Offline-Boosted Actor-Critic:
Adaptively Blending Optimal Historical Behaviors in Deep Off-Policy RL

Yu Luo [1] Tianying Ji [1] Fuchun Sun [1] Jianwei Zhang [2] Huazhe Xu [3 4 5] Xianyuan Zhan [5 6]

Abstract

Off-policy reinforcement learning (RL) has achieved notable success in tackling many complex real-world tasks, by leveraging previously collected data for policy learning. However, most existing off-policy RL algorithms fail to maximally exploit the information in the replay buffer, limiting sample efficiency and policy performance. In this work, we discover that concurrently training an offline RL policy based on the shared online replay buffer can sometimes outperform the original online learning policy, though the occurrence of such performance gains remains uncertain. This motivates a new possibility of harnessing the emergent outperforming offline optimal policy to improve online policy learning. Based on this insight, we present Offline-Boosted Actor-Critic (OBAC), a model-free online RL framework that elegantly identifies the outperforming offline policy through value comparison, and uses it as an adaptive constraint to guarantee stronger policy learning performance. Our experiments demonstrate that OBAC outperforms other popular model-free RL baselines and rivals advanced model-based RL methods in terms of sample efficiency and asymptotic performance across **53** tasks spanning **6** task suites[1].

1. Introduction

Online model-free deep reinforcement learning (RL) methods have achieved success in many challenging sequential
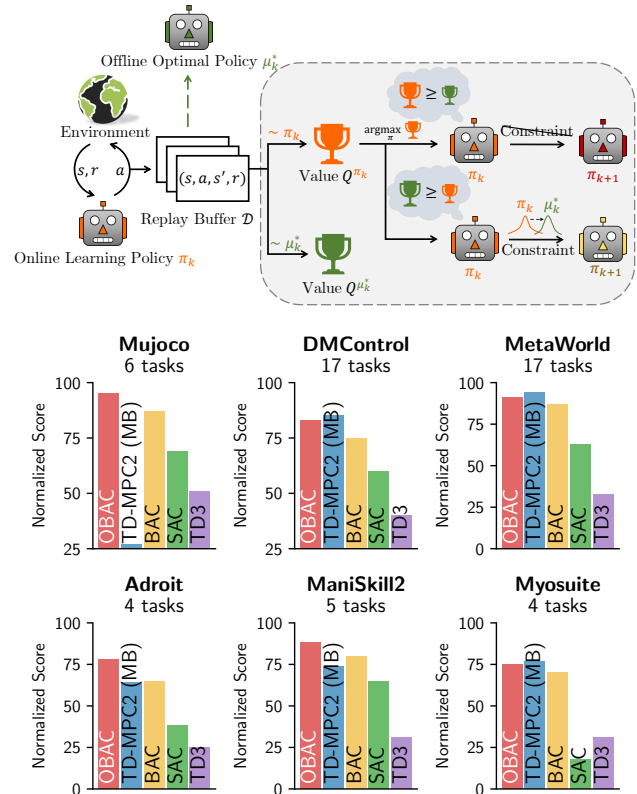

Figure 1. **Overview**. *(Top)*: we illustrate the framework of OBAC, where the concurrent offline optimal policy can boost the online learning policy with an adaptive constraint mechanism. *(Bottom)*: comparison of normalized score. Our OBAC can be comparable with advanced model-based RL method TD-MPCs, and outperform several popular model-free RL methods BAC, SAC and TD3.

decision-making tasks (Mnih et al., 2015; Van Hasselt et al., 2016; Wang et al., 2022), including gaming AI (Perolat et al., 2022), chip design (Mirhoseini et al., 2021), and automatic driving (Kiran et al., 2021). Many of these advances are attributed to off-policy RL methods (Kallus & Uehara, 2020), that enable agents to leverage collected data from historical policies to train the current policy. However, the reliance on millions of environment interaction steps still hampers the real-world deployment of RL (Haarnoja et al., 2018a). We boil the algorithmic inefficiency down to their

---

[1]Department of Computer Science and Technology, Tsinghua University [2]Department of Informatics, University of Hamburg [3]Institute for Interdisciplinary Information Sciences, Tsinghua University [4]Shanghai Qi Zhi Institute [5]Shanghai Artificial Intelligence Laboratory [6]Institute for AI Industry Research, Tsinghua University. Correspondence to: Fuchun Sun <fcsun@tsinghua.edu.cn>.

[1]Please refer to `https://roythuly.github.io/OBAC_web/` for experiment videos and benchmark results.

insufficient data utilization: when performing policy evaluation for value function learning and policy extraction via value maximization, most algorithms neglect and thus fail to leverage the inherent patterns and knowledge from the heterogeneous data in the replay buffer.

One cure for the inefficient data utilization of RL methods is model-based RL (Moerland et al., 2023): learning an environmental dynamics model as the reservoir of domain knowledge, by generating new pseudo-samples for Dyna-style (Ji et al., 2022) or planning-style (Hansen et al., 2024) policy learning. However, these approaches can be computationally complex and sometimes brittle due to the use of imperfect explicit model learning and long propagation chains. Alternatively, offline RL (Levine et al., 2020; Kumar et al., 2020; Kostrikov et al., 2021) provides a new possibility by allowing the learning of an optimal policy and the corresponding value function from fixed datasets without interacting with the environment. From an online RL perspective, leveraging such an offline learned policy and its offline value offers two advantages: *(i)* the offline learned policy, a blend of optimal historical behaviors, can serve as an explicit performance baseline for current online policy optimization; and *(ii)* the pessimistic training scheme (Kostrikov et al., 2021; Xu et al., 2022b) enables in-distribution value estimation and offline policy learning, preventing bias propagation issues seen in model-based RL.

Several prior studies have explored the use of offline RL to enhance online off-policy RL training, generally falling into two directions, each exploiting key advantages of offline RL: *(i)* incorporating an additional offline dataset for sampling augmentation (Song et al., 2022; Wagenmaker & Pacchiano, 2023; Ball et al., 2023), however, this may be expensive and easily impacted by data quality; and *(ii)* without the offline dataset, learning an optimal offline value function from collected data (*e.g.*, replay buffer) to adjust the online value function accordingly (*e.g.*, via linear interpolation) (Zhang et al., 2022a; Ji et al., 2023; Xu et al., 2024). However, this approach may result in inaccurate policy evaluation, particularly when the offline policy in the replay buffer is of low quality, thus naïvely mixing the offline value sometimes can be ineffective or even harmful. Although we do observe in this work that the optimal offline policy learned from the replay buffer can often outperform the online policy, the occurrence of such superiority varies across tasks as well as different training stages. Yet, despite numerous previous attempts, a unified understanding and framework for leveraging offline RL for effective online off-policy RL remains lacking. This raises the following questions: *When and how can we effectively leverage offline RL to ensure improvement in online off-policy RL?*

In this work, we introduce **O**ffline-**B**oosted **A**ctor-**C**ritic (**OBAC**) to address the above questions, providing a new

solution to leverage offline RL for adaptively blending optimal historical behaviors in online off-policy RL, as shown in Figure 1. To tackle the "when" issue, we compare the evaluated state-values of the online learning policy and the offline optimal policy to identify the superior offline optimal policy. To address the "how" challenge, we derive an adaptive mechanism that utilizes the superior offline optimal policy as a constraint to guide policy optimization. In short, OBAC can accumulate small performance gains when the offline optimal policy is better than the online learning policy, forming a positive cycle that a better offline policy helps online policy explore better data in return enhancing both policies in the next update, finally leading to significant overall performance improvement. Notably, to circumvent the computational complexity *w.r.t* explicitly learning the offline optimal policy—a similar issue seen in model-based RL when learning a dynamics model—we make a key technical contribution by introducing implicit offline policy learning in both evaluation and improvement steps, resulting in a cost-effective practical algorithm.

We evaluate our method across **53** diverse continuous control tasks spanning **6** domains: Mujoco (Todorov et al., 2012), DMControl (Tassa et al., 2018), Meta-World (Yu et al., 2020), Adroit (Kumar & Todorov, 2015), Myosuite (Caggiano et al., 2022), and Maniskill2 (Gu et al., 2022), comparing it with BAC (Ji et al., 2023), TD-MPC2 (Hansen et al., 2024), SAC (Haarnoja et al., 2018a), and TD3 (Fujimoto et al., 2018). Our results, summarized in Figure 1, showcase OBAC's superiority. It outperforms BAC, the first documented effective model-free algorithm on challenging high-dimensional dog locomotion series tasks, by adjusting Q values with offline values. When compared with TD-MPC2, a state-of-the-art model-based planning method known for efficiency in various tasks, OBAC demonstrates comparable performance with only **50%** of the parameters and **20%** less training time[2].

2. Preliminaries

We consider the conventional Markov Decision Process (MDP) (Bellman, 1957) defined by a 6-tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma, d_0 \rangle$, where $\mathcal{S} \in \mathbb{R}^n$ and $\mathcal{A} \in \mathbb{R}^m$ represent the continuous state and action spaces, $\mathcal{P}(s'|s, a) : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ denotes a Markovian transition (dynamics) distribution, $r(s, a) : \mathcal{S} \times \mathcal{A} \to \Delta(\mathbb{R})$ is a stochastic reward function, $\gamma \in [0, 1)$ gives the discounted factor for future rewards, and d_0 is the initial state distribution. The RL agent's objective is to find a policy $\pi(a|s) : \mathcal{S} \to \Delta(\mathcal{A})$ that maximizes the discounted cumulative reward from the environment, $J_\pi = \mathbb{E}_{d_0, \pi, \mathcal{P}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]$.

---

[2]We have released our code here: https://github.com/Roythuly/OBAC

We focus on the off-policy RL setting, where the agent interacts with the environment, collects new data into a replay buffer $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s, a, s', r)\}$, and updates the learning policy using the stored data. At the k-th iteration step, the online learning policy is denoted as π_k, with its corresponding Q value function

$$Q^{\pi_k}(s, a) = \mathbb{E}_{\pi_k, \mathcal{P}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a \right] \quad (1)$$

and the value function $V^{\pi_k}(s) = \mathbb{E}_{a \sim \pi_k} [Q^{\pi_k}(s, a)]$.

Considering the replay buffer as a given dataset allows us to derive a concurrent offline optimal policy $\mu_k^*(a|s)$, given by

$$\mu_k^* \triangleq \arg\max \mathbb{E}_{a \sim \mathcal{D}} [Q^{\mu_k}(s, a)]. \quad (2)$$

Unlike previous off-policy RL methods, we simultaneously train an online learning policy π_k and an offline optimal policy μ_k^* by sharing a communal replay buffer $\mathcal{D}$. A key property of μ_k^* is its strong relevance to dataset distribution (Fujimoto et al., 2019; Kumar et al., 2020), which means, action derived from it are restricted in support of actions in the replay buffer, $a \sim \mu_k^* \Rightarrow a \in \mathcal{D}$, while the online learning policy can have unrestricted action choices, $a \sim \pi_k \Rightarrow a \in \mathcal{A}$. Thus, μ_k^* characterizes the historical optimal behavior from the mixture of collected data, serving as an explicit and reliable performance baseline for π_k. Despite its conceptual simplicity, little previous work has introduced such a baseline for online learning optimization, resulting in wasted knowledge exploitation and sample inefficiency.

3. Offline-Boosted Off-Policy RL

In this section, we introduce Offline-Boosted Actor-Critic (OBAC), a framework aimed at improving the performance of online learning policies through the incorporation of an offline optimal policy. To gain insights into the behavior of such an offline optimal policy during online concurrent training, we conduct a set of thorough experiments, revealing its potential for outperforming the online learning policy. However, the timing of this superiority proves to be task-dependent and varies across different training stages. Taking this property into consideration, we detail how we adaptively integrate the offline optimal policy into alternating policy evaluation and optimization steps in off-policy RL paradigm. Following it up, we present a practical model-free RL algorithm based on the general actor-critic framework, achieving low computational cost and high sample efficiency.

3.1. A Motivating Example

In general, offline RL adheres to a principle of pessimism in policy training, aiming to prevent extrapolation errors in Q-value estimation by avoiding Out-Of-Distribution (OOD) actions (Fujimoto et al., 2019; Xie et al., 2021a; Shi et al.,
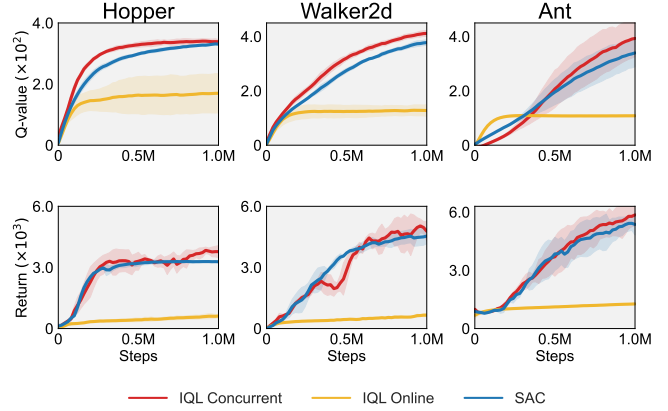

Figure 2. **Motivating example**. *(Top)*: Q-value estimation, *(Bottom)*: policy performance. We compare three agents including pure off-policy, concurrent offline and pure offline in online settings. The results demonstrate that the concurrent offline optimal policy can outperform the learning policy by sharing the replay buffer.

2022). However, these concerns typically stem from training policies on fixed datasets. In contrast, during the online RL training process, as new samples are continuously collected through interaction with the environment, both the *quantity* and *quality* of data in the replay buffer grow dynamically. These characteristics break the limitations inherent in offline RL, prompting us to explore better performance by integrating historical optimal behavior as more data becomes available. To explore these properties, we conduct investigations across three OpenAI Gym environments (Hopper-v3, Walker2d-v3, Ant-v3), employing three different agents:

Pure off-policy agent. We train a Soft Actor-Critic (*SAC*) agent (Haarnoja et al., 2018a) through 1 million steps of environmental interaction. At each time step, new data is accumulated in a dedicated replay buffer $\mathcal{D}_{SAC}$, updating the online learning policy.

Concurrent offline agent. Simultaneously with training the SAC agent, we employ the Implicit Q-Learning (IQL) (Kostrikov et al., 2021), an offline RL algorithm, to learn an offline optimal policy concurrently within the dynamically changing SAC buffer $\mathcal{D}_{SAC}$, referred to *IQL Concurrent*. Notably, the concurrent IQL agent does not interact with the environment during whole training process.

Online-training offline agent. In an online setting, we use IQL. This involves the IQL agent interacting with the environment, collecting new experiences stored in a replay buffer $\mathcal{D}_{IQL}$, and updating its policy, denoted as *IQL Online*. The training procedure aligns with that of the SAC agent but employs a different algorithm.

We present the performance of each agent, alongside their Q-value estimations in Figure 2. In each task, when com-

paring the *IQL concurrent* agent with the *SAC* agent, we observe the potential superiority of the offline optimal policy over the online one, even though both share the same replay buffer explored by SAC. Considering the different actor training methods between IQL (by forward KL-divergence) and SAC (by reverse KL-divergence) may cause performance loss (Chan et al., 2022), the Q-value comparison serves as a clearer indicator of the performance gap. These findings suggest that, contrary to its conservative reputation, offline RL can identify a potentially superior policy with a growing dataset when compared with off-policy RL. However, without the online policy buffer, even though the *IQL Online* agent can collect new samples, it exhibits notable conservatism, leading to premature convergence in both Q-value estimation and overall performance.

Despite these interesting discoveries, the timing of the offline optimal policy's superiority is uncertain, varying across tasks. And even within a single task, it depends on the quality of online learning policy interactions. While some works (Ji et al., 2023; Zhang et al., 2022a) have utilized the offline optimal Q-value to regularize the Q-value of the online learning policy, the challenge arises when the offline optimal policy is inferior, potentially leading to harm to the online policy. Thus, the uncertainty in the timing of superiority makes it non-trivial to leverage the offline optimal policy effectively.

3.2. Derivation of Offline-Boosted Policy Iteration

To better determine the suitable timing for introducing μ_k^*, we first individually evaluate both π_k and μ_k^*. Let $V^{\pi_k}(s)$ denote the state value function and $Q^{\pi_k}(s, a)$ represent the state-action value function of the online learning policy π_k. Similarly, let $V^{\mu_k^*}(s)$ and $Q^{\mu_k^*}(s, a)$ denote the value function and state-action value function for the offline optimal policy μ_k^*. We exploit the Bellman Expectation Operator given by:

$$\mathcal{T}^\chi Q^\chi(s, a) = r(s, a) + \gamma \mathbb{E}_{s', a' \sim \chi} \left[Q^\chi(s', a') \right], \quad (3)$$

$$V^\chi(s) = \mathbb{E}_{a \sim \chi} \left[Q^\chi(s, a) \right], \quad \chi = \pi_k \text{ or } \mu_k^*, \quad (4)$$

within the replay buffer for the evaluation of π_k and μ_k^*. Since $\mathcal{T}^\chi$ is a γ-contraction mapping within a single policy evaluation step *w.r.t* either π_k or μ_k^* (Denardo, 1967; Bellemare et al., 2017), the (state-action) value function $Q^\chi(s, a)$ and $V^\chi(s)$ can be obtained by repeatedly applying $\mathcal{T}^\chi$.

Next, when performing policy improvement, we utilize the maximization of $Q^{\pi_k}(s, a)$ as the objective function, as it provides an unbiased evaluation of the current learning policy. In contrast with previous off-policy learning methods, we introduce the offline optimal policy as a guidance policy to assist in generating a new online learning policy. Specifically, using the value function $V^\chi(s)$ as the performance indicator, which measures the performance of policies at

each state, we design the following adaptive mechanism for any state $s \in \mathcal{D}$:

- When $V^{\pi_k}(s) \geq V^{\mu_k^*}(s)$, *i.e.*, $\mathbb{E}_{a \sim \pi_k}[Q^{\pi_k}(s, a)] \geq \mathbb{E}_{a \sim \mu_k^*}[Q^{\mu_k^*}(s, a)]$, according to the definition of μ_k^* (2), it implies that even using the optimal actions in the replay buffer, the online learning policy π_k would still perform better than them. Thus, we can directly solve the objective function without the introduction of μ_k^*, avoiding potential negative effects.

- When $V^{\pi_k}(s) \leq V^{\mu_k^*}(s)$, we can identify better actions in the replay buffer compared to the current learning policy. In this case, we consider adding a policy constraint to the objective function. This ensures that the updated policy not only optimizes the objective function, but also integrates the distribution of better actions from the offline optimal policy as guidance, thus leveraging the historical optimal behavior when it surpasses the online learning policy.

Following this insight, we reconstruct the optimization problem in the policy improvement step as:

$$\pi_{k+1} = \arg \max_\pi \mathbb{E}_{a \sim \pi}[Q^{\pi_k}(s, a)] \quad (5)$$

$$\text{s.t.} \int_{a \in \mathcal{A}} f\left(\frac{\pi(a|s)}{\mu_k^*(a|s)} \right) \mathbb{1}\left(V^{\mu_k^*}(s) - V^{\pi_k}(s) \right) \mu_k^*(a|s) da \leq \epsilon, \quad (6)$$

$$\int_{a \in \mathcal{A}} \pi(a|s) da = 1, \ \forall s \in \mathcal{D}, \quad (7)$$

where $f(\cdot)$ is a regularization function, and $\mathbb{1}(\cdot)$ is an indicator function with $x \geq 0, \mathbb{1}(x) = 1; x < 0, \mathbb{1}(x) = 0$. Constraint (6) allows us to adaptively blend the offline optimal policy into online policy learning. By leveraging the Lagrangian multiplier and KKT condition (Peters et al., 2010; Peng et al., 2019), we derive the closed-form solution for the constrained optimization problem, as outlined in the following proposition.

Proposition 3.1. *For the constrained optimization problem defined by (5)∼(7), if $V^{\mu_k^*}(s) \geq V^{\pi_k}(s)$, the closed-form solution is*

$$\pi_{k+1} = \frac{1}{Z(s)} \mu_k^*(a|s) (f')^{-1} \left(Q^{\pi_k}(s, a) \right), \quad (8)$$

where $Z(s)$ is a partition function to normalise the action distribution. Or, when $V^{\mu_k^*}(s) < V^{\pi_k}(s)$, π_{k+1} is an ordinary solution to maximize $Q^{\pi_k}(s, a)$.

The proof is provided in Appendix A. Then, based on this closed-form solution, we show that the newly generated learning policy π_{k+1} would have a higher value than the old learning policy π_k *w.r.t.* the state-action distribution of the replay buffer in the following proposition.

Proposition 3.2. *Let π_k be the older learning policy and the newer one π_{k+1} be the solution of (5)∼(7). Then we achieve $Q^{\pi_{k+1}}(s,a) \geq Q^{\pi_k}(s,a)$ for all $(s,a) \in \mathcal{D}$, with the offline optimal policy μ_k^* serving as a performance baseline policy.*

With the convergence of policy evaluations on π_k and μ_k^*, as well as the results of policy improvement, we can alternate both steps and the online learning policy would provably converge to the optimal policy.

Proposition 3.3. *Assume $|\mathcal{A}| < \infty$, repeating the alternation of the policy evaluation (3)∼(4) and policy improvement (5)∼(7) can make any online learning policy $\pi_k \in \Pi$ converge to the optimal policies π^*, s.t. $Q^{\pi^*}(s_t,a_t) \geq Q^{\pi_k}(s_t,a_t), \forall(s_t,a_t) \in \mathcal{S} \times \mathcal{A}$.*

3.3. Offline-Boosted Actor Critic

Inspired by the previous analyses, we introduce our method for online off-policy RL, Offline-Boosted Actor-Critic (OBAC), summarized in Algorithm 1. To extend OBAC to large continuous control domains, we derive a practical implementation with high-quality function approximators following previous works (Haarnoja et al., 2018a; Lee et al., 2020). Besides, we highlight that OBAC introduces implicit regularization of the offline optimal policy for online policy training, without explicitly learning it, thereby mitigating computational complexity at each iteration.

Specifically, we consider parameterized state value functions $V_\psi^\chi(s)$ and state-action value functions $Q_\phi^\chi(s,a)$ with parameters ψ and ϕ, where χ represents both the online learning policy π and the offline optimal policy μ^*, alongside a tractable online learning policy $\pi_\theta(a|s)$ defined as a Gaussian policy with parameters θ. We utilize the updated online learning policy to interact with the environment, collecting new samples $\{(s,a,s',r)\}$ into the buffer $\mathcal{D}$.

Policy Evaluation. In this step, we first derive $V^\pi(s)$ and $Q_\phi^\pi(s,a)$ through the Bellman Expectation Operator (3), by minimizing the squared residual error

$$\arg\min_{Q_\phi^\pi} \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}} \left[\frac{1}{2}\left(Q_\phi^\pi(s,a)-\mathcal{T}^\pi Q_\phi^\pi(s,a)\right)^2\right] \quad (9)$$

and then we compute the value function $V^\pi(s)$ forward without gradient step,

$$V^\pi(s) = \mathbb{E}_{a\sim\pi}[Q_\phi^\pi(s,a)]. \quad (10)$$

Recalling the definition (2) of the offline optimal policy $\mu_k^* = \arg\max_{a\sim\mathcal{D}} Q^{\mu_k}(s,a)$. To eliminate the requirement of μ_k^* in the evaluation step, we transfer the Bellman Expectation Operator $\mathcal{T}^{\mu_k^*}$ as

$$\mathcal{T}^{\mu_k^*} Q_\phi^{\mu_k^*}(s,a) = r(s,a) + \gamma\mathbb{E}_{s',a'\sim\mu_k^*}\left[Q_\phi^{\mu_k^*}(s',a')\right]$$
$$= r(s,a) + \gamma\mathbb{E}_{s'}\left[\max_{a'\sim\mathcal{D}} Q_\phi^{\mu_k^*}(s',a')\right]. \quad (11)$$

Algorithm 1 Offline-Boosted Actor-Critic (OBAC)

1: **Input:** Critic Q_ϕ^π, critic $Q_\phi^{\mu^*}$, value $V_\psi^{\mu^*}$, actor π_θ, replay buffer $\mathcal{D}$.
2: **repeat**
3:     **for** each environment step **do**
4:         $a \sim \pi_\phi(a|s)$ and $r, s' \sim \mathcal{P}(s'|s,a)$
5:         $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s,a,s',r)\}$
6:     **end for**
7:     **for** each gradient step **do**
8:         *For π*: Update Q_ϕ^π by (9), compute V^π by (10)
9:         *For μ^**: Update $Q_\phi^{\mu^*}$ by (13), update $V_\psi^{\mu^*}$ by (12)
10:        Update π_θ by (15)
11:     **end for**
12: **until** the policy performs well in the environment

Within Equation (11), prior works on offline RL have effectively addressed $\max_{a'\sim\mathcal{D}}$ without explicitly requiring μ_k^*, such as expectile regression used in IQL (Kostrikov et al., 2021). For simplicity, we use expectile regression to achieve $Q^{\mu^*}(s,a)$ and $V^{\mu_k^*}(s)$, with two specific steps:

$$\arg\min_{V_\psi^{\mu^*}} \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[L_2^\tau\left(Q_\phi^{\mu^*}(s,a) - V_\psi^{\mu^*}(s)\right)\right] \quad (12)$$

where $L_2^\tau(x) = |\tau - \mathbb{1}(x<0)|x^2$ is the expectile regression function, and τ is an expectile factor. And,

$$\arg\min_{Q_\phi^{\mu^*}} \mathbb{E}_{(s,a,s',r)\sim\mathcal{D}}\left[\frac{1}{2}\left(r + \gamma V_\psi^{\mu^*}(s') - Q_\phi^{\mu^*}(s,a)\right)^2\right]. \quad (13)$$

Based on Theorem 3 in Kostrikov et al. (2021), when $\tau \to 1$, the term $\max_{a'\sim\mathcal{D}} Q_\phi^{\mu_k^*}(s',a')$ can be approached to derive $Q_\phi^{\mu_k^*}(s,a)$. Thus, we complete the policy evaluation for both π_k and μ_k^*, where the former is based on (9) and (10), and the latter by (12) and (13). In our implementation, we employ the Clipped Double Q-technique (Fujimoto et al., 2018) for stability and mitigating overestimation.

Policy Improvement. Directly using the closed-form solution (8) for policy updates is intractable with the unknown $Z(s)$ and μ_k^*. In our implementation, we opt to restrict the solution within a tractable set of Gaussian policies and project the improved policy into these desired policies by Kullback-Leibler divergence $D_{\mathrm{KL}}(\cdot)$. Then, if we choose the regularization function $f(x) = x\log x$, the objective of the updated policy is

$$\arg\min_\pi \begin{cases} D_{\mathrm{KL}}\left(\pi \middle\| \dfrac{\exp(Q^{\pi_k})}{Z}\right), V^{\pi_k} \geq V^{\mu_k^*}, \\ D_{\mathrm{KL}}\left(\pi \middle\| \dfrac{\mu_k^* \exp(Q^{\pi_k})}{Z}\right), V^{\pi_k} \leq V^{\mu_k^*}. \end{cases} \quad (14)$$
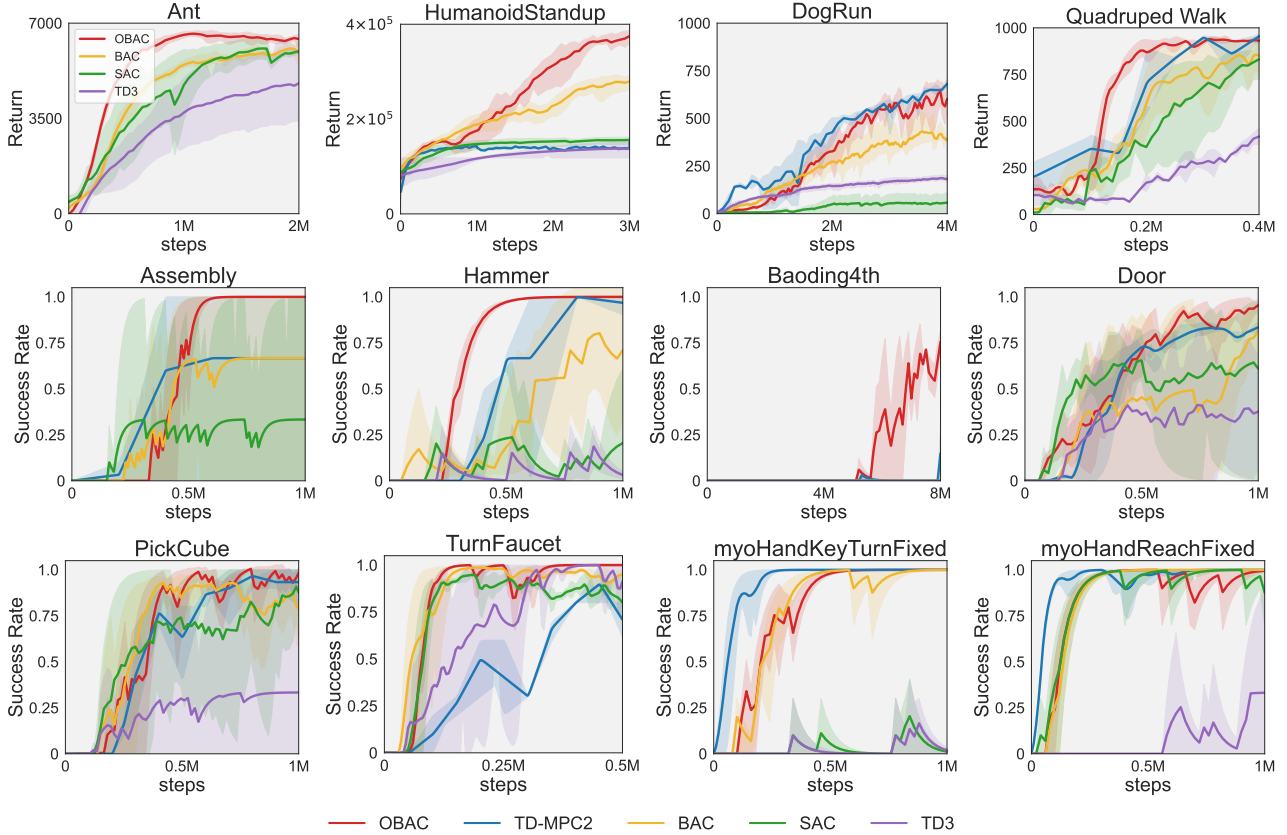
5

Figure 3. **Main results.** We provide performance comparisons for 12 of the 53 tasks, two for each task suite. Please refer to Appendix C.2 for the comprehensive results. The solid lines are the average return/success rate, while the shades indicate 95% confidence intervals. All algorithms are evaluated with 5 random seeds.

Given the data $\{(s, a, s', r)\}$ randomly sampled from $\mathcal{D}$, if $V^{\mu_k^*}(s) < V^{\pi_k}(s)$, we disable the constraint and update the policy as prior off-policy RL methods. In contrast, if $V^{\mu_k^*}(s) \geq V^{\pi_k}(s)$, we consider the distribution of sampled data can achieve better performance than the learning policy, thus it can approximate the offline optimal policy (Zhang et al., 2022a). Thus, with the Gaussian policy set, we summarise both cases for policy updating:

$$\arg \min_{\pi_\theta \in \Pi} \mathbb{E}_{s \sim \mathcal{D}} \left\{ \mathbb{E}_{a \sim \pi_\theta} [\log \pi_\theta(a|s) - Q^{\pi_k}(s, a)] \right.$$
$$\left. - \lambda \mathbb{E}_{a \sim \mathcal{D}} [\log \pi_\theta(a|s)] \mathbb{1} \left(V^{\mu_k^*}(s) - V^{\pi_k}(s) \right) \right\}. \quad (15)$$

where λ is a behavior clone weighted factor, similar to previous offline RL works (Kostrikov et al., 2021).

4. Experiment

We evaluate OBAC across **53** continuous control tasks spanning **6** domains: Mujoco (Todorov et al., 2012), DMControl (Tassa et al., 2018), Meta-World (Yu et al., 2020), Adroit (Kumar & Todorov, 2015), Myosuite (Caggiano et al., 2022), and Maniskill2 (Gu et al., 2022). These tasks cover

a wide range of challenges, including high-dimensional states and actions (up to $\mathcal{S} \in \mathbb{R}^{375}$ and $\mathcal{A} \in \mathbb{R}^{39}$), sparse rewards, multi-object and delicate manipulation, musculoskeletal control, and complex locomotion. Please refer to Appendix B for the implementation details and environment settings in our experiments.

With these experimental evaluations, we seek to investigate the following questions: 1) Does the introduction of the concurrent offline optimal policy significantly improve performance? 2) How does OBAC compare to the popular model-free and model-based RL methods for sample efficiency and eventual performance? 3) How does the adaptive mechanism work in OBAC?

Baselines. Our baselines contain: 1) **SAC** (Haarnoja et al., 2018a) and **TD3** (Fujimoto et al., 2018), two data-efficient off-policy model-free RL methods, where the former utilizes stochastic policy while the latter uses deterministic policy; 2) **BAC** (Ji et al., 2023), an off-policy model-free RL method to employ the state-action value of offline optimal policy to mitigate the underestimation of the Q-value of online learning policy; 3) **TD-MPC2** (Hansen et al., 2024), a
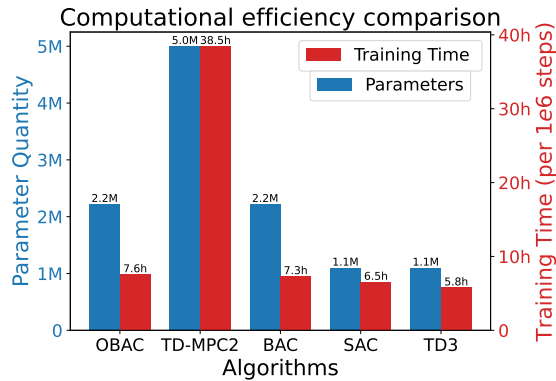
Figure 4. **Parameters and Wall time.** *(left)*: parameter quantity of each algorithm, and *(right)*: training wall time per 1 million steps of environmental interaction. With these comparisons, OBAC can achieve similar simplicity with other model-free RL methods, and require only **50**% parameters and **20**% training time compared to model-based RL methods.

Figure 5. **Ablation on adaptive constraints.** *(Top)*: Policy performance, and *(Bottom)*: Q value comparison. We adopt two tasks from Mujoco suite and DMControl suite to demonstrate the necessity of adaptive constraints in OBAC. Mean of 5 runs; shaded areas are 95% confidence intervals.

high-efficient model-based RL method that combines model predictive control and TD-learning.

4.1. Experimental Results

Performance comparison. The learning curves presented in Figure 3 demonstrate the performance of OBAC alongside various baselines across diverse task suites. Overall, OBAC, as a model-free RL method, obviously outperforms other model-free RL baselines and exhibits comparable capabilities to model-based RL methods in terms of exploration efficiency and asymptotic performance. Notably, with identical hyperparameters, OBAC excels in high-dimensional locomotion tasks (DogRun), multi-body contact manipulation (Baoding4th), and intricate muscle control tasks (myoHand). A noteworthy achievement is the successful application of OBAC to solve the Baoding-4th task, a challenging scenario involving a shadow hand managing the rotation cycle of two Baoding balls, without any prior demonstrations. Additionally, we observe that TD-MPC2, even with different prediction horizons (ranging from 3 to 6), does not perform well on the Mujoco suite, possibly influenced by variations of the *done* signal settings of the environments (please refer to Appendix C.2 for more discussion). We provide a comprehensive presentation of experimental results in Appendix C.2, with accompanying visualizations provided in `https://roythuly.github.io/OBAC/`.

Computational efficiency. To showcase OBAC's efficiency, we provide a detailed comparison with our baselines in Figure 4, considering parameter quantity and training wall time on a single RTX3090 GPU. In contrast to model-based RL methods TD-MPC2, OBAC achieves a **50**% reduction in parameter quantity, featuring a simpler architecture with
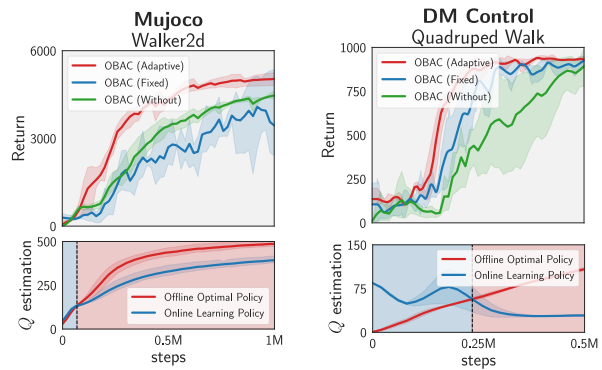
just two critics, one value and one actor, each component employing a 3-layer MLP with a hidden-layer size of 512. In comparison, TD-MPC2 requires additional components such as a dynamics model, reward model, and 5 ensemble critics. Furthermore, OBAC exhibits notable improvements in training efficiency, requiring only **20**% of the training time per 1 million steps of environmental interaction while maintaining comparable sample efficiency and convergent performance. When compared with other model-free RL methods, OBAC demonstrates a similar level of simplicity and cost-effectiveness, highlighting the effectiveness of our algorithmic implementation.

4.2. Ablation Studies

We conduct several investigations to ablate the effectiveness of OBAC's design in this section.

Necessity of adaptive constraints. One of the key designs of OBAC is the adaptive policy constraint, which dynamically adjusts based on value comparisons. To evaluate its necessity, we conducted experiments comparing *OBAC (Adaptive)* with a fixed constraint *OBAC (Fixed)*, where policy updates are consistently constrained by the empirical distribution of the replay buffer, as well as *OBAC (Without)* constraint. The top of Figure 5 illustrates that the fixed constraint leads to performance degradation due to excessive conservatism, underscoring the effectiveness of our adaptive mechanism for performance improvement. Additionally, the comparison of Q values between the offline optimal policy and the learning policy is visualized at the bottom of Figure 5. The blue region indicates $Q^{\mu_k^*} \leq Q^{\pi_k}$ while the red region signifies $Q^{\mu_k^*} \geq Q^{\pi_k}$. The combined results indicate that when $Q^{\mu_k^*} \geq Q^{\pi_k}$ occurs, activating the policy constraint significantly improves OBAC's performance.

Extension in noise and sparse tasks. To better ground

7

(a) **Noise tasks.**
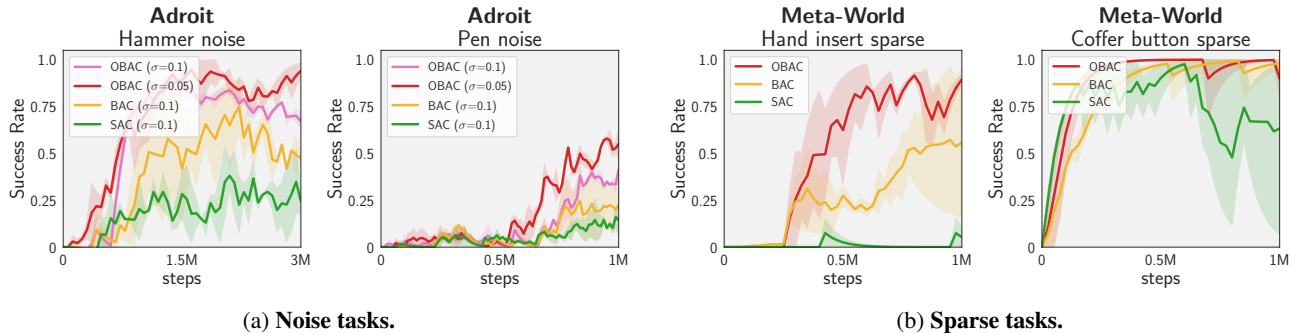
(b) **Sparse tasks.**

Figure 6. **Extension on noise and sparse tasks.** We evaluate OBAC in noisy and sparse tasks for comprehensive investigations. Mean of 5 runs, shaded areas are 95% confidence intervals.

the effectiveness of OBAC, we applied it to a set of stochastic tasks, where a Gaussian noise with different standard deviations σ to each dimension of actions at every step, including $\sigma = 0.05$ and $\sigma = 0.1$ to represent the increasing environmental stochasticity. Compared to our baselines BAC and SAC, as shown in Figure 6a, OBAC consistently demonstrates superior robustness, even in challenging scenarios with higher noise levels (*e.g.*, $\sigma = 0.1$). Furthermore, we extended the evaluation to tasks featuring sparse rewards, where success yields a reward of 1 and failure results in a reward of 0. The results in Figure 6b highlight OBAC's capacity to outperform the baselines in sparse reward scenarios, underscoring its effectiveness across diverse challenges. Appendix C.3 provides a detailed comparison of performance decline rates to better explain the robustness of OBAC.

More Ablations. In addition to these results, we provide the variant of OBAC, where a deterministic policy is employed for online learning, as shown in Figure 19, Appendix D. The results show that when employing a deterministic online learning policy, the adaptive constraint can also provide performance improvement across various scenarios, indicating the versatility of our methods.

5. Related Works

We briefly summarize relevant methods about off-policy RL, offline RL and off-policy RL with offline techniques.

Off-policy RL. Off-policy RL offers a general paradigm for reusing previously collected data in current policy training (Munos et al., 2016; Peng et al., 2019; Rakelly et al., 2019; Duan et al., 2020). This involves alternating between policy evaluation, where the performance is assessed by computing a value function, and policy improvement, where the policy is updated based on the value function (Chan et al., 2022). Many approaches focus on accurate value function estimation (Fujimoto et al., 2018; Moskovitz et al., 2021; Wei et al., 2022), exploration term design (Haarnoja

et al., 2018a; Liu et al., 2020), and real-world applications (Delarue et al., 2020; Chen et al., 2022). However, in the training process, the *i.i.d* assumption of data (Judah et al., 2014; Chen & Jiang, 2019; Zhang et al., 2021; Liu et al., 2022) makes them often overlook the inherent domain knowledge (DEramo et al., 2020), which is available as an offline optimal policy by re-stitching different state-action pairs (Xu et al., 2022a) in the replay buffer. This oversight leads to issues of sample inefficiency and training instability, which OBAC addresses by leveraging this knowledge to enhance policy performance.

Offline RL. Given a fixed dataset without environmental interaction, offline RL aims to train a policy within the support of the training distribution (Levine et al., 2020; Prudencio et al., 2023), categorized by either explicit or implicit constraints. Explicit constraints involve learning an empirical distribution of behavior policy (Fujimoto et al., 2019; Kumar et al., 2019) for policy regularization or improving in-distribution action values while decreasing out-of-distribution (OOD) actions' values (Kumar et al., 2020). Implicit constraints (Kostrikov et al., 2021; Xiao et al., 2022; Mao et al., 2024) achieve similar value/policy regularization without additional behavior approximation through implicit constraints. However, the pessimistic principle in these methods makes their application challenging in online settings (Xie et al., 2021b), as the policy tends to be too conservative to explore better actions. While, our work adopts offline RL techniques within an online training setting, where the data is growing as the policy training and exploring, avoiding the conservative nature of offline RL.

Bridging off-policy and offline RL. Recent works have explored the settings of offline-to-online RL (Lee et al., 2022; Yu & Zhang, 2023; Zhang et al., 2022b) or Hybrid RL (Niu et al., 2022; Panaganti et al., 2022), leveraging additional offline datasets for policy pretraining (Zhang et al., 2022b; Yu & Zhang, 2023) or for training data augmentation (Wagenmaker & Pacchiano, 2023; Ball et al., 2023; Song et al., 2022; Uchendu et al., 2023) to fine-tune the

learning policy and improve sample efficiency. In contrast, OBAC's training can start with zero data, following a pure online setting without pretraining. Some similar methods introduce the offline optimal Q-value, estimated by the transitions observed in the replay memory, to regularize the Q-value of the online learning policy (Zhang et al., 2022a; Ji et al., 2023). However, our motivating examples demonstrate that offline optimal Q-value may be suboptimal to the learning one, especially in the early training stage, leading to a potential drawback for the learning policy. OBAC addresses this issue by introducing an adaptive constraint that smartly determines the timing of introducing the offline optimal policy.

6. Conclusion

In this work, we propose a novel off-policy RL framework OBAC, where an agent can exploit an offline optimal policy concurrently trained by the replay buffer to boost the performance of the online learning policy. Based on the theoretical results of offline-boosted policy iteration, which shows the convergence to the optimal policy, this naturally derives a practical algorithm with low computational cost and high sample efficiency. Abundant experimental results demonstrate the superiority of OBAC when compared with both model-free and model-based RL baselines. Our findings offer valuable insights that leveraging collected data to derive an offline optimal policy can effectively improve the sample efficiency, introducing a novel and practical approach to combining off-policy RL and offline RL. Future works of OBAC can be extended by applying more advanced offline RL methods when deriving the offline optimal policy, or adding more exploration in the online policy to facilitate exploring better data, thus allowing OBAC to improve policy performance from both exploitation and exploration.

Acknowledgements

Impact Statement

This work contributes to advancing the field of Reinforcement Learning (RL), particularly in the domain of off-policy RL algorithms. The proposed algorithm holds potential implications for real-world applications, especially in areas such as robotics. As a general off-policy RL algorithm, the main uncertainty of the proposed method might be the fact that the RL training process itself is somewhat brittle. Besides, the exploration of an RL agent in real-world environments may require several safety considerations, to avoid unsafe behavior during the exploration process.

References

Ball, P. J., Smith, L., Kostrikov, I., and Levine, S. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, 2023.

Bellemare, M. G., Dabney, W., and Munos, R. A distributional perspective on reinforcement learning. In *International conference on machine learning*, 2017.

Bellman, R. A markovian decision process. *Journal of mathematics and mechanics*, pp. 679–684, 1957.

Caggiano, V., Wang, H., Durandau, G., Sartori, M., and Kumar, V. Myosuite–a contact-rich simulation suite for musculoskeletal motor control. *Proceedings of Machine Learning Research*, 168, 2022.

Chan, A., Silva, H., Lim, S., Kozuno, T., Mahmood, A. R., and White, M. Greedification operators for policy optimization: Investigating forward and reverse kl divergences. *The Journal of Machine Learning Research*, 23 (1):11474–11552, 2022.

Chen, J. and Jiang, N. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, 2019.

Chen, X., Qu, G., Tang, Y., Low, S., and Li, N. Reinforcement learning for selective key applications in power systems: Recent advances and future challenges. *IEEE Transactions on Smart Grid*, 13(4):2935–2958, 2022.

Delarue, A., Anderson, R., and Tjandraatmadja, C. Reinforcement learning with combinatorial actions: An application to vehicle routing. In *Advances in Neural Information Processing Systems*, 2020.

Denardo, E. V. Contraction mappings in the theory underlying dynamic programming. *Siam Review*, 9(2):165–177, 1967.

DEramo, C., Tateo, D., Bonarini, A., Restelli, M., and Peters, J. Sharing knowledge in multi-task deep reinforcement learning. In *International Conference on Learning Representations*, 2020.

Duan, Y., Jia, Z., and Wang, M. Minimax-optimal off-policy evaluation with linear function approximation. In *International Conference on Machine Learning*, 2020.

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. In *Advances in neural information processing systems*, 2021.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, 2018.

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

Gu, J., Xiang, F., Li, X., Ling, Z., Liu, X., Mu, T., Tang, Y., Tao, S., Wei, X., Yao, Y., et al. Maniskill2: A unified benchmark for generalizable manipulation skills. In *The Eleventh International Conference on Learning Representations*, 2022.

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

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018b.

Hansen, N., Su, H., and Wang, X. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

Ji, T., Luo, Y., Sun, F., Jing, M., He, F., and Huang, W. When to update your model: Constrained model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022.

Ji, T., Luo, Y., Sun, F., Zhan, X., Zhang, J., and Xu, H. Seizing serendipity: Exploiting the value of past success in off-policy actor-critic. *arXiv preprint arXiv:2306.02865*, 2023.

Judah, K., Fern, A. P., Dietterich, T. G., and Tadepalli, P. Active lmitation learning: formal and practical reductions to iid learning. *J. Mach. Learn. Res.*, 15(1):3925–3963, 2014.

Kallus, N. and Uehara, M. Double reinforcement learning for efficient off-policy evaluation in markov decision processes. *The Journal of Machine Learning Research*, 21 (1):6742–6804, 2020.

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

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2021.

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

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

Kumar, V. and Todorov, E. Mujoco haptix: A virtual reality system for hand manipulation. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 657–663. IEEE, 2015.

Lee, A. X., Nagabandi, A., Abbeel, P., and Levine, S. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. In *Advances in Neural Information Processing Systems*, 2020.

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

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Liu, F., Viano, L., and Cevher, V. Understanding deep neural function approximation in reinforcement learning via ϵ-greedy exploration. In *Advances in Neural Information Processing Systems*, 2022.

Liu, Y., Swaminathan, A., Agarwal, A., and Brunskill, E. Provably good batch off-policy reinforcement learning without great exploration. In *Advances in neural information processing systems*, 2020.

Mao, L., Xu, H., Zhang, W., and Zhan, X. Odice: Revealing the mystery of distribution correction estimation via orthogonal-gradient update. *arXiv preprint arXiv:2402.00348*, 2024.

Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J. W., Songhori, E., Wang, S., Lee, Y.-J., Johnson, E., Pathak, O., Nazi, A., et al. A graph placement methodology for fast chip design. *Nature*, 594(7862):207–212, 2021.

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

Moerland, T. M., Broekens, J., Plaat, A., Jonker, C. M., et al. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1): 1–118, 2023.

Moskovitz, T., Parker-Holder, J., Pacchiano, A., Arbel, M., and Jordan, M. Tactical optimism and pessimism for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 34:12849–12863, 2021.

Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. Safe and efficient off-policy reinforcement learning. In *Advances in neural information processing systems*, 2016.

Niu, H., Qiu, Y., Li, M., Zhou, G., HU, J., Zhan, X., et al. When to trust your simulator: Dynamics-aware hybrid offline-and-online reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022.

Panaganti, K., Xu, Z., Kalathil, D., and Ghavamzadeh, M. Robust reinforcement learning using offline data. In *Advances in neural information processing systems*, 2022.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Perolat, J., De Vylder, B., Hennes, D., Tarassov, E., Strub, F., de Boer, V., Muller, P., Connor, J. T., Burch, N., Anthony, T., et al. Mastering the game of stratego with model-free multiagent reinforcement learning. *Science*, 378(6623): 990–996, 2022.

Peters, J., Mulling, K., and Altun, Y. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2010.

Prudencio, R. F., Maximo, M. R., and Colombini, E. L. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Rakelly, K., Zhou, A., Finn, C., Levine, S., and Quillen, D. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, 2019.

Shi, L., Li, G., Wei, Y., Chen, Y., and Chi, Y. Pessimistic q-learning for offline reinforcement learning: Towards optimal sample complexity. In *International Conference on Machine Learning*, 2022.

Song, Y., Zhou, Y., Sekhari, A., Bagnell, D., Krishnamurthy, A., and Sun, W. Hybrid rl: Using both offline and online data can make rl efficient. In *International Conference on Learning Representations*, 2022.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D. d. L., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

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

Van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, 2016.

Wagenmaker, A. and Pacchiano, A. Leveraging offline data in online reinforcement learning. In *International Conference on Machine Learning*, 2023.

Wang, X., Wang, S., Liang, X., Zhao, D., Huang, J., Xu, X., Dai, B., and Miao, Q. Deep reinforcement learning: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Wei, W., Zhang, Y., Liang, J., Li, L., and Li, Y. Controlling underestimation bias in reinforcement learning via quasi-median operation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

Xiao, C., Wang, H., Pan, Y., White, A., and White, M. The in-sample softmax for offline reinforcement learning. In *International Conference on Learning Representations*, 2022.

Xie, T., Cheng, C.-A., Jiang, N., Mineiro, P., and Agarwal, A. Bellman-consistent pessimism for offline reinforcement learning. In *Advances in neural information processing systems*, 2021a.

Xie, T., Jiang, N., Wang, H., Xiong, C., and Bai, Y. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. In *Advances in neural information processing systems*, 2021b.

Xu, G., Zheng, R., Liang, Y., Wang, X., Yuan, Z., Ji, T., Luo, Y., Liu, X., Yuan, J., Hua, P., et al. Drm: Mastering visual reinforcement learning through dormant ratio minimization. In *International Conference on Learning Representations*, 2024.

Xu, H., Jiang, L., Jianxiong, L., and Zhan, X. A policy-guided imitation approach for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:4085–4098, 2022a.

Xu, H., Jiang, L., Li, J., Yang, Z., Wang, Z., Chan, V. W. K., and Zhan, X. Offline rl with no ood actions: In-sample learning via implicit value regularization. In *International Conference on Learning Representations*, 2022b.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, 2020.

Yu, Z. and Zhang, X. Actor-critic alignment for offline-to-online reinforcement learning. In *International Conference on Machine Learning*, 2023.

Zhang, H., Xiao, C., Wang, H., Jin, J., Müller, M., et al. Replay memory as an empirical mdp: Combining conservative estimation with experience replay. In *International Conference on Learning Representations*, 2022a.

Zhang, H., Xu, W., and Yu, H. Policy expansion for bridging offline-to-online reinforcement learning. In *International Conference on Learning Representations*, 2022b.

Zhang, J., Kim, J., O'Donoghue, B., and Boyd, S. Sample efficient reinforcement learning with reinforce. In *Proceedings of the AAAI conference on artificial intelligence*, 2021.

A. Theoretical Analyses

Lemma A.1. *Given a fixed policy χ, the Bellman Expectation Operator $\mathcal{T}^\chi$ is a γ-contraction mapping within a single evaluation step.*

Proof. Let $Q_1(s, a)$ and $Q_2(s, a)$ be two arbitrary state-action value functions. Based on the definition of $\mathcal{T}^\chi$, we have

$$
\begin{aligned}
\|\mathcal{T}^\chi Q_1(s, a) - \mathcal{T}^\chi Q_2(s, a)\|_\infty &= \|r(s, a) - \gamma \mathbb{E}_{s', a' \sim \chi}[Q_1(s', a')] - r(s, a) + \gamma \mathbb{E}_{s', a' \sim \chi}[Q_2(s', a')]\|_\infty \\
&\leq \gamma \mathbb{E}_{s'} |\mathbb{E}_{a' \sim \chi}[Q_1(s', a')] - \mathbb{E}_{a' \sim \chi}[Q_2(s', a')]| \\
&\leq \gamma \mathbb{E}_{s', a' \sim \chi} |Q_1(s', a') - Q_2(s', a')| \\
&\leq \gamma \max_{s, a} |Q_1(s, a) - Q_2(s, a)| \\
&= \gamma \|Q_1(s, a) - Q_2(s, a)\|_\infty.
\end{aligned}
\tag{16}
$$

Thus, we conclude that $\mathcal{T}^\chi$ is a γ-contraction mapping. Further, this property guarantees that given any fixed policy χ, any initial Q function will converge to a unique fixed point by repeatedly applying this operator. $\square$

Recall the constrained optimization problem in policy improvement step,

$$
\pi_{k+1} = \arg\max_\pi \mathbb{E}_{a \sim \pi}[Q^{\pi_k}(s, a)]
$$

$$
\text{s.t.} \int_{a \in \mathcal{A}} f\left(\frac{\pi(a|s)}{\mu_k^*(a|s)}\right) \mathbb{1}\left(V^{\mu_k^*}(s) - V^{\pi_k}(s)\right) \mu_k^*(a|s) da \leq \epsilon,
$$

$$
\int_{a \in \mathcal{A}} \pi(a|s) da = 1, \ \forall s \in \mathcal{D},
$$

Proposition A.2. *For the constrained optimization problem defined by (5)∼(7), if $V^{\mu_k^*}(s) \geq V^{\pi_k}(s)$, the closed-form solution is*

$$
\pi_{k+1} = \frac{1}{Z(s)} \mu_k^*(a|s) \left(f'\right)^{-1}\left(Q^{\pi_k}(s, a)\right),
\tag{17}
$$

where $Z(s)$ is a partition function to normalise the action distribution. Or, when $V^{\mu_k^*}(s) < V^{\pi_k}(s)$, π_{k+1} is an ordinary solution to maximize $Q^{\pi_k}(s, a)$.

Proof. Following prior methods (Peters et al., 2010; Peng et al., 2019), we apply the KKT conditions for the constrained optimization problem. The Lagrangian is:

$$
\mathcal{L}(\pi, \lambda, \alpha) = \mathbb{E}_{a \sim \pi}[Q^{\pi_k}(s, a)] + \lambda\left[\epsilon - f\left(\frac{\pi(a|s)}{\mu^*(a|s)}\right) \mathbb{1}\left(V^{\mu_k^*}(s) - V^{\pi_k}(s)\right) \mu_k^*(a|s)\right] + \alpha\left(1 - \int_{a \in \mathcal{A}} \pi(a|s) da\right). \tag{18}
$$

Then, we perform differentiation with respect to π, and have

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \pi} &= Q^{\pi_k}(s, a) - \lambda \cancel{\mu_k^*(a|s)} \frac{\mathbb{1}\left(V^{\mu_k^*}(s) - V^{\pi_k}(s)\right)}{\cancel{\mu_k^*(a|s)}} f'\left(\frac{\pi(a|s)}{\mu^*(a|s)}\right) - \alpha \\
&= Q^{\pi_k}(s, a) - \lambda \mathbb{1}\left(V^{\mu_k^*}(s) - V^{\pi_k}(s)\right) f'\left(\frac{\pi(a|s)}{\mu^*(a|s)}\right) - \alpha
\end{aligned}
\tag{19}
$$

By KKT conditions, we set $\frac{\partial \mathcal{L}}{\partial \pi} = 0$. When $V^{\mu_k^*}(s) \geq V^{\pi_k}(s)$, *i.e.*, $\mathbb{1}\left(V^{\mu_k^*}(s) - V^{\pi_k}(s)\right) = 1$, then we have

$$
\pi_{k+1} = \frac{1}{Z(s)} \mu^*(a|s) \left(f'\right)^{-1}\left(Q^{\pi_k}(s, a)\right),
\tag{20}
$$

where $Z(s)$ is a partition function to normalise the action distribution, and λ is a behavior clone weight.

In contrast, if $V^{\mu_k^*}(s) < V^{\pi_k}(s)$, the constraint (6) is ineffective. Thus, π_{k+1} is an ordinary solution to maximize $Q^{\pi_k}(s, a)$. The proof is completed. $\square$

Proposition A.3. *Let π_k be the older learning policy and the newer one π_{k+1} be the solution of (5)$\sim$(7). Then we achieve $Q^{\pi_{k+1}}(s,a) \geq Q^{\pi_k}(s,a)$ for all $(s,a) \in \mathcal{D}$, with the offline optimal policy μ_k^* serving as a performance baseline policy.*

Proof. Since π_{k+1} is the solution of (5)$\sim$(7), we discuss it into two cases:

- Unconstrained Optimization Problem

In this case, we have $\pi_{k+1} = \arg\max_\pi \mathbb{E}_{a\sim\pi}[Q^{\pi_k}(s,a)]$. Thus, it satisfies $\mathbb{E}_{a\sim\pi_{k+1}}[Q^{\pi_k}(s,a)] \geq \mathbb{E}_{a\sim\pi_k}[Q^{\pi_k}(s,a)]$. In a similar way to the proof of the soft policy improvement (Haarnoja et al., 2018a), we come to the following inequality:

$$
\begin{aligned}
Q^{\pi_k}(s_t, a_t) &= r(s_t, a_t) + \gamma\mathbb{E}_{s_{t+1},a_{t+1}\sim\pi_k}[Q^{\pi_k}(s_{t+1}, a_{t+1})] \\
&\leq r(s_t, a_t) + \gamma\mathbb{E}_{s_{t+1},a_{t+1}\sim\pi_{k+1}}[Q^{\pi_k}(s_{t+1}, a_{t+1})] \\
&\vdots \\
&= Q^{\pi_{k+1}}(s_t, a_t)
\end{aligned}
\tag{21}
$$

Thus, we can obtain that $Q^{\pi_{k+1}}(s,a) \geq Q^{\pi_k}(s,a)$.

- Constrained Optimization Problem

Based on Proposition A.2, we have the closed-form solution of π_{k+1} as

$$
\pi_{k+1} = \frac{1}{Z(s)}\mu^*(a|s)\,(f')^{-1}\left(Q^{\pi_k}(s,a)\right).
$$

Note that the definition of the offline optimal policy is $\mu_k^*(a|s) = \arg\max_{a\sim\mathcal{D}} Q^{\mu_k^*}(s,a)$. Then, when the constraint is effective, *i.e.*, $V^{\mu_k^*}(s) \geq V^{\pi_k}(s)$, we can derive that

$$
\begin{aligned}
\mathbb{E}_{a\sim\pi_{k+1}}[Q^{\pi_k}(s,a)] &= \int_{a\sim\mathcal{A}} \pi_{k+1}(a|s)Q^{\pi_k}(s,a)da \\
&= \int_{a\sim\mathcal{A}} \frac{1}{Z(s)}\mu^*(a|s)\,(f')^{-1}\left(Q^{\pi_k}(s,a)\right)Q^{\pi_k}(s,a)da \\
&= \int_{a\sim\mathcal{A}} \frac{1}{Z(s)}\left[\arg\max_{a\sim\mathcal{D}} Q^{\mu_k^*}(s,a)\right](f')^{-1}\left(Q^{\pi_k}(s,a)\right)Q^{\pi_k}(s,a)da \\
&\geq \int_{a\sim\mathcal{A}} \frac{1}{Z(s)}\left[\arg\max_{a\sim\mathcal{D}} Q^{\pi_k}(s,a)\right](f')^{-1}\left(Q^{\pi_k}(s,a)\right)Q^{\pi_k}(s,a)da \\
&\geq \int_{a\sim\mathcal{A}} \frac{1}{Z(s)}\pi_k(a|s)\,(f')^{-1}\left(Q^{\pi_k}(s,a)\right)Q^{\pi_k}(s,a)da \quad \lhd \text{ Satisfied in } \mathcal{D} \\
&\geq \int_{a\sim\mathcal{A}} \pi_k(a|s)Q^{\pi_k}(s,a)da \\
&= \mathbb{E}_{a\sim\pi_k}[Q^{\pi_k}(s,a)].
\end{aligned}
\tag{22}
$$

Thus, we reuse the results in the unconstrained optimization problem, we can have $Q^{\pi_{k+1}}(s,a) \geq Q^{\pi_k}(s,a)$.

Combining the results in both cases, we achieve $Q^{\pi_{k+1}}(s,a) \geq Q^{\pi_k}(s,a)$ for all $(s,a) \in \mathcal{D}$. $\qquad\square$

Proposition A.4. *Assume $|\mathcal{A}| < \infty$, repeating the alternation of the policy evaluation (3)$\sim$(4) and policy improvement (5)$\sim$(7) can make any online learning policy $\pi_k \in \Pi$ converge to the optimal policies π^*, s.t. $Q^{\pi^*}(s_t, a_t) \geq Q^{\pi_k}(s_t, a_t), \forall(s_t, a_t) \in \mathcal{S} \times \mathcal{A}$.*

Proof. Suppose Π is the policy set and π_k is the policy at iteration k. At each iteration, we guarantee the sequence Q^{π_k} is monotonically increasing through Proposition A.3. Besides, $\forall(s_t, a_t) \in \mathcal{S} \times \mathcal{A}$, Q^{π_i} would converge by repeatedly using the Bellman Expectation Equation as a γ-contraction mapping, which is proved in Lemma A.1. Thus, the sequence of π_k converges to some π^* that are local optimum. Then, we would show that π^* is indeed optimal. Using the same iterative argument as in the proof of Proposition A.3, the optimal policy π^* would satisfy that $Q^{\pi^*}(\mathbf{s},\mathbf{a}) \geq Q^{\pi'}(s,a)$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$. Hence, π^* are optimal in Π. $\qquad\square$

B. Implementation Details

In this section, we delve into the specific implementation details of OBAC. We use the same hyperparameters for all OBAC experiments in this paper. In terms of architecture, we use a simple 2-layer ELU network with a hidden size of 512 to parameterize all components, which contains: a Q-value network and a policy network for the online learning policy π; a Q-value network and a V-value network for the offline optimal policy μ^*.

Specifically, to encourage online learning policy exploration, we utilize a max-entropy framework (Haarnoja et al., 2018b) for π with automatic temperature tuning.

Table 1. The hyperparameters of OBAC

	Hyperparameter	Value
	Optimizer	Adam
	Critic learning rate	3e-4
	Actor learning rate	3e-4
	Discount factor	0.99
OBAC Hyperparameters	Mini-batch	512
	Actor Log Std. Clipping	$(-20, 2)$
	Replay buffer size	1e6
	Expectile factor τ	0.9
	Behavior clone weight λ	0.001
	Network hidden dim	512
Architecture $\times 4$	Network hidden layers	2
	Network activation function	elu

B.1. Hyper-parameters

In all of our experiments, we use a single set of hyper-parameters with $\tau = 0.9$ and $\lambda = 0.001$. Here, a big expectile factor τ can approach the maximization of offline Q value in the replay buffer, thus enabling better online policy learning. To balance the training stability and optimality of offline policy, we choose $\tau = 0.9$. For the behavior clone weight λ which is from the solution within the KKT condition, we can apply dual gradient descent for auto-tuning in principle, while we find a fixed $\lambda = 0.001$ can achieve satisfied performance. Besides, several offline RL works (Kumar et al., 2019; Fujimoto & Gu, 2021) also use fixed weight for policy constraint. Thus, we apply a fixed λ in our works.

B.2. Baselines and Environments

In our experiments, we have implemented SAC, TD3 and TD-MPC2 using their original code bases to ensure a fair and consistent comparison.

- For SAC (Haarnoja et al., 2018a), we utilized the open-source PyTorch implementation, available at `https://github.com/pranz24/pytorch-soft-actor-critic`.

- TD3 (Fujimoto et al., 2018) was integrated into our experiments through its official codebase, accessible at `https://github.com/sfujim/TD3`.

- TD-MPC2 (Hansen et al., 2024) was employed with its official implementation from `https://github.com/nicklashansen/tdmpc2`.

For BAC (Ji et al., 2023), we reproduce the proposed BEE operator. Specifically, the Bellman Exploitation operator $\mathcal{T}^{\mu}_{exploit}$ is implemented by IQL (Kostrikov et al., 2021), and the Bellman Exploration operator $\mathcal{T}^{\pi}_{explore}$ with the entropy exploration term is based on SAC (Haarnoja et al., 2018a), both of which are suggested in its original paper. We choose the trade-off hyper-parameter in BAC as 0.5 and expectile factor 0.7 aligning with its most suggestion.

We use the official setting of each task domain, including the reward setting, the task horizon, the *done* signal and the original state-action spaces.

C. More Experimental Results

We provide complete experimental results to show the superiority of OBAC.
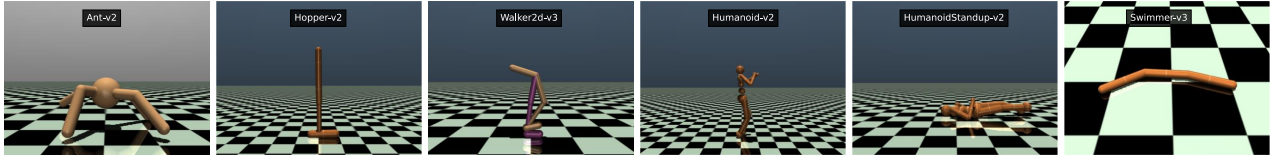
C.1. Task Visualization

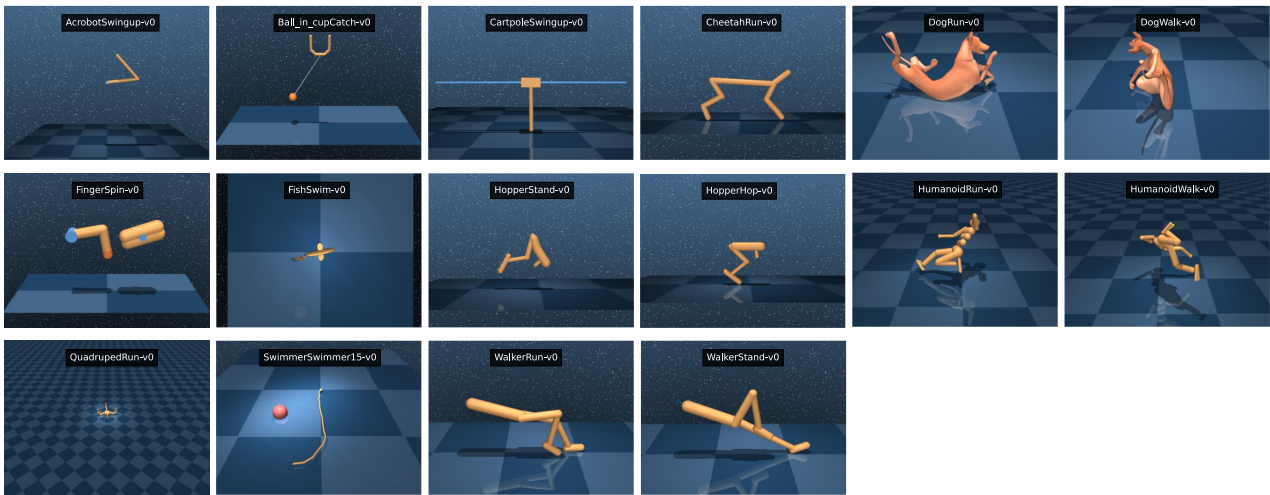*Figure 7.* Visualization of tasks in **Mujoco**.

Figure 8. Visualization of tasks in **DM Control**.
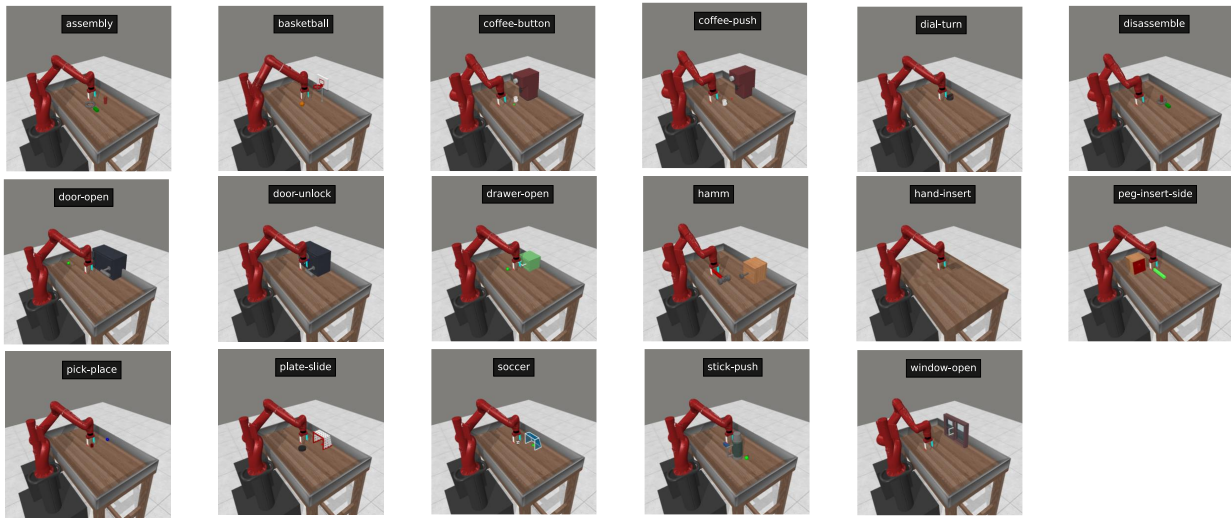
Figure 9. Visualization of tasks in **Meta-World**.

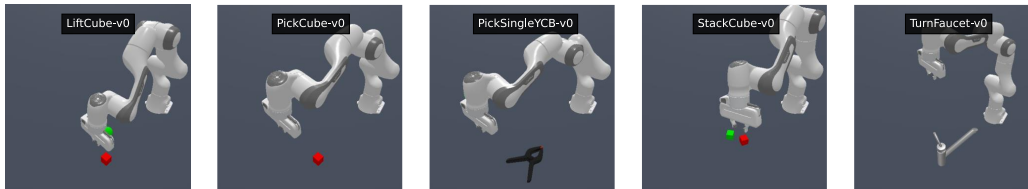*Figure 10.* Visualization of tasks in **Adroit**.

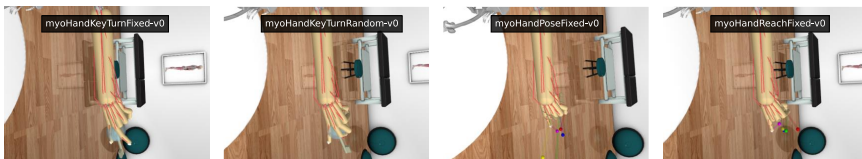*Figure 11.* Visualization of tasks in **Maniskill2**.

Figure 12. Visualization of tasks in **Myosuite**.

C.2. Complete Experimental Results

We note that, since the task-specific *done* signal setting, TD-MPC2 may perform poorly in Mujoco suite, especially for Ant, Hopper, Walker2d and Humanoid, where the episode may be terminated early if the robot falls. In such cases, TD-MPC2 can not find a feasible policy to prevent the fall thus achieving limited performance. Besides, we did not find the results of Mujoco in TD-MPC2's paper.

However, in the other suites, we find TD-MPC2 can perform well even within unseen tasks in its original paper. Thus, we think the reproduction results are reasonable. Note that, in these suites, the episode would be done only when the task horizon comes to an end, which may provide more exploration information compared with Mujoco suite.

For the consideration of fair comparison, we follow the official setting of each task suite when evaluating all algorithms.
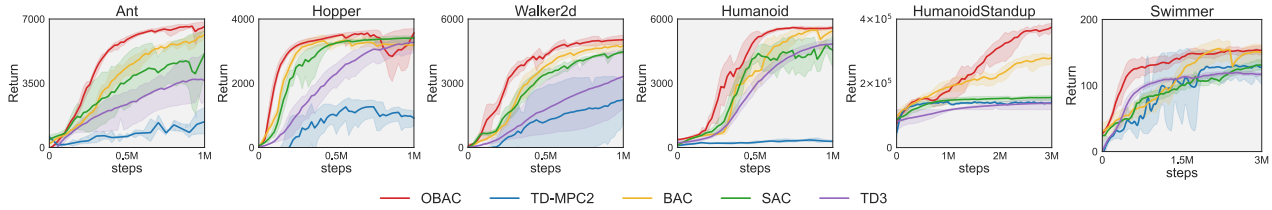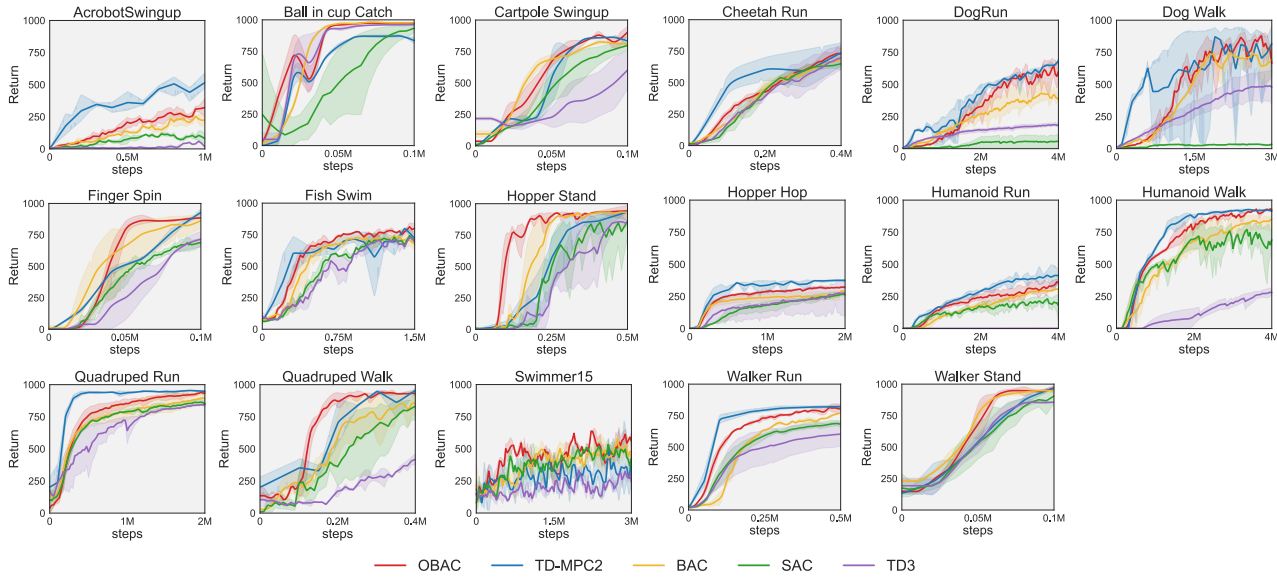

Figure 13. The results of **6** tasks in **Mujoco**.

Figure 14. The results of **17** tasks in **DM Control**.

18
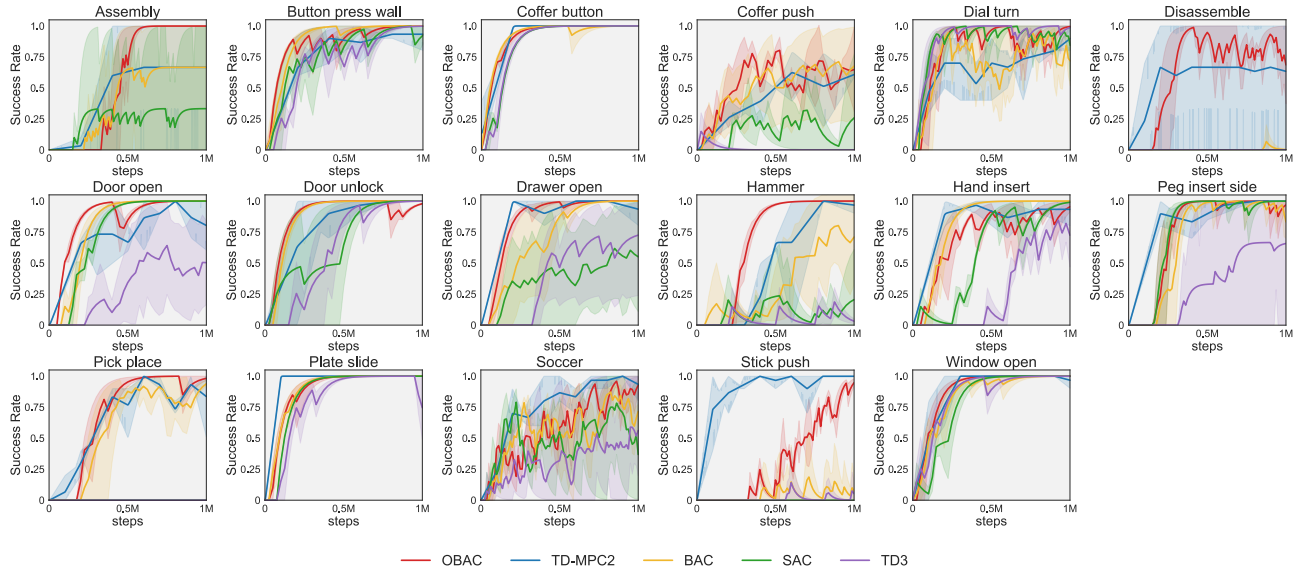
Figure 15. The results of **17** tasks in **Meta-World**.
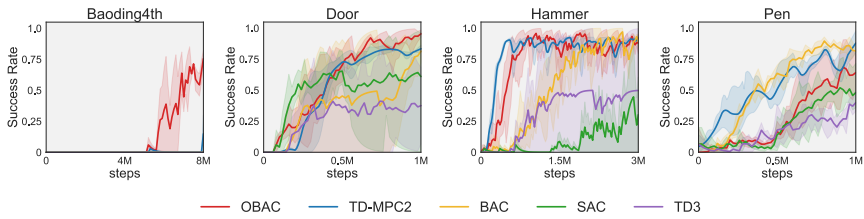
Figure 16. The results of **4** tasks in **Adroit**.
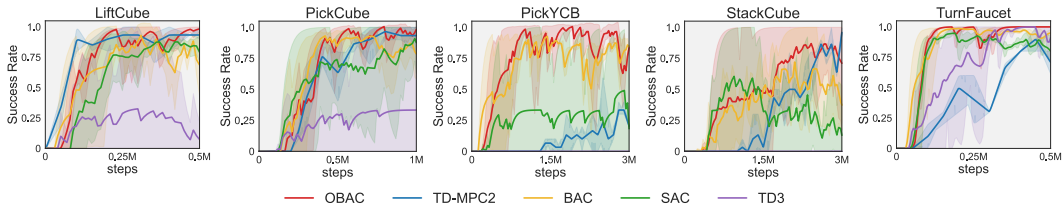
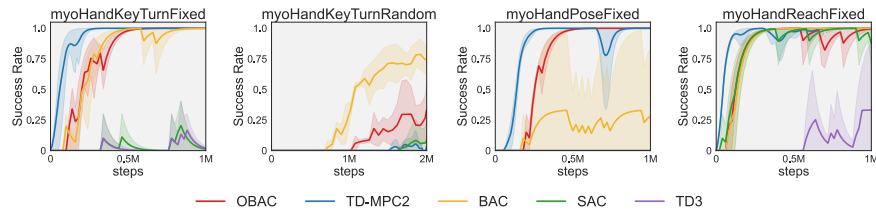*Figure 17.* The results of **5** tasks in **Maniskill2**.

Figure 18. The results of **4** tasks in **Myosuite**.

19

C.3. Performance comparison under noise tasks

To better explain the robustness of OBAC, we provide the performance decline rates of OBAC and baselines here, whose results are similar to Figure 6a. Our results show that OBAC exhibits a lower performance decline than the baselines.

- Given the same noise level ($\sigma = 0.1$)

Table 2. Performance comparison of OBAC and baselines with the same noise level

Task (success rate)	OBAC	BAC	SAC
Hammer ($\sigma = 0$)	0.9	0.9	0.4
Hammer ($\sigma = 0.1$)	0.8	0.75	0.35
Decline rate	**11.11%**	16.67%	12.5%
Pen ($\sigma = 0$)	0.6	0.75	0.5
Pen ($\sigma = 0.1$)	0.45	0.25	0.15
Decline rate	**25.00%**	66.67%	70.00%

- Given the different noise level ($\sigma = 0.1$ and $\sigma = 0.05$)

Table 3. Performance comparison of OBAC with different noise levels

Task (success rate)	OBAC
Hammer ($\sigma = 0.05$)	0.85
Hammer ($\sigma = 0.1$)	0.8
Decline rate	**5.88%**
Pen ($\sigma = 0.05$)	0.5
Pen ($\sigma = 0.1$)	0.45
Decline rate	**10.00%**

D. Additional Ablations

Except for the ablation studies in the main paper, we additionally provide the results of OBAC's variant to assess OBAC completely.

Variant of OBAC. In our implementation, we employ the stochastic Gaussian policy for online policy learning. On the other side, we derive a variant of OBAC by using a deterministic policy for the online learning policy. Thus, the policy objective can be

$$\arg \min_{\pi_\theta \in \Pi} \mathbb{E}_{s \sim \mathcal{D}} \Big\{ \mathbb{E}_{a \sim \pi_\theta} [-Q^{\pi_k}(s,a)] - \lambda \mathbb{E}_{a \sim \mathcal{D}} \left[(\pi_\theta(s) - a)^2 \right] \mathbb{1} \left(V^{\mu_k^*}(s) - V^{\pi_k}(s) \right) \Big\}. \tag{23}$$

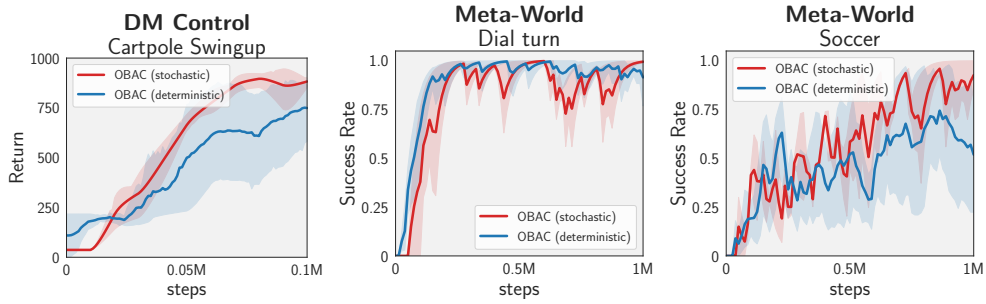We conduct several experiments on such a variant. The results show that our method can also

Figure 19. Comparison of OBAC and its variant.