic calibration, policy regularization, and policy exploration, along with a distinct no-retention fine-tuning paradigm.

Critic Calibration. OPT [15] introduces an online pre-training stage to learn an additional value function, then fine-tunes it jointly with the offline critic to guide policy updates. SO2 [20] and SUF [2] both use high update-to-data ratios to help the critic rapidly adapt to the online distribution, a setting now widely used in O2O RL. Ensemble methods [11, 21] stabilize Q estimation under distribution shift by ensembling value functions across offline and online phases. Cal-QL [14] modifies CQL [10] by calibrating its conservatism to reduce overly pessimistic value estimates on out-of-distribution (OOD) actions and improve value learning during fine-tuning. O2TD3 [13] and ACA [18] both abandon the offline-pretrained Q function and re-align value estimation with the offline policy for stable fine-tuning, via retraining a new critic or re-estimating Q values under the offline policy, respectively.

Policy Regularization. ABCR [22] adjusts constraint strength based on policy performance. FamO2O [17] adjusts constraint strength at the sample level based on state quality. APL [23] applies constraints only to offline samples to mitigate excessive conservatism during fine-tuning. SPA [12] establishes state-proficiency-based sample priorities, emphasizing low-proficiency samples for policy improvement and high-proficiency samples for policy constraint.

Policy Exploration. PEX [19] collects online data using both the frozen offline policy and randomly initialized policies to broaden action coverage. CPR [8] improves fine-tuning stability and efficiency via periodic policy revitalization. BOORL [7] trains multiple offline agents to approximate a posterior over policies and fine-tunes by sampling actions for online interaction in a Thompson-sampling style.

No-Retention Fine-tuning. Unlike prior methods that mix offline and online data during fine-tuning, WSRL [24] fine-tunes the offline policy without retaining the offline dataset and initializes the replay buffer with a few online trajectories collected by the frozen offline policy.

3 Method

In this section, we propose OQA, an efficient O2O RL framework that can be combined with different offline backbones. In Section 3.1, we analyze why continuing to fine-tune with offline RL algorithms can improve slowly, from the perspective of Q-value estimation bias. In Section 3.2, we present OQA and describe how it adjusts value learning to mitigate underestimation during fine-tuning. In Section 3.3, we summarize implementation details of the OQA framework.

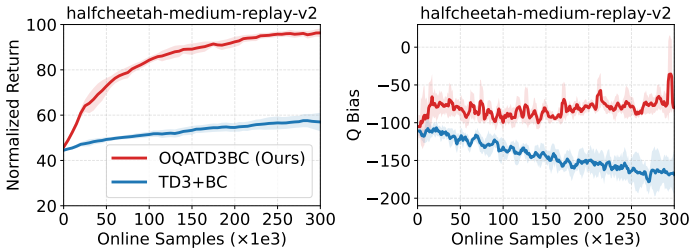


Figure 2: Online fine-tuning dynamics of TD3+BC. Although performance steadily improves, critic underestimation persists and progressively intensifies during fine-tuning.

3.1 Critic Underestimation During Fine-Tuning

We first examine why online fine-tuning from an offline RL backbone can remain slow after online interaction begins. For policy-constraint methods such as TD3+BC, slow improvement is often attributed to the behavior-cloning term, which restricts how far the policy can move away from the offline data. This explanation is important but incomplete. We find that TD3+BC also inherits a negatively biased critic from offline pretraining, and that this bias can persist and grow during online fine-tuning.

To measure this effect, we track Q-value estimation bias during fine-tuning. At each evaluation point, we sample initial states from the environment, roll out 10 trajectories of length 1000 under the current policy, and use the resulting discounted Monte Carlo returns as empirical reference targets. To reduce temporal correlation, we subsample one transition every 5 steps, yielding roughly 2000 state–action pairs per evaluation. As shown in Figure 2, TD3+BC improves its normalized return during online fine-tuning, but its critic remains negatively biased. The magnitude of this negative bias also increases as training proceeds, suggesting that online interaction alone does not necessarily correct the underestimation inherited from offline pretraining.

Table 1: Fine-tuning dynamics with TD3+BC on halfcheetah-medium-replay-v2. Q-bias and median error are computed against Monte Carlo returns.

Step	Q-value	Q bias	Error quantile (q=0.5)
0	379.0	-109.8	-148.6
50k	418.4	-109.2	-152.5
100k	433.7	-131.8	-177.8
150k	450.5	-146.3	-201.7
200k	460.0	-148.6	-195.9
250k	467.5	-160.9	-204.8
300k	474.2	-163.5	-221.8

Table 1 gives a more detailed view of this trend. Although the mean Q-value increases from 379.0 to 474.2, the Q-bias becomes more negative, changing from -109.8 to -163.5, and the median error remains below zero throughout fine-tuning. Thus, the critic does not only remain conservative at initialization; it continues to underestimate a large fraction of evaluated state–action pairs as fine-tuning progresses. This behavior is expected in explicitly pessimistic methods such as CQL, but it is more notable in TD3+BC, where the main bottleneck is often assumed to be the policy constraint. In TD3+BC, clipped double-Q targets can bias value estimates downward, and bootstrapped Bellman updates can pass this bias across training iterations. As a result, pessimistic targets can continue to weaken the policy-improvement signal even after new online data is collected. This motivates correcting underestimation at the Bellman-backup level during online fine-tuning.

3.2 Optimistic Q-value Adaptation

As discussed above, offline-pretrained critics often exhibit underestimation. This underestimation can propagate through bootstrapped updates, making critic learning overly conservative and slowing performance gains during fine-tuning. This motivates addressing the problem from the Bellman-backup perspective, by modifying the one-step backup target to counteract the propagation of pessimism.

Based on this insight, we propose Optimistic Q-value Adaptation (OQA), a simple yet effective O2O RL framework that can be combined with different offline backbones. OQA introduces an additional offline-pretrained actor–critic pair and, together with the original backbone pair, constructs

a controlled optimistic Bellman operator that corrects the conservative Bellman backup during fine-tuning. We describe OQA in two parts: 1) **Offline training** describes how we obtain the additional offline-pretrained actor–critic pair, and 2) **Online fine-tuning** describes how we leverage both the original and additional pairs to construct the optimistic Bellman operator for value learning.

Offline training. OQA starts from two independently trained offline priors. Let the offline dataset be $\mathcal{D} = \{\tau_n\}_{n=1}^{N}$, where each transition is $\tau_n = (s_n, a_n, r_n, s'_n)$. For each branch $i \in \{1, 2\}$, we introduce a transition selector $M_i = (m_{i,1}, \ldots, m_{i,N}) \in \{0, 1\}^N$, with $m_{i,n} \sim \text{Bernoulli}(p)$ independently across samples and branches. The corresponding perturbed dataset is

$$\mathcal{D}^{(i)} = M_i \odot \mathcal{D} = \{\tau_n \in \mathcal{D} : m_{i,n} = 1\}. \tag{3}$$

Here p controls the retention probability and therefore determines the strength of data perturbation.

Given $\mathcal{D}^{(i)}$, the i-th branch is pretrained with the same offline RL objective as the backbone:

$$(\pi_{\psi_i}, Q_{\theta_{i,1}}, Q_{\theta_{i,2}}) = \arg\min_{\pi, Q} \mathcal{L}_{\text{off}}(\pi, Q; \mathcal{D}^{(i)}), \qquad i \in \{1, 2\}. \tag{4}$$

OQA does not alter the offline objective, network architecture, or hyperparameters in this stage. The two branches share the same offline training configuration but are trained independently on separately sampled subsets. This design has two effects. First, both branches inherit the conservative value estimates produced by the offline backbone. Second, because their empirical training distributions are slightly different, their estimation errors are not fully coupled. Thus, the offline stage produces two conservative but non-identical actor–critic priors. The online stage then uses these priors to construct a controlled optimistic Bellman backup rather than relying on a single pessimistic target.

Online fine-tuning. Let $\mathcal{B}$ denote the online replay buffer collected during fine-tuning. We first recall the standard critic update in many offline backbones, which trains a single actor–critic pair with a double-Q critic. For a given pair indexed by i, let $\theta_i \triangleq (\theta_{i,1}, \theta_{i,2})$; the critic is trained by minimizing:

$$\mathcal{L}(\theta_i) = \mathop{\mathbb{E}}_{(s,a,r,s') \sim \mathcal{D} \cup \mathcal{B}} \left[\sum_{j=1,2} \left(Q_{\theta_{i,j}}(s, a) - \mathcal{T}_{\text{con}}^{\pi_{\psi_i}} \bar{Q}(s, a) \right)^2 \right]. \tag{5}$$

Here $\mathcal{T}_{\text{con}}^{\pi_{\psi_i}} \bar{Q}(s, a)$ is the standard conservative one-step Bellman backup under policy π_{ψ_i}, defined as

$$\mathcal{T}_{\text{con}}^{\pi_{\psi_i}} \bar{Q}(s, a) = r + \gamma \min_{j \in \{1,2\}} \bar{Q}_{\theta_{i,j}}\big(s',\, \pi_{\psi_i}(s')\big), \tag{6}$$

where $\bar{Q}_{\theta_{i,j}}$ denotes the target critic. When $\bar{Q}$ is underestimated after offline pretraining, Eq. (6) bootstraps from pessimistic values at s', so minimizing Eq. (5) can propagate and compound this pessimism over updates.

To counteract this effect, OQA introduces a controlled optimistic component into the Bellman backup. Given two underestimated backups, the less underestimated one provides a reasonable optimistic backup, which helps mitigate underestimation while avoiding excessive optimism and reducing the risk of overestimation. Accordingly, we define the optimistic Bellman operator $\mathcal{T}_{\text{opt}}^{\pi}$ by selecting the less conservative backup between the two pretrained actor–critic pairs $\{(\pi_{\psi_i},\, Q_{\theta_{i,1}},\, Q_{\theta_{i,2}})\}_{i=1,2}$:

$$\mathcal{T}_{\text{opt}} \bar{Q}(s, a) = \max_{i \in \{1,2\}} \mathcal{T}_{\text{con}}^{(i)} \bar{Q}(s, a). \tag{7}$$

To balance optimism with the stabilizing effect of conservative backups, OQA defines a corrected Bellman backup as an interpolation between the prior-specific conservative backup and the optimistic backup, controlled by $\alpha \in [0, 1]$:

$$\mathcal{T}_{\text{OQA}}^{(i)} \bar{Q}(s, a) = (1 - \alpha)\, \mathcal{T}_{\text{con}}^{(i)} \bar{Q}(s, a) + \alpha\, \mathcal{T}_{\text{opt}} \bar{Q}(s, a). \tag{8}$$

OQA does not require directly replacing the conservative backup with the optimistic backup; instead, α provides a knob to interpolate between them. In most tasks, we use $\alpha < 1$ to control potential overestimation, while in navigation tasks we allow $\alpha = 1$ because stronger optimism is empirically stable and beneficial in sparse-reward settings.

Algorithm 1 Optimistic Q-value Adaptation (OQA)

Input: Offline dataset $\mathcal{D}$; pretrained pairs $\{(\pi_{\psi_i}, Q_{\theta_{i,1}}, Q_{\theta_{i,2}})\}_{i=1}^{2}$; optimism coefficient α
Initialize target critics $\{\bar{Q}_{\theta_{i,1}}, \bar{Q}_{\theta_{i,2}}\}_{i=1}^{2}$ and online buffer $\mathcal{B}$
for $t = 1$ **to** T **do**
  Select $i^\star(s_t)$ by Eq. (9), execute $a_t \sim \pi_{\psi_{i^\star}}(\cdot \mid s_t)$, and add transition to $\mathcal{B}$
  Sample a minibatch from $\mathcal{D} \cup \mathcal{B}$
  Compute $\{\mathcal{T}_{\mathrm{con}}^{(i)} \bar{Q}(s,a)\}_{i=1}^{2}$ by Eq. (6)
  Compute $\mathcal{T}_{\mathrm{opt}} \bar{Q}(s,a)$ by Eq. (7)
  for $i \in \{1,2\}$ **do**
    Form $\mathcal{T}_{\mathrm{OQA}}^{(i)} \bar{Q}(s,a)$ by Eq. (8)
    Update critics with Eq. (5), replacing $\mathcal{T}_{\mathrm{con}}^{(i)} \bar{Q}(s,a)$ by $\mathcal{T}_{\mathrm{OQA}}^{(i)} \bar{Q}(s,a)$
    Update actor using Eq. (10) or Eq. (11); update target networks
  end for
end for

3.3 Implementation Details

We integrate OQA with TD3+BC and CQL, resulting in OQATD3BC and OQACQL. During fine-tuning, both actor–critic pairs use the same online replay buffer $\mathcal{B}$ and train on minibatches from $\mathcal{D} \cup \mathcal{B}$. At each state s, we execute the policy with the larger conservative value estimate:

$$i^\star(s) = \arg\max_{i \in \{1,2\}} \min_{k \in \{1,2\}} Q_{\theta_{i,k}}\big(s, \pi_{\psi_i}(s)\big). \tag{9}$$

The action is then taken as $a = \pi_{\psi_{i^\star(s)}}(s)$. This selection executes the policy with the higher conservative Q-value, promoting optimistic yet stable exploration.

For actor updates, we use a unified constraint-weighted parameterization during online fine-tuning. A fixed coefficient λ_{BC} directly controls the strength of the explicit behavior constraint in both OQATD3BC and OQACQL. This design separates the behavior-constraint coefficient from the adaptive value-scaling coefficient λ_Q in the original TD3+BC actor objective, making the constraint strength directly tunable across backbones. For OQATD3BC, for each $i \in \{1,2\}$, we minimize

$$\mathcal{L}(\psi_i) = \mathbb{E}_{(s,a)\sim\mathcal{D}\cup\mathcal{B}}\Big[- \min_{k \in \{1,2\}} Q_{\theta_{i,k}}\big(s, \pi_{\psi_i}(s)\big) + \lambda_{\mathrm{BC}} \left\| \pi_{\psi_i}(s) - a \right\|_2^2 \Big]. \tag{10}$$

For OQACQL, we retain the entropy-regularized actor objective and also apply a similar policy constraint during fine-tuning. Specifically, for each $i \in \{1,2\}$, we sample $\tilde{a}_i \sim \pi_{\psi_i}(\cdot \mid s)$ and minimize

$$\mathcal{L}(\psi_i) = \mathbb{E}_{(s,a)\sim\mathcal{D}\cup\mathcal{B},\, \tilde{a}_i \sim \pi_{\psi_i}(\cdot|s)} \Big[\alpha_\pi \log \pi_{\psi_i}(\tilde{a}_i \mid s) - \min_{k \in \{1,2\}} Q_{\theta_{i,k}}(s, \tilde{a}_i) - \lambda_{\mathrm{BC}} \log \pi_{\psi_i}(a \mid s) \Big]. \tag{11}$$

This constraint helps prevent excessive drift into strongly OOD regions where the offline critic is less reliable. Algorithm 1 outlines the OQA fine-tuning procedure.

4 Experiments

Experimental Setup. We evaluate our method on four domains from the D4RL benchmark [3]: MuJoCo [16] locomotion, AntMaze navigation, Maze2D navigation, and Adroit dexterous manipulation. Additional details of the environments are provided in Appendix A. In the offline pre-training phase, we train policies for 1M gradient steps using the corresponding offline RL algorithms on the provided datasets. In the online fine-tuning phase, the offline policies are used for initialization and further trained for an additional 300k environment interactions. We report the normalized return defined in D4RL, averaged over 5 random seeds.

Baselines. We compare our method against two groups of baselines: offline RL and O2O RL. Offline RL backbones include TD3+BC [4] and CQL [10]. For O2O RL, we consider (i) TD3+BC-based methods, including SPA [12], OPT [15], O2TD3 [13], and CPR [8]. (ii) CQL/IQL-based methods, including Cal-QL [14], PEX [19], FamO2O [17] and IQL [9]. Implementation details for all methods are provided in Appendix B.

Table 2: Online fine-tuning performance on D4RL locomotion tasks (r = random, m = medium, mr = medium-replay, me = medium-expert). ± denotes one standard deviation over 5 random seeds. **Bold** indicates the best mean performance, and underlining indicates the second-best mean performance.

Dataset (v2)	TD3+BC	CQL	FamO2O	Cal-QL	CPR	O2TD3	OPT	SPA	OQATD3BC	OQACQL
halfcheetah-r	45.4	13.6	40.0	4.5	47.7	42.9	85.0	83.4	**101.6**±2.4	93.7±6.7
halfcheetah-m	64.4	66.1	49.6	68.0	70.6	74.5	96.0	91.7	104.1±0.7	**108.3**±4.1
halfcheetah-mr	52.9	37.2	47.7	52.1	68.6	67.4	92.3	84.7	95.3±1.2	**102.2**±1.4
halfcheetah-me	94.6	96.8	87.3	96.6	91.9	99.7	93.4	100.7	105.9±2.2	**126.3**±2.9
hopper-r	10.2	8.8	10.4	12.0	9.4	8.7	96.2	97.7	103.9±10.2	**109.9**±3.7
hopper-m	60.5	73.2	71.8	99.0	100.8	104.5	109.5	110.9	**111.4**±1.3	111.1±1.8
hopper-mr	92.7	103.0	101.8	99.9	102.4	103.9	**112.4**	107.7	111.5±1.1	110.4±3.9
hopper-me	99.8	111.9	86.1	110.7	111.7	104.6	112.0	111.8	113.1±0.1	**113.7**±0.8
walker2d-r	0.9	7.2	10.5	1.2	3.9	6.2	76.8	0.1	**82.0**±23.7	60.9±32.4
walker2d-m	87.2	84.5	84.3	83.2	100.9	104.6	103.6	116.1	122.1±1.3	**127.6**±2.5
walker2d-mr	89.7	88.1	88.2	94.3	106.8	100.0	115.8	113.8	118.2±3.4	**122.3**±3.9
walker2d-me	114.2	110.3	108.9	110.9	120.2	114.3	124.7	124.1	128.1±2.7	**131.4**±1.8
Total	812.6	800.7	786.6	832.4	935.0	931.1	1217.7	1142.7	1297.3	**1317.6**
95% CIs low	792.4	778.1	767.4	808.7	912.9	909.3	1209.8	1124.3	1290.3	1306.9
95% CIs high	832.7	823.3	805.5	856.0	957.1	953.0	1225.5	1161.1	1304.2	1328.3

Table 3: Online fine-tuning on AntMaze tasks. Mean ± std over 5 seeds; bold denotes the best mean.

Dataset (v2)	CQL	IQL	PEX	FamO2O	Cal-QL	OPT	OQACQL (Ours)
antmaze-umaze	97.6±0.9	91.2±4.8	96.0±2.9	93.6±3.9	99.4±0.9	98.4±1.1	**100.0**±0.0
antmaze-medium-play	96.0±2.7	81.4±4.9	84.8±4.0	86.4±2.2	93.8±3.1	96.0±2.3	**97.6**±2.1
antmaze-large-play	33.4±45.8	53.4±5.8	59.0±6.8	49.8±12.4	45.6±28.6	53.4±33.6	**84.4**±9.2
antmaze-umaze-diverse	55.2±49.0	39.8±28.0	11.8±5.8	57.4±22.8	95.8±2.2	95.8±1.8	**97.8**±1.9
antmaze-medium-diverse	96.6±1.3	78.4±4.0	83.8±3.5	88.0±4.9	95.6±1.1	96.2±1.9	**97.6**±2.7
antmaze-large-diverse	13.2±18.1	49.8±5.0	63.4±4.6	55.6±11.1	61.0±14.1	68.6±38.4	**86.8**±5.6
Total	392.0	394.0	398.8	430.8	491.2	508.4	**564.2**

4.1 Comparison with O2O RL Methods

Locomotion Results. Table 2 reports online fine-tuning on D4RL locomotion tasks and includes 95% confidence intervals for the total locomotion score. Additional results in Appendix C report per-task returns before and after fine-tuning and corresponding performance gains. OQA improves TD3+BC and CQL backbones, suggesting compatibility with different offline algorithms while improving fine-tuning performance. At the per-task level, OQATD3BC and OQACQL achieve the best or second-best mean on most tasks. On the aggregate metric, OQACQL achieves the highest total score and OQATD3BC attains the second-highest total. The 95% confidence intervals of both totals do not overlap with those of the baselines in Table 2. Learning curves are provided in Appendix D.

AntMaze and Maze2D Results. Although AntMaze is a standard sparse-reward benchmark, TD3+BC often gives weak initializations on medium/large mazes, so we follow SPA [12] and use Maze2D to evaluate TD3+BC-based fine-tuning in sparse-reward navigation. Tables 3 and 4 show consistent gains from OQA in both settings. On AntMaze, OQACQL achieves the best mean on all six tasks and the highest total score, with particularly large gains on large-play and large-diverse. On Maze2D, OQATD3BC also obtains the best mean on all six tasks and improves the total score from 1115.8 for SPA to 1155.5. These results indicate that OQA improves fine-tuning not only in dense-reward locomotion but also in sparse-reward navigation. We additionally evaluate OQATD3BC on AntMaze with randomly initialized actors and critics, using both offline and online data but no offline pretraining. The optimistic backup improves average performance in this setting, although performance remains limited on the large mazes; see Appendix E.3.

Adroit Results. Following OPT [15], Table 5 reports Adroit clone-task results. We exclude expert tasks because expert datasets already yield strong offline policies, leaving limited room for online

Table 4: Online fine-tuning on Maze2D tasks. Mean ± std over 5 seeds; bold denotes the best mean.

Dataset (v1)	TD3+BC	CPR	O2TD3	OPT	SPA	OQATD3BC (Ours)
maze2d-umaze	126.6±5.0	127.6±36.1	165.7±4.9	164.7±6.9	166.9±5.9	**168.2**±4.5
maze2d-medium	142.9±5.7	147.4±18.3	181.1±11.0	183.9±11.6	184.1±1.5	**187.4**±3.8
maze2d-large	195.7±17.9	161.9±19.3	180.6±45.6	186.0±46.5	235.0±8.1	**242.6**±6.1
maze2d-umaze-dense	49.7±5.8	87.5±20.1	150.0±5.0	151.2±5.5	152.6±5.9	**155.2**±3.1
maze2d-medium-dense	90.2±2.2	95.0±30.2	163.9±6.4	163.1±30.8	171.2±12.4	**181.0**±6.6
maze2d-large-dense	154.5±13.3	152.9±19.4	176.7±30.2	180.8±25.6	206.0±21.3	**221.1**±16.5
Total	759.6	772.4	1017.9	1029.7	1115.8	**1155.5**

Table 5: Online fine-tuning on Adroit tasks. Mean ± std over 5 seeds; bold denotes the best mean.

Dataset	CQL	Cal-QL	IQL	OPT	OQACQL
pen-clone-v1	101.7±9.2	0.0±0.0	71.7±1.8	136.2±4.8	**140.8**±5.7
hammer-clone-v1	0.2±0.0	0.1±0.2	1.6±0.7	121.9±1.6	**140.7**±3.6
door-clone-v1	-0.1±0.1	0.0±0.0	1.3±1.7	51.1±12.9	**104.7**±2.2
relocate-clone-v1	-0.2±0.1	0.0±0.0	0.7±0.4	-0.06±0.03	**2.8**±2.3
Adroit Total	101.6	0.1	75.3	309.1	**389.0**

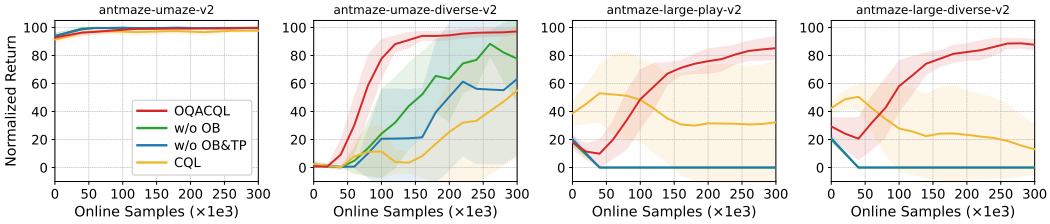


Figure 3: Ablation of OQA on AntMaze tasks. Curves show the normalized return over training.

fine-tuning gains. OQACQL achieves the best mean return on all four clone tasks and the highest aggregate score. While CQL and Cal-QL largely fail here, OQACQL substantially improves over them and also outperforms OPT, the strongest baseline in this domain. These results suggest that OQA can improve fine-tuning in high-dimensional dexterous manipulation.

4.2 Ablation Studies and Analysis

Component Ablation. OQA adds two modifications to offline backbones: a second offline-pretrained actor–critic pair and an α-controlled optimistic Bellman backup. We ablate them on AntMaze with three variants: w/o Optimistic Backup (w/o OB), w/o Optimistic Backup and Two-Pair Prior (w/o OB&TP), and CQL Fine-tuning (CQL). The w/o OB variant keeps both pretrained pairs but sets $\alpha = 0$, while w/o OB&TP removes the additional pair and cannot construct the two-prior optimistic backup. Figure 3 shows that OQACQL achieves the highest returns and converges faster than all ablated variants. Removing the optimistic backup degrades performance—returns drop to zero on harder large-maze tasks—indicating that this backup is a major driver of improvement. Maze2D results with paired sparse/dense rewards are in Appendix E.2 and Figure 12.

Choice of optimistic aggregation. In OQA, optimistic Bellman backup is implemented by aggregating value estimates from two pretrained critics using a maximum operator. To examine whether this design choice is essential, we consider alternative aggregations, including taking the mean or the minimum of the two Q-value estimates. As shown in Table 6, min aggregation, which further amplifies underestimation, collapses to zero on most AntMaze tasks, while the mean aggregation also collapses to zero on more challenging large tasks and underperforms overall. In contrast, OQACQL achieves the best AntMaze total return and remains strong across different environments. We further study how the number of pretrained actor–critic pairs affects performance by increasing the number of pairs beyond two. As shown in Table 7, using more pairs (e.g., $n = 3, 4, 5$) does not yield additional

Table 6: Optimistic target variants on AntMaze.

Dataset(v2)	Min	Mean	OQACQL
umaze	99.8±0.4	99.8±0.4	**100.0**±0.0
medium-play	0.0±0.0	**99.6**±0.5	97.6±2.1
large-play	0.0±0.0	0.0±0.0	**84.4**±9.2
umaze-diverse	0.0±0.0	79.4±36.2	**97.8**±1.9
medium-diverse	0.0±0.0	**98.4**±2.1	97.6±2.7
large-diverse	0.0±0.0	0.0±0.0	**86.8**±5.6
Total	99.8	377.2	**564.2**

Table 7: Effect of varying n agents on Maze2D.

Dataset(v1)	n=2	n=3	n=4	n=5
umaze	168.2	164.7	**169.6**	168.2
medium	187.4	188.3	**188.4**	188.0
large	**242.6**	217.8	232.1	234.8
umaze-dense	155.2	**155.3**	153.1	154.7
medium-dense	181.0	182.3	182.4	**183.9**
large-dense	221.1	**227.3**	205.1	216.0
Total	**1155.5**	1135.7	1130.7	1145.6

gains over the two-pair design, while incurring extra computational overhead. Therefore, we adopt $n = 2$ as a favorable trade-off between performance and efficiency.

Sensitivity to the optimism coefficient α. The coefficient α controls how strongly OQA interpolates the optimistic Bellman backup into the conservative target during online fine-tuning. In the main experiments, we use $\alpha = 1$ for navigation tasks and smaller values for locomotion tasks, such as $\alpha = 0.7$ for HalfCheetah and $\alpha = 0.2$ for Walker/Hopper. Figure 4 sweeps α and compares each setting with $\alpha = 0$, which disables the optimistic backup. The results show that moderate optimism ($\alpha \in [0.2, 0.7]$) generally improves performance over $\alpha = 0$, while overly large values can reduce stability. Overall, OQA is not overly sensitive to α, and a nonzero level of optimism within a reasonable range is beneficial.

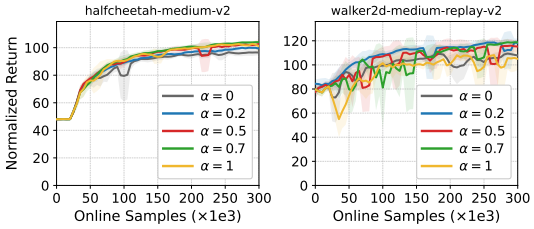


Figure 4: Sensitivity to the optimism coefficient α in OQATD3BC.

Per-Transition Min-Q Correction. To better understand the mechanism underlying OQA's effectiveness, we investigate how diversity between independently pretrained actor–critic branches contributes to the optimistic correction at the transition level. Specifically, we fix the actor–critic pairs pretrained with different retention probabilities p and evaluate them on the same halfcheetah-medium-expert-v2 offline dataset, without online interaction. This controlled evaluation allows us to examine the transition-wise disagreement between branches and the corresponding corrections introduced by the optimistic backup.

Table 8 shows that similar global Min-Q statistics can conceal differences at individual transitions. By taking the pointwise maximum of the two branches' Min-Q estimates, the optimistic backup selectively raises the lower estimate at transitions where the branches disagree, yielding nonzero average corrections even when their global statistics are similar. The average pointwise disagreement generally decreases with increasing p, as higher retention probabilities reduce the perturbation introduced by data masking. However, this trend is not strictly monotonic, since branch disagreement is also influenced by independent initialization, stochastic optimization, and differences between the learned actors. Importantly, disagreement can persist even at $p = 1.0$, indicating that data masking augments the diversity between independently trained branches rather than creating it from scratch.

Although the resulting transition-wise corrections are small relative to the overall Q-value scale, their effects can propagate through repeated bootstrapped Bellman updates and accumulate during fine-tuning, potentially exerting a substantial influence on subsequent Q-value estimates. Additional sensitivity analysis of the retention probability across seven tasks is provided in Appendix E.6.

CQL Ablation Studies. We test whether the gains of OQACQL can be reproduced by modifying only the actor objective or removing the conservative regularizer from CQL. First, we retain the CQL critic update and adopt the OQACQL actor objective with its behavior-cloning log-likelihood term. This variant, denoted *CQL w/ BC Actor*, uses a single actor–critic pair without the optimistic backup. Table 9 shows that this change does not improve average performance over CQL and falls well below OQACQL. Second, we remove only the conservative Q-regularizer during CQL fine-tuning, retaining its original actor update and single actor–critic pair. This improves performance

Table 8: Min-Q statistics over 1,998,000 offline transitions (mean $\pm$ std across transitions). Gap is the absolute branch difference; Δ_i is the raw correction from taking the pointwise maximum relative to branch i's Min-Q.

Retention p	Min-Q (branch 0)	Min-Q (branch 1)	Pointwise gap	Correction Δ_0	Correction Δ_1
0.3	641.46 $\pm$ 326.93	640.07 $\pm$ 327.39	7.60 $\pm$ 14.67	3.11 $\pm$ 11.19	4.50 $\pm$ 10.86
0.5	691.78 $\pm$ 296.66	692.31 $\pm$ 294.27	6.54 $\pm$ 10.93	3.54 $\pm$ 9.23	3.01 $\pm$ 7.44
0.7	710.07 $\pm$ 285.63	716.03 $\pm$ 283.91	7.48 $\pm$ 9.33	6.72 $\pm$ 8.12	0.75 $\pm$ 5.58
0.9	691.87 $\pm$ 292.96	694.60 $\pm$ 293.16	5.47 $\pm$ 8.67	4.10 $\pm$ 7.23	1.37 $\pm$ 5.85
1.0	727.13 $\pm$ 277.62	727.96 $\pm$ 278.47	4.51 $\pm$ 7.25	2.67 $\pm$ 5.94	1.84 $\pm$ 5.20

Table 9: Actor-only CQL comparison on eight locomotion tasks. CQL w/ BC Actor modifies only the actor objective while retaining the CQL critic update. Bold denotes the highest mean in each row.

Task	CQL	CQL w/ BC Actor	OQACQL
halfcheetah-r	13.6	2.3	**93.7**
halfcheetah-m	66.1	52.1	**108.3**
halfcheetah-mr	37.2	51.4	**102.2**
halfcheetah-me	96.8	96.5	**126.3**
hopper-m	73.2	67.6	**111.1**
hopper-mr	103.0	100.7	**110.4**
walker2d-mr	88.1	92.9	**122.3**
walker2d-me	110.3	110.4	**131.4**
Average	73.5	71.7	**113.2**

on hopper-random but degrades it on the other three Hopper datasets, leaving the average nearly unchanged (Appendix E.7). Neither modification alone reproduces the gains of OQACQL. Together with the w/o OB ablation in Figure 3, these results support the importance of the optimistic Bellman backup.

Evaluating Critic Calibration. To assess whether OQA alleviates critic underestimation, we directly evaluate Q-value estimation accuracy during online fine-tuning. Following prior work [15], we adopt a reference-critic protocol by training a purely online SAC agent to convergence and treating its critic as a strong empirical reference. We then compare the Q estimates of the fine-tuned critic against this reference on identical state–action pairs. As shown in Figure 5, fine-tuning with offline backbones exhibits severe negative bias, whereas OQA substantially mitigates underestimation. Additional results are deferred to Appendix E.1.

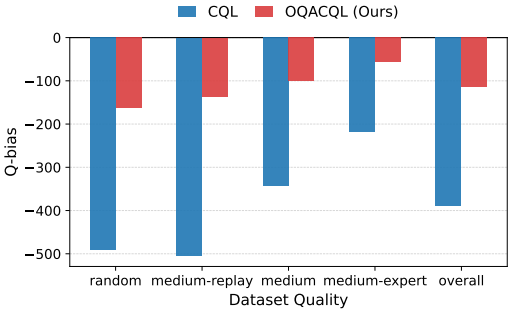


Figure 5: Aggregated Q-bias across datasets of varying quality, comparing CQL and OQACQL.

5 Conclusion

In this work, we propose Optimistic Q-value Adaptation (OQA), an efficient O2O RL framework that can be applied on top of different offline RL algorithms. By constructing an additional offline-pretrained actor–critic pair, OQA constructs a controlled optimistic Bellman backup that mitigates overly pessimistic offline critics and improves online fine-tuning performance. Across diverse D4RL domains, OQA achieves higher fine-tuning performance and larger overall improvements than strong O2O RL baselines. Ablation studies further show that optimistic adaptation is the main driver of the gains and consistently speeds up learning progress. A current limitation is that the optimism coefficient α is set as a fixed hyperparameter throughout fine-tuning. Learning an adaptive schedule for α could better help the critic mitigate compounding underestimation as training progresses, and we leave this direction for future work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 62407010, and the Science and Technology Development Plan Project of Jilin Province of China under Grant No. 20240602108RC.

References

[1] Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pages 1577–1594, 2023.

[2] Jiaheng Feng, Mingxiao Feng, Haolin Song, Wengang Zhou, and Houqiang Li. SUF: Stabilized unconstrained fine-tuning for offline-to-online reinforcement learning. In *AAAI Conference on Artificial Intelligence*, pages 11961–11969, 2024.

[3] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[4] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Annual Conference on Neural Information Processing Systems*, pages 20132–20145, 2021.

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

[6] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1856–1865, 2018.

[7] Hao Hu, Yiqin Yang, Jianing Ye, Chengjie Wu, Ziqing Mai, Yujing Hu, Tangjie Lv, Changjie Fan, Qianchuan Zhao, and Chongjie Zhang. Bayesian design principles for offline-to-online reinforcement learning. In *International Conference on Machine Learning*, pages 19491–19515, 2024.

[8] Rui Kong, Chenyang Wu, Chen Xiao Gao, Zongzhang Zhang, and Ming Li. Efficient and stable offline-to-online reinforcement learning via continual policy revitalization. In *International Joint Conference on Artificial Intelligence*, pages 4317–4325, 2024.

[9] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022.

[10] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Annual Conference on Neural Information Processing Systems*, pages 1179–1191, 2020.

[11] Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic Q-ensemble. In *Conference on Robot Learning*, pages 1702–1712, 2021.

[12] Songlin Li, Wei Xiao, Hao Wu, Xiaodan Zhang, Daolong An, and Shuai Lü. State proficiency-based adaptive fine-tuning for offline-to-online reinforcement learning. In *AAAI Conference on Artificial Intelligence*, volume 40, pages 23169–23176, 2026.

[13] Qin Wen Luo, Ming Kun Xie, Ye Wen Wang, and Sheng Jun Huang. Optimistic critic reconstruction and constrained fine-tuning for general offline-to-online RL. In *Annual Conference on Neural Information Processing Systems*, pages 108167–108207, 2024.

[14] Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-QL: Calibrated offline RL pre-training for efficient online fine-tuning. In *Annual Conference on Neural Information Processing Systems*, pages 62244–62269, 2023.

[15] Yongjae Shin, Jeonghye Kim, Whiyoung Jung, Sunghoon Hong, Deunsol Yoon, Youngsoo Jang, Geon-Hyeong Kim, Jongseong Chae, Youngchul Sung, Kanghoon Lee, and Woohyung Lim. Online pre-training for offline-to-online reinforcement learning. In *International Conference on Machine Learning*, 2025.

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

[17] Shenzhi Wang, Qisen Yang, Jiawei Gao, Matthieu Gaetan Lin, Hao Chen, Liwei Wu, Ning Jia, Shiji Song, and Gao Huang. Train once, get a family: State-adaptive balances for offline-to-online reinforcement learning. In *Annual Conference on Neural Information Processing Systems*, pages 47081–47104, 2023.

[18] Zishun Yu and Xinhua Zhang. Actor-critic alignment for offline-to-online reinforcement learning. In *International Conference on Machine Learning*, pages 40452–40474, 2023.

[19] Haichao Zhang, Wei Xu, and Haonan Yu. Policy expansion for bridging offline-to-online reinforcement learning. In *International Conference on Learning Representations*, 2023.

[20] Yinmin Zhang, Jie Liu, Chuming Li, Yazhe Niu, Yaodong Yang, Yu Liu, and Wanli Ouyang. A perspective of Q-value estimation on offline-to-online reinforcement learning. In *AAAI Conference on Artificial Intelligence*, pages 16908–16916, 2024.

[21] Kai Zhao, Jianye Hao, Yi Ma, Jinyi Liu, Yan Zheng, and Zhaopeng Meng. ENOTO: Improving offline-to-online reinforcement learning with Q-ensembles. In *International Joint Conference on Artificial Intelligence*, pages 5563–5571, 2024.

[22] Yi Zhao, Rinu Boney, Alexander Ilin, Juho Kannala, and Joni Pajarinen. Adaptive behavior cloning regularization for stable offline-to-online reinforcement learning. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2022.

[23] Han Zheng, Xufang Luo, Pengfei Wei, Xuan Song, Dongsheng Li, and Jing Jiang. Adaptive policy learning for offline-to-online reinforcement learning. In *AAAI Conference on Artificial Intelligence*, pages 11372–11380, 2023.

[24] Zhiyuan Zhou, Andy Peng, Qiyang Li, Sergey Levine, and Aviral Kumar. Efficient online reinforcement learning fine-tuning need not retain offline data. In *International Conference on Learning Representations*, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state the main contributions of OQA, including the observation of critic underestimation beyond CQL-style methods, the proposed optimistic Bellman backup, and empirical improvements on D4RL locomotion, Maze2D, AntMaze, and Adroit benchmarks. These claims are supported by the method section, main experimental tables, ablations, and additional analyses in the appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The conclusion discusses a current limitation of OQA, namely that the optimism coefficient α is fixed throughout fine-tuning. The paper also analyzes sensitivity to α, retention probability, UTD, and computational cost in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

   Justification: The paper does not present formal theoretical results or proofs. The equations define the algorithmic objectives and Bellman backup operators used by OQA.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: The paper specifies the benchmark domains, datasets, offline pretraining steps, online fine-tuning budget, number of seeds, replay-buffer mixing strategy, UTD setting, backbone implementations, and key hyperparameters in Section A and Appendix B. Algorithm 1 and the method section describe the OQA training procedure.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: The experiments use public D4RL datasets, and we provide anonymized code and run scripts in the supplemental material. The instructions include data preparation, offline pretraining, online fine-tuning, and evaluation commands for reproducing the main OQA results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Appendix A describes the evaluated D4RL domains and datasets, while Appendix B provides implementation details, baseline codebases, learning rates, UTD settings, replay mixing, reward scaling, policy-constraint coefficients, and fine-tuning protocols.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

Justification: The main experimental results report mean performance with standard deviation over 5 random seeds, as stated in the captions of Tables 2, 3, 4, and 5. Table 2 additionally reports 95% confidence intervals for aggregate locomotion scores, and the learning curves in Appendix D use shaded regions to indicate standard deviation across seeds.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Appendix E.5 reports the compute worker used in our experiments, namely one NVIDIA RTX 3090 GPU per run, and provides a representative wall-clock comparison for online fine-tuning.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: We have reviewed the NeurIPS Code of Ethics. The work uses public simulation benchmarks and public reference implementations, does not involve human subjects, private data, or sensitive personal information, and is intended for algorithmic research on offline-to-online reinforcement learning.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: Appendix G discusses the broader impacts of this work. It addresses potential positive impacts, such as improving data-efficient policy adaptation in robotics and sequential decision-making systems, and potential negative impacts, such as risks in safety-critical control settings without adequate validation. It also notes that real-world deployment should be combined with task-specific safety constraints and deployment-time monitoring.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper does not release high-risk models, scraped datasets, pre-trained language models, image generators, or other assets with a high risk of misuse. The experiments are conducted on public simulation benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: The paper properly credits the creators of the existing assets used in this work, including the D4RL benchmark, MuJoCo environments, and public baseline implementations. We cite the corresponding papers and provide URLs for the reference codebases, and we use these assets in accordance with their licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: The paper introduces an implementation of OQA and provides anonymized code and run scripts in the supplemental material. The accompanying documentation describes data preparation, offline pretraining, online fine-tuning, and evaluation commands for reproducing the main results.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing, user studies, human-subject experiments, or participant compensation.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve human subjects, crowdsourcing, or collection of personal data, so IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: The core method, experiments, and analysis do not use LLMs as an important, original, or non-standard component. Any use of LLMs for writing or editing assistance, if applicable, does not affect the methodology or scientific claims.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

Appendix

A Detailed Description of Experimental Setup

A.1 MuJoCo Locomotion

The MuJoCo domain comprises continuous-control locomotion tasks with dense rewards. Following prior O2O RL works, we conduct experiments on the `halfcheetah`, `hopper`, and `walker2d` environments. For each environment, we use the "-v2" versions of the D4RL datasets with four quality levels: `random`, `medium`, `medium-replay`, and `medium-expert`. These datasets vary in both behavior policy quality and state–action coverage, allowing us to evaluate our method under diverse dataset conditions.

A.2 AntMaze Navigation

AntMaze is a sparse-reward navigation benchmark in which an ant robot must move from a start location to a designated goal region within a maze. The agent receives a reward of $+1$ only when it successfully reaches the goal and 0 otherwise. We consider three maze sizes—`umaze`, `medium`, and `large`—and use the "-v2" versions of the D4RL datasets: `umaze`, `umaze-diverse`, `medium-play`, `medium-diverse`, `large-play`, and `large-diverse`. These datasets vary in terms of exploration patterns and trajectory richness, providing a challenging testbed for offline-to-online policy improvement in sparse-reward settings.

A.3 Maze2D Navigation

Maze2D contains planar navigation tasks where an agent moves in a 2D maze with continuous actions. Although AntMaze is a standard sparse-reward benchmark, prior work [13] shows that TD3+BC often learns poor offline policies on harder AntMaze tasks, making subsequent fine-tuning only marginally better than training from scratch. To better evaluate TD3+BC-style O2O algorithms in sparse-reward navigation, we include Maze2D and use the D4RL `umaze`, `medium`, `large`, `umaze-dense`, `medium-dense`, and `large-dense` datasets, all with "-v1" versions.

A.4 Adroit Manipulation

Adroit contains high-dimensional dexterous manipulation tasks with a multi-finger robotic hand. Following prior O2O RL studies, we evaluate `pen`, `hammer`, `door`, and `relocate` using the D4RL "-v1" cloned datasets. These tasks are challenging due to complex contact dynamics, sparse rewards, and limited state–action coverage from imperfect demonstrations.

B Implementation Details

B.1 Implementation Details of Baselines

We use all public datasets, environments, and reference implementations in accordance with their respective licenses and terms of use. To ensure a fair and reproducible comparison, we follow reference implementations whenever possible. We adopt the CORL implementation[2] for TD3+BC [4], CQL [10], and IQL [9]. For other baselines, we use the official open-source codebases: O2TD3 [13][3], Cal-QL [14][4], CPR [8][5], PEX [19][6], FamO2O [17][7], RLPD [1][8], SPA [12][9], and OPT [15][10]. For OPT, we use TD3+BC as the offline backbone on MuJoCo locomotion and Maze2D, and on AntMaze

[2] https://github.com/tinkoff-ai/CORL
[3] https://github.com/QinwenLuo/OCR-CFT
[4] https://github.com/nakamotoo/Cal-QL
[5] https://github.com/LAMDA-RL/CPR
[6] https://github.com/Haichao-Zhang/PEX
[7] https://github.com/LeapLabTHU/FamO2O
[8] https://github.com/ikostrikov/rlpd
[9] https://github.com/mikiya1213/SPA
[10] https://github.com/LGAI-Research/opt

we report the better-performing official variant (SPOT+OPT) among IQL+OPT and SPOT+OPT. For Adroit, we directly report the OPT results from the original paper [15], following its clone-task evaluation protocol. Unless stated otherwise, we keep hyperparameters consistent with the corresponding reference implementations. For methods whose original pipelines rely on high-UTD training, we standardize the update-to-data ratio (UTD) to 5 to ensure a fair and stable comparison. For MuJoCo locomotion and Maze2D, we use learning rates of 3×10^{-4} for both actor and critic. For AntMaze, we follow the Cal-QL setup with critic learning rate 3×10^{-4} and actor learning rate 1×10^{-4}, and apply sparse reward scaling ($+5$ at the goal and -5 per step). During fine-tuning, all methods except OPT construct each minibatch by uniformly mixing offline and online samples (50%/50%), while OPT follows its original balanced replay strategy that prioritizes samples close to the current policy distribution.

B.2 Implementation Details of OQA

For Adroit tasks, we use $\alpha = 0.2$ and $\lambda_{\mathrm{BC}} = 1$. For navigation tasks, since navigation environments typically require stronger exploration, we set the policy constraint coefficient $\lambda_{\mathrm{BC}} = 0$ during online fine-tuning. For locomotion tasks, we adopt moderate policy constraints to balance stable control and online adaptation. Specifically, for OQATD3BC we set $\lambda_{\mathrm{BC}} = 1.0$ on medium and medium-expert datasets, and $\lambda_{\mathrm{BC}} = 0.5$ on the remaining locomotion datasets. For OQACQL, we use the same λ_{BC} values as OQATD3BC on all non-random datasets, while a smaller value $\lambda_{\mathrm{BC}} = 0.1$ is used on random datasets to avoid excessive conservatism. Unless otherwise stated, we set the retention probability to $p = 0.9$ for constructing the perturbed offline datasets in all main experiments.

C Additional Experimental Results

C.1 Aggregate Statistical Metrics

Figure 6 reports aggregate performance using four statistical metrics: mean, interquartile mean (IQM), median, and optimality gap (OG). Beyond the average return, IQM and median provide more robust summaries by reducing the influence of extreme runs, while OG measures the remaining gap to high performance. Across these metrics, OQA consistently ranks ahead of the baseline methods, indicating that its advantage is not driven only by a small number of high-scoring tasks. The improvement in IQM and median further suggests that OQA provides more reliable gains across the task distribution.

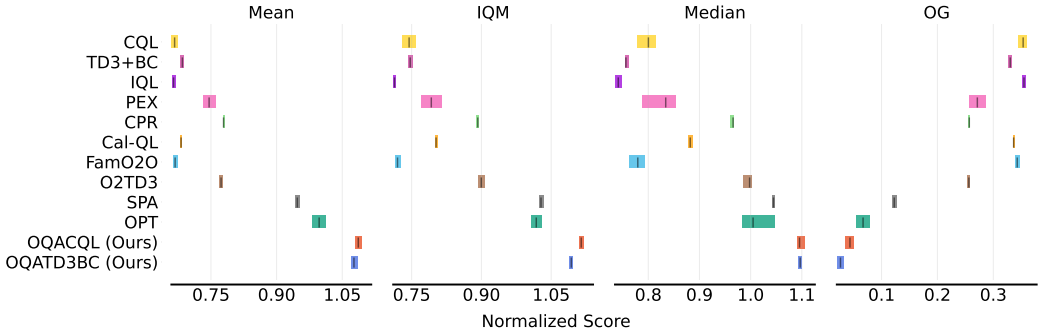


Figure 6: Aggregate normalized scores under mean, IQM, median, and optimality gap (OG). Higher is better for mean, IQM, and median, while lower is better for OG.

C.2 Online Fine-Tuning Performance Gains

Table 2 in the main paper reports the final returns after online fine-tuning. In addition to final performance, O2O evaluation also considers the gain achieved during online fine-tuning. Therefore, we report the mean return at initialization and at the end of fine-tuning in Table 10 and Table 11. For clarity, we organize the results by offline backbone, separating TD3+BC-based methods from CQL/IQL-based methods, with OPT reported under the backbone used in its corresponding setup. Across both MuJoCo locomotion and navigation tasks, OQA achieves the best aggregate totals and

Table 10: Normalized mean returns before → after online fine-tuning on D4RL locomotion and Maze2D tasks. Bold denotes the highest post fine-tuning normalized mean return.

Dataset (v2)	TD3+BC	CPR	O2TD3	OPT	SPA	OQATD3BC
halfcheetah-random-v2	12.1 → 45.4	11.6 → 47.7	11.0 → 42.9	11.0 → 85.0	11.7 → 83.4	11.3 → **101.6**
halfcheetah-medium-v2	48.0 → 64.4	47.6 → 70.6	48.2 → 74.5	47.9 → 96.0	48.0 → 91.7	48.0 → **104.1**
halfcheetah-medium-replay-v2	44.4 → 52.9	44.4 → 68.6	44.9 → 67.4	44.2 → 92.3	44.4 → 84.7	44.5 → **95.3**
halfcheetah-medium-expert-v2	92.3 → 94.6	71.1 → 91.9	89.9 → 99.7	92.7 → 93.4	92.3 → 100.7	94.1 → **105.9**
hopper-random-v2	8.5 → 10.2	9.3 → 9.4	8.1 → 8.7	7.8 → 96.2	8.5 → 97.7	10.3 → **103.9**
hopper-medium-v2	60.3 → 60.5	58.2 → 100.8	60.0 → 104.5	57.9 → 109.5	60.3 → 110.9	59.8 → **111.4**
hopper-medium-replay-v2	51.0 → 92.7	57.5 → 102.4	46.2 → 103.9	60.8 → **112.4**	51.0 → 107.7	59.9 → 111.5
hopper-medium-expert-v2	99.3 → 99.8	97.7 → 111.7	81.8 → 104.6	86.9 → 112.0	99.3 → 111.8	92.9 → **113.1**
walker2d-random-v2	3.8 → 0.9	0.3 → 3.9	0.9 → 6.2	1.8 → 76.8	4.2 → 0.1	2.9 → **82.0**
walker2d-medium-v2	83.2 → 87.2	85.5 → 100.9	85.7 → 104.6	79.6 → 103.6	83.2 → 116.1	81.6 → **122.1**
walker2d-medium-replay-v2	81.3 → 89.7	78.9 → 106.8	67.7 → 100.0	80.2 → 115.8	81.3 → 113.8	85.5 → **118.2**
walker2d-medium-expert-v2	110.2 → 114.2	111.0 → 120.2	110.2 → 114.3	110.3 → 124.7	110.2 → 124.1	110.7 → **128.1**
MuJoCo Total	694.2 → 812.6	673.1 → 935.0	654.8 → 931.1	681.2 → 1217.7	694.3 → 1142.7	701.6 → **1297.3**
MuJoCo Improvement	118.4	261.9	276.3	536.5	448.4	**595.7**
maze2d-umaze-v1	42.2 → 126.6	43.0 → 127.6	37.8 → 165.7	37.2 → 164.7	43.6 → 166.9	17.1 → **168.2**
maze2d-medium-v1	36.3 → 142.9	66.6 → 147.4	31.4 → 181.1	41.6 → 183.9	64.2 → 184.1	51.0 → **187.4**
maze2d-large-v1	83.1 → 195.7	69.3 → 161.9	123.9 → 180.6	54.9 → 186.0	90.7 → 235.0	73.7 → **242.6**
maze2d-umaze-dense-v1	34.1 → 49.7	28.6 → 87.5	35.8 → 150.0	26.9 → 151.2	35.6 → 152.6	28.0 → **155.2**
maze2d-medium-dense-v1	39.6 → 90.2	43.8 → 95.0	43.8 → 163.9	33.1 → 163.1	55.6 → 171.2	37.8 → **181.0**
maze2d-large-dense-v1	94.3 → 154.5	70.9 → 152.9	63.6 → 176.7	68.6 → 180.8	72.7 → 206.0	92.7 → **221.1**
Maze2D Total	329.6 → 759.6	322.2 → 772.4	336.3 → 1017.9	262.4 → 1029.7	362.5 → 1115.8	300.3 → **1155.5**
Maze2D Improvement	430.0	450.1	681.7	767.4	753.3	**855.2**

Table 11: Normalized mean returns before → after online fine-tuning on D4RL locomotion and AntMaze tasks. Bold denotes the highest post fine-tuning normalized mean return.

Dataset (v2)	CQL	IQL	PEX	FamO2O	Cal-QL	OPT	OQACQL
halfcheetah-r	12.5 → 13.6	10.4 → 34.2	13.3 → 54.6	8.2 → 40.0	20.1 → 4.5	11.0 → 85.0	12.9 → **93.7**
halfcheetah-m	47.0 → 66.1	48.2 → 50.1	48.2 → 70.0	47.9 → 49.6	47.8 → 68.0	47.9 → 96.0	46.6 → **108.3**
halfcheetah-mr	41.1 → 37.2	44.1 → 46.1	43.5 → 55.9	42.2 → 47.7	46.4 → 52.1	44.2 → 92.3	45.1 → **102.2**
halfcheetah-me	94.3 → 96.8	93.2 → 93.1	90.7 → 90.1	80.0 → 87.3	67.6 → 96.6	92.7 → 93.4	89.0 → **126.3**
hopper-r	8.1 → 8.8	7.2 → 10.7	8.2 → 39.5	8.1 → 10.4	7.6 → 12	7.8 → 96.2	8.2 → **109.9**
hopper-m	59.3 → 73.2	62.6 → 65.5	60.0 → 89.9	69.7 → 71.8	60.2 → 99.0	57.9 → 109.5	59.0 → **111.1**
hopper-mr	86.6 → 103	87.1 → 92.2	76.9 → 86.1	84.2 → 101.8	99.1 → 99.9	60.8 → **112.4**	85.7 → 110.4
hopper-me	72.3 → 111.9	87.8 → 107.5	38.9 → 97.4	48.2 → 86.1	105.7 → 110.7	86.9 → 112.0	87.9 → **113.7**
walker2d-r	2.5 → 7.2	3.5 → 7.3	5.8 → 12.7	6.2 → 10.5	0.4 → 1.2	1.8 → **76.8**	1.5 → 60.9
walker2d-m	81.3 → 84.5	81.2 → 81.9	74.8 → 91.7	77.2 → 84.3	80.3 → 83.2	79.6 → 103.6	81.0 → **127.6**
walker2d-mr	81.4 → 88.1	79.6 → 94.2	72.8 → 98.8	59.1 → 88.2	84.4 → 94.3	80.2 → 115.8	74.7 → **122.3**
walker2d-me	108.8 → 110.3	111.6 → 114.3	109.0 → 113.0	109.4 → 108.9	108.1 → 110.9	110.3 → 124.7	106.9 → **131.4**
MuJoCo Total	695.3 → 800.7	716.6 → 797.1	643.0 → 899.6	640.2 → 786.6	727.7 → 832.4	681.2 → 1217.7	698.4 → **1317.6**
MuJoCo Improvement	105.4	80.5	256.6	146.4	104.7	536.5	**619.2**
umaze	89.8 → 97.6	89.0 → 91.2	75.0 → 96.0	90.4 → 93.6	87.0 → 99.4	92.6 → 98.4	91.6 → **100.0**
medium-play	50.0 → 96.0	73.2 → 81.4	61.2 → 84.8	68.8 → 86.4	70.0 → 93.8	64.8 → 96.0	54.4 → **97.6**
large-play	31.0 → 33.4	40.6 → 53.4	3.6 → 59.0	40.6 → 49.8	21.4 → 45.6	36.8 → 53.4	24.0 → **84.4**
umaze-diverse	3.4 → 55.2	61.4 → 39.8	0.2 → 11.8	71.2 → 57.4	43.8 → 95.8	48.2 → 95.8	1.2 → **97.8**
medium-diverse	63.0 → 96.6	71.4 → 78.4	49.0 → 83.8	67.8 → 88.0	62.8 → 95.6	68.0 → 96.2	67.4 → **97.6**
large-diverse	36.0 → 13.2	49.8 → 49.8	0.6 → 63.4	47.0 → 55.6	37.2 → 61.0	23.4 → 68.6	37.0 → **86.8**
AntMaze Total	273.2 → 392.0	385.4 → 394.0	189.6 → 398.8	385.8 → 430.8	322.2 → 491.2	333.8 → 508.4	275.6 → **564.2**
AntMaze Improvement	118.8	8.6	209.2	45.0	169.0	174.6	**288.6**

the largest improvement. Overall, the consistent increase in both final returns and O2O gains supports OQA as a practical mechanism for improving online fine-tuning across domains.

C.3 Comparison with Non-standard O2O RL Baselines

In addition to standard O2O RL baselines, we compare OQA with two related protocols that use offline data differently from the conventional offline-to-online setting. RLPD [1] does not include an offline pre-training stage; instead, it initializes the replay buffer with offline data and directly performs online SAC training. WSRL [24] follows the opposite design: it first obtains an offline-pretrained policy, but discards the offline dataset during online fine-tuning and warms up the replay buffer with rollouts collected by the frozen offline policy. Thus, RLPD removes the offline training stage, while WSRL removes offline data retention during fine-tuning.

We compare these methods against OQACQL on D4RL locomotion tasks. All RLPD and WSRL results follow their reference settings. As shown in Table 12, OQACQL achieves the best performance on most tasks and obtains a substantially higher total score. This suggests that combining offline pre-training with continued offline-data retention, together with OQA critic adaptation, provides a stronger fine-tuning signal than these alternative O2O protocols.

Table 12: Comparison with non-standard O2O baselines on D4RL locomotion. Mean ± std over 5 seeds; bold indicates the best mean.

Dataset (v2)	RLPD	WSRL	OQACQL
halfcheetah-r	86.7±5.4	77.2±2.6	**93.7**±6.7
halfcheetah-m	89.0±1.2	89.9±2.0	**108.3**±4.1
halfcheetah-mr	85.7±1.7	84.2±2.4	**102.2**±1.4
halfcheetah-me	97.4±1.6	95.5±1.6	**126.3**±2.9
hopper-r	86.8±31.2	78.6±20.3	**109.9**±3.7
hopper-m	97.1±14.8	66.1±23.8	**111.1**±1.8
hopper-mr	80.7±28.4	57.8±16.7	**110.4**±3.9
hopper-me	89.5±26.3	80.4±9.5	**113.7**±0.8
walker2d-r	80.1±16.3	**83.5**±10.7	60.9±32.4
walker2d-m	107.9±9.0	107.4±6.2	**127.6**±2.5
walker2d-mr	109.8±1.7	108.9±3.4	**122.3**±3.9
walker2d-me	123.4±0.8	106.9±12.1	**131.4**±1.8
Total	1134.1	1036.4	**1317.6**

D Learning Curves

Across MuJoCo locomotion, Figure 7 shows that OQATD3BC converges faster and achieves higher final returns than the baselines. Figure 8 shows similar gains on most Maze2D tasks, indicating effectiveness in sparse-reward settings. Figure 9 further shows that OQACQL improves AntMaze performance, especially on harder variants such as `large-diverse` and `large-play`. Figure 10 shows competitive fine-tuning on Adroit, suggesting that OQA can also generalize to dexterous manipulation.

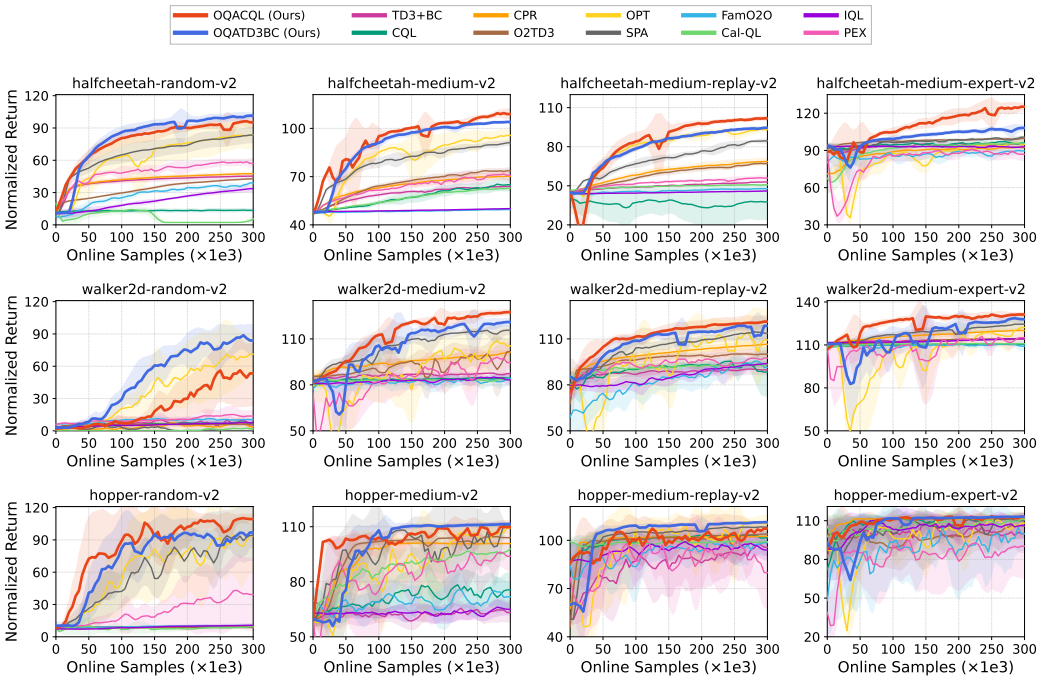


Figure 7: Learning curves on D4RL Locomotion tasks during online fine-tuning. Curves show the mean normalized return over 5 seeds, with shaded regions indicating standard deviation.

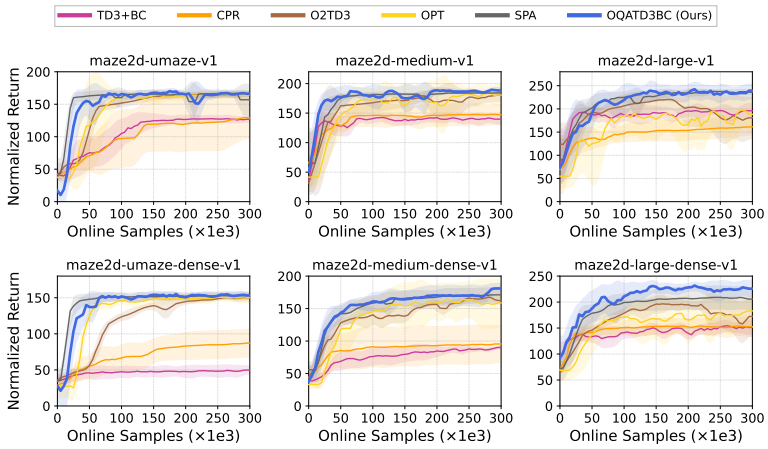


Figure 8: Learning curves on D4RL Maze2D tasks during online fine-tuning. Curves show the mean normalized return over 5 seeds, with shaded regions indicating standard deviation.

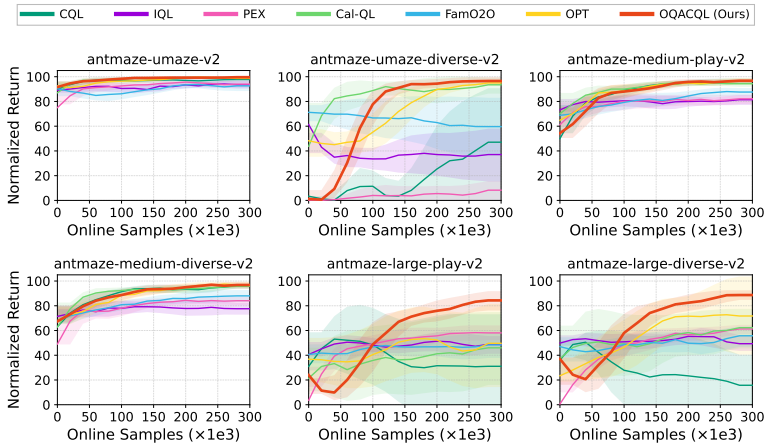


Figure 9: Learning curves on D4RL AntMaze tasks during online fine-tuning. Curves show the mean normalized return over 5 seeds, with shaded regions indicating standard deviation.

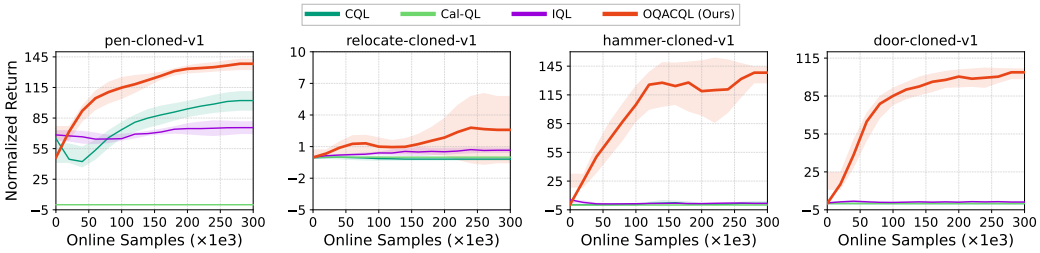


Figure 10: Learning curves on D4RL Adroit tasks during online fine-tuning. Curves show the mean normalized return over 5 seeds, with shaded regions indicating standard deviation.

E Additional Analysis

E.1 Detailed Q-Bias Analysis on Locomotion Benchmarks

This section reports more detailed results for the Q-bias evaluation. Every 5k online samples, we reset the environment to sample states and generate actions using both the current and reference policies. We then compute Q-bias by comparing the minimum of the twin Q-values from the fine-tuned critic against those from the reference critic on the same state–action pairs.

Figure 11 presents Q-bias during online fine-tuning across all locomotion benchmarks. Fine-tuning with offline backbones consistently exhibits pronounced negative Q-bias across environments with varying dataset quality, and can even worsen as training progresses. In contrast, OQA substantially reduces underestimation and yields less biased Q-value estimates over training.

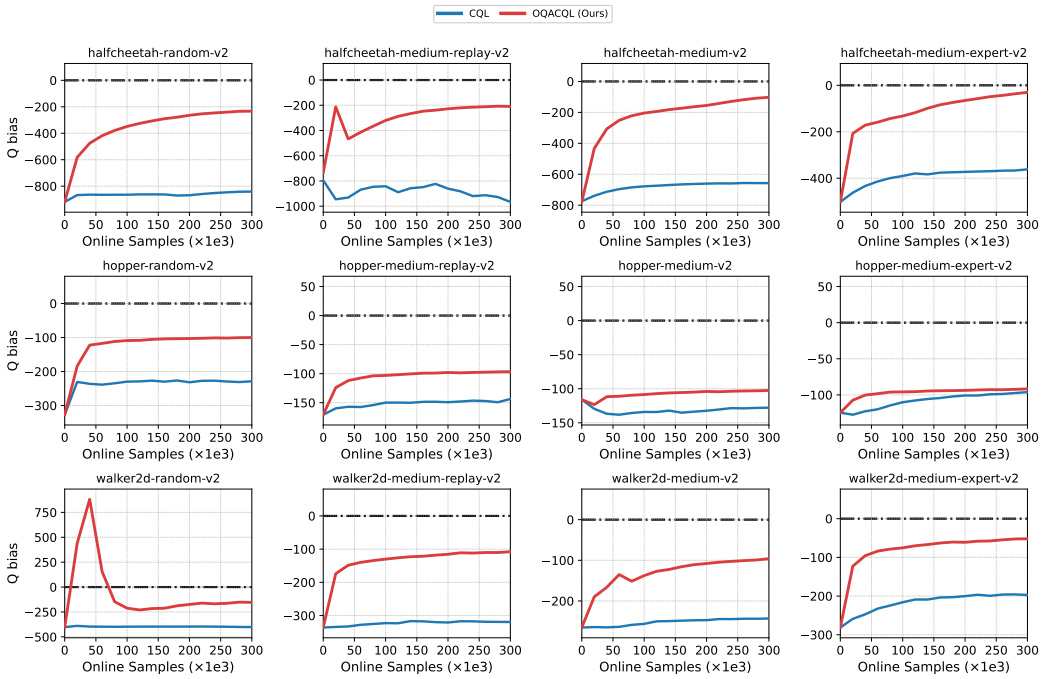


Figure 11: Curves of Q-bias during online fine-tuning on MuJoCo locomotion.

E.2 Sparse vs. Dense Reward Ablation

Figure 12 provides paired comparisons on Maze2D, which features identical maze layouts under both sparse and dense rewards. OQATD3BC (red) consistently achieves higher returns and faster convergence than TD3+BC (yellow), the variant without the optimistic backup (w/o OB, green), and the variant without both the optimistic backup and the two-pair prior (w/o OB&TP, blue) across all tasks. These results indicate that OQA effectively mitigates critic underestimation and improves online fine-tuning performance under both sparse and dense reward settings.

E.3 AntMaze without Offline Pretraining

Our main experiments study online fine-tuning from offline-pretrained actor–critic pairs. To examine whether the benefit of the optimistic backup depends on this initialization, we additionally evaluate OQATD3BC on AntMaze without offline pretraining. All actors and critics are randomly initialized, and training uses both the offline dataset and newly collected online transitions. Thus, this experiment removes the offline pretraining stage while retaining access to offline data.

We compare three variants: TD3+BC with a single actor–critic pair, OQATD3BC without the optimistic backup (w/o OB), and the full OQATD3BC method. The w/o OB variant retains two

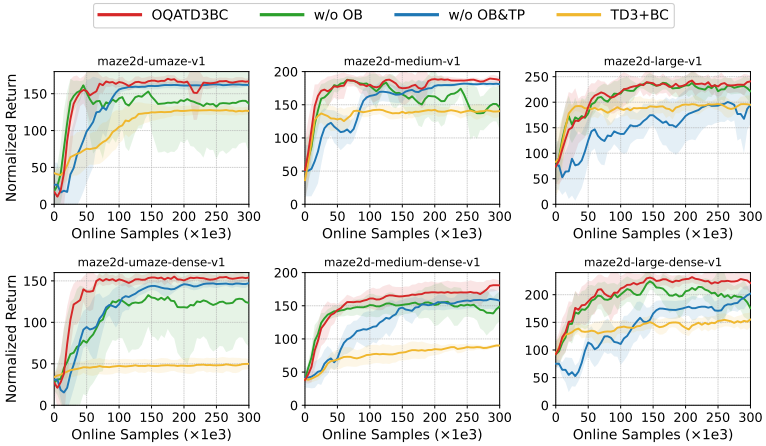


Figure 12: Paired comparisons on Maze2D tasks under sparse (top row) and dense (bottom row) rewards. Curves show normalized returns over 5 seeds.

Table 13: AntMaze results without offline pretraining. All methods start from randomly initialized actors and critics and train using both offline and online data. The w/o OB variant retains two actor–critic pairs but disables the optimistic backup. Entries report normalized returns; bold denotes the highest value in each row, including ties.

Dataset (v2)	TD3+BC	OQATD3BC w/o OB	OQATD3BC
antmaze-umaze	49.0	87.4	**93.8**
antmaze-umaze-diverse	24.6	14.0	**77.0**
antmaze-medium-play	0.0	3.0	**21.0**
antmaze-medium-diverse	0.0	2.2	**54.2**
antmaze-large-play	0.0	0.0	**0.4**
antmaze-large-diverse	**0.0**	**0.0**	**0.0**
Average	12.3	17.8	**41.1**

actor–critic pairs but sets $\alpha = 0$, disabling the optimistic correction. Unlike the corresponding ablation in the main experiments, neither branch is initialized from an offline-pretrained checkpoint.

As shown in Table 13, OQATD3BC achieves an average normalized return of 41.1, compared with 12.3 for TD3+BC and 17.8 for the w/o OB variant. On the two medium-maze tasks, OQATD3BC reaches 21.0 and 54.2, whereas TD3+BC scores zero on both and w/o OB achieves only 3.0 and 2.2. These results indicate that maintaining two branches alone does not account for the full improvement and support the benefit of the optimistic backup even without offline-pretrained value estimates.

The gains are nevertheless task-dependent. Performance remains near zero on both large-maze tasks, and all three methods score zero on antmaze-large-diverse. Thus, the optimistic backup does not by itself resolve the difficulty of learning on the hardest AntMaze tasks without offline pretraining. Overall, this experiment extends the evidence for OQA beyond correction of pessimism inherited from offline pretraining, but does not independently establish which sources of value-estimation error are responsible for the observed gains.

E.4 Sensitivity to UTD

The above comparison uses $\mathrm{UTD} = 5$, following the setting commonly adopted in prior O2O RL methods. To further examine whether OQA mainly benefits from this high-UTD setting, we sweep UTD from 1 to 10 and compare OQATD3BC with TD3+BC and TD3 under the same UTD values.

Figure 13 shows that OQATD3BC remains stable across different UTD values and does not exhibit a pronounced performance drop as UTD decreases. Moreover, OQATD3BC consistently outperforms both baselines across all tested UTD values. These results indicate that the gains of OQA are primarily driven by the optimistic Bellman backup rather than by simply increasing the number of critic updates.

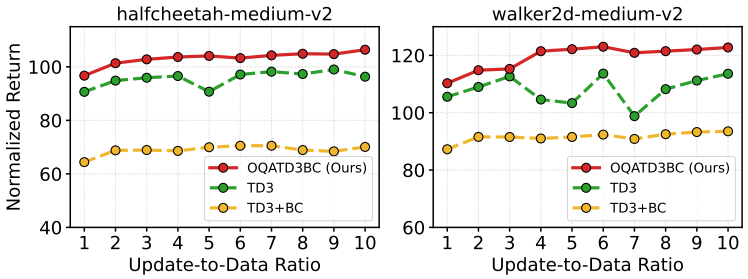


Figure 13: Sensitivity to the update-to-data ratio (UTD). Results are averaged over multiple seeds. OQATD3BC remains stable across different UTD values and consistently outperforms TD3+BC and TD3.

Table 14: Retention probability sensitivity of OQACQL across seven tasks. Entries are mean normalized returns; Average denotes the mean across tasks. HC denotes HalfCheetah, with r, me, m, and mr indicating random, medium-expert, medium, and medium-replay datasets, respectively. Adroit tasks use cloned datasets. Bold indicates the best result in each column.

p	HC-r	HC-me	HC-m	HC-mr	Hammer	Pen	Door	Average
0.3	89.8	125.6	90.0	103.4	140.0	120.5	103.5	110.4
0.5	89.3	**128.1**	96.4	111.5	**143.5**	139.1	92.0	114.3
0.7	79.0	125.7	97.1	**111.7**	138.3	139.4	104.1	113.6
0.9	**93.7**	126.3	**108.3**	102.2	140.7	**140.8**	**104.7**	**116.7**
1.0	69.9	125.5	100.3	105.5	135.6	131.1	99.7	109.7

E.5 Computational Cost

All experiments were run on GPU workers with one NVIDIA RTX 3090 GPU per run. Figure 14 compares wall-clock training time on walker2d-medium-replay-v2 with UTD$=1$. OQATD3BC matches the runtime of OPT, and requires substantially less training time than O2TD3.

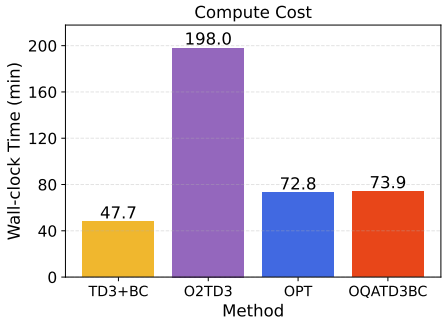


Figure 14: Compute cost on walker2d-medium-replay-v2. Bars report wall-clock time (minutes) for different methods under the same evaluation setting.

E.6 Sensitivity to the Retention Probability

The retention probability p controls the fraction of offline transitions retained for pretraining each actor–critic branch. We evaluate five retention probabilities on seven tasks, including four HalfCheetah locomotion tasks and three Adroit manipulation tasks. Table 14 summarizes the results.

The default setting $p = 0.9$ achieves the highest average return of 116.7, compared with 109.7 when both branches use the complete offline dataset ($p = 1.0$). However, the best retention probability remains task-dependent: HC-me and Hammer perform best at $p = 0.5$, while HC-mr performs best at $p = 0.7$. The remaining four tasks achieve their highest returns at $p = 0.9$. For all seven tasks, at

least one setting with $p < 1$ outperforms $p = 1.0$, although masking does not improve performance uniformly across all settings.

These results support $p = 0.9$ as a practical default across the evaluated tasks rather than a universally optimal choice. They are consistent with a trade-off between introducing branch diversity and preserving the coverage of each offline training subset. Together with the per-transition Min-Q analysis in the main text, the results suggest that data masking can complement the diversity already present between independently trained branches.

E.7 CQL without Conservative Regularization

To examine whether the gains of OQACQL can be attributed solely to removing CQL's conservative Q-regularization, we conduct an additional ablation on four Hopper datasets with different data qualities: random, medium, medium-replay, and medium-expert. The ablated variant retains a single actor–critic pair and the standard CQL twin-Q critic, but removes the conservative Q-regularizer during online fine-tuning without introducing the two-pair prior or the optimistic Bellman backup. Unlike CQL w/ BC Actor in Table 9, this variant retains the original CQL actor update, without the behavioral log-likelihood term used in OQACQL.

As shown in Table 15, removing conservative regularization alone does not yield consistent improvements. Although it substantially improves performance on the random dataset, it degrades performance on the other three datasets, resulting in only a marginal change in average performance. In contrast, OQACQL consistently achieves strong performance across all four datasets. These results indicate that simply removing conservative regularization is insufficient to address the pessimism inherited from offline pretraining.

This difference reflects the distinct roles of the two mechanisms. Removing the conservative penalty prevents additional conservative pressure during fine-tuning, but does not directly correct underestimation already present in the offline critic, which can continue to propagate through Bellman bootstrapping. Meanwhile, without the conservative penalty, out-of-distribution (OOD) actions receive less suppression, potentially distorting their relative value estimates compared with underestimated in-distribution actions. OQA instead actively corrects inherited underestimation through its optimistic Bellman backup while retaining conservative estimation within each branch. These results further demonstrate that the gains of OQACQL cannot be explained simply by removing CQL's conservative regularization.

Table 15: Effect of removing CQL's conservative Q-regularization on four Hopper datasets. OQACQL is included for comparison.

Task	CQL	CQL w/o Cons. Reg.	OQACQL
hopper-r	8.8	84.8	109.9
hopper-m	73.2	65.2	111.1
hopper-mr	103.0	83.7	110.4
hopper-me	111.9	66.0	113.7
Average	74.2	74.9	111.3

F Comparison with TD3 Fine-Tuning under High UTD

Prior works [7, 13, 19] have shown that directly removing offline constraints during fine-tuning can cause a sharp performance drop at the beginning of online training. This degradation is mainly caused by distribution shift between the offline data and the online state–action distribution. A high update-to-data ratio (UTD) can partly alleviate this issue by accelerating critic adaptation to the online distribution.

Under high UTD, TD3 fine-tuning, which removes the behavior-cloning term during online updates, becomes a strong baseline. We therefore compare OQATD3BC with TD3+BC and TD3 on MuJoCo locomotion under the same setting ($\mathrm{UTD} = 5$), using identical TD3+BC offline pretraining for all methods. During online fine-tuning, TD3+BC and OQATD3BC retain the behavior-cloning term, while TD3 removes it.

As shown in Figure 15, TD3 suffers from a clear early-stage performance drop, especially on the `medium` and `medium-expert` datasets. In contrast, OQATD3BC mitigates this degradation and recovers performance more efficiently, while achieving stronger final performance than both TD3+BC and TD3 on most tasks. This suggests that OQA provides a more reliable fine-tuning signal than simply removing offline regularization under high UTD.

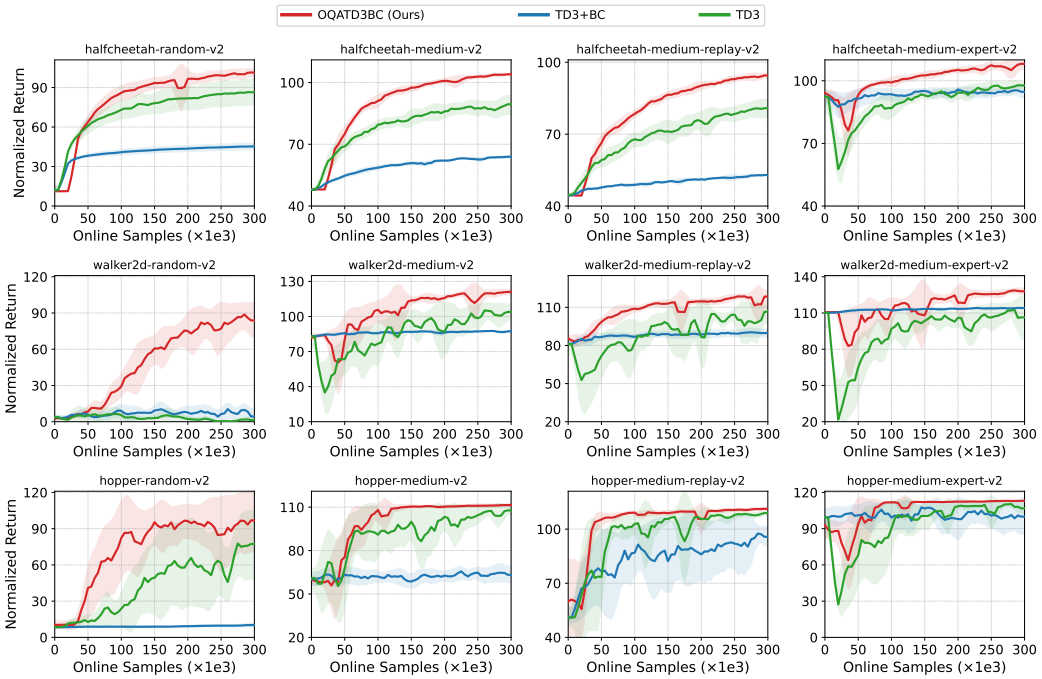


Figure 15: Learning curves on D4RL locomotion tasks during online fine-tuning with UTD $= 5$. Curves show the mean normalized return over 5 seeds, with shaded regions indicating standard deviation.

G Broader Impacts

This work studies offline-to-online reinforcement learning in simulated control benchmarks. Potential positive impacts include improving data-efficient policy adaptation in robotics and other sequential decision-making systems where online interaction is costly. Potential negative impacts may arise if similar methods are deployed in safety-critical control settings without adequate validation, since faster online adaptation can also amplify errors from misspecified rewards, unsafe exploration, or unreliable simulators. We therefore view OQA as an algorithmic contribution that should be combined with task-specific safety constraints and deployment-time monitoring before use in real-world systems.