Adversarially Trained Actor Critic for Offline Reinforcement Learning

Ching-An Cheng [*1] **Tengyang Xie** [*2] **Nan Jiang** [2] **Alekh Agarwal** [3]

Abstract

We propose Adversarially Trained Actor Critic (ATAC), a new model-free algorithm for offline reinforcement learning (RL) under insufficient data coverage, based on the concept of *relative pessimism*. ATAC is designed as a two-player Stackelberg game: A policy actor competes against an adversarially trained value critic, who finds data-consistent scenarios where the actor is inferior to the data-collection behavior policy. We prove that, when the actor attains no regret in the two-player game, running ATAC produces a policy that provably *1)* outperforms the behavior policy over a wide range of hyperparameters that control the degree of pessimism, and *2)* competes with the best policy covered by data with appropriately chosen hyperparameters. Compared with existing works, notably our framework offers both theoretical guarantees for general function approximation and a deep RL implementation scalable to complex environments and large datasets. In the D4RL benchmark, ATAC consistently outperforms state-of-the-art offline RL algorithms on a range of continuous control tasks.

1. Introduction

Online reinforcement learning (RL) has been successfully applied in many simulation domains (Mnih et al., 2015; Silver et al., 2016), demonstrating the promise of solving sequential decision making problems by direct exploratory interactions. However, collecting diverse interaction data is prohibitively expensive or infeasible in many real-world applications such as robotics, healthcare, and conversational agents. Due to these problems' risk-sensitive nature, data can only be collected by behavior policies that satisfy certain baseline performance or safety requirements.

[*]Equal contribution [1]Microsoft Research [2]University of Illinois at Urbana-Champaign [3]Google Research. Correspondence to: Ching-An Cheng <chinganc@microsoft.com>.

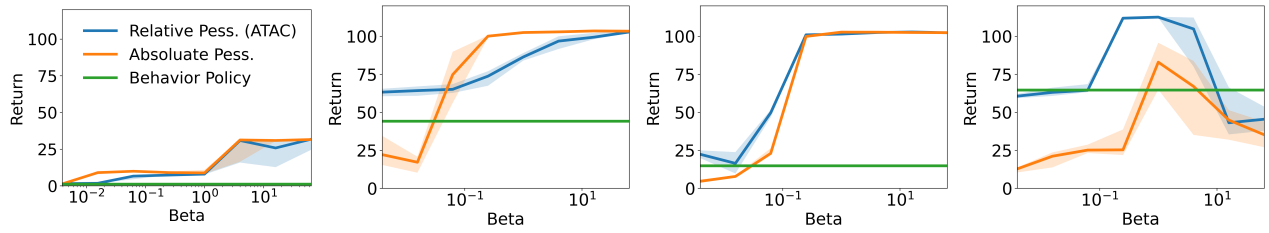
Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

The restriction on real-world data collection calls for *offline* RL algorithms that can reliably learn with historical experiences that potentially have limited coverage over the state-action space. Ideally, an offline RL algorithm should *1)* always improve upon the behavior policies that collected the data, and *2)* learn from large datasets to outperform any other policy whose state-action distribution is well covered by the data. The first condition is known as safe policy improvement (Fujimoto et al., 2019; Laroche et al., 2019), and the second is a form of learning consistency, that the algorithm makes the best use of the available data.

In particular, it is desirable that the algorithm can maintain safe policy improvement across *large* and *anchored* hyperparameter choices, a property we call *robust policy improvement*. Since offline hyperparameter selection is a difficult open question (Paine et al., 2020; Zhang & Jiang, 2021), robust policy improvement ensures the learned policy is always no worse than the baseline behavior policies and therefore can be reliably deployed in risk-sensitive decision making applications. For example, in healthcare, it is only ethical to deploy new treatment policies when we confidently know they are no worse than existing ones. In addition, robust policy improvement makes tuning hyperparameters using additional online interactions possible. While online interactions are expensive, they are not completely prohibited in many application scenarios, especially when the tested policies are no worse than the behavior policy that collected the data in the first place. Therefore, if the algorithm has robust policy improvement, then its performance can potentially be more directly tuned.

However, few existing works possess all the desiderata above. Regarding consistency guarantees, deep offline RL algorithms (e.g. Kumar et al., 2020; Kostrikov et al., 2021) show strong empirical performance, but are analyzed theoretically in highly simplified tabular cases. Theoretical works (Liu et al., 2020; Jin et al., 2021; Xie et al., 2021; Uehara et al., 2021) provide systematic analyses of learning correctness and consistency, but most of them have little empirical evaluation (Liu et al., 2020) or consider only the linear case (Jin et al., 2021; Zanette et al., 2021).

Turning to the robust policy improvement property, this is relatively rare in state-of-the-art offline RL literature. Behavior regularization approaches (Fujimoto et al., 2019; Kumar



(a) hopper-random (b) hopper-medium (c) hopper-medium-replay (d) hopper-medium-expert

Figure 1. Robust Policy Improvement. ATAC based on relative pessimism improves from behavior policies over a wide range of hyperparameters (β) that controls the degree of pessimism, and has a known safe policy improvement anchor point at $\beta = 0$. Thus, we can gradually increase β from zero to online tune ATAC, while not violating the performance baseline of the behavior policy. By contrast, offline RL based on absolute pessimism (e.g., Xie et al., 2021) has safe policy improvement only for well-tuned hyperparameters. The differences are most stark in panel (d) where ATAC outperforms behavior for β ranging over 3 orders of magnitude (0.01 to 10), compared with the narrow band of choices for absolute pessimism. The plots show the 25^{th}, 50^{th}, 75^{th} percentiles over 10 random seeds.

et al., 2019; Wu et al., 2019; Laroche et al., 2019; Fujimoto & Gu, 2021) are scalable and show robustness for a broad range of hyperparameters that controls the pessimism degree; however, they are often more conservative, which ultimately limits the policy performance, as their robustness is achieved by a proximity regularization/constraint that ignores the reward information. Some pessimistic algorithms (Liu et al., 2020; Xie et al., 2021) have safe policy improvement guarantees but only for carefully selected hyperparameters. For a more detailed discussion of related works, see Appendix A.

In this paper, we propose a new model-free offline RL algorithm, Adversarially Trained Actor Critic (ATAC). Compared with existing works, ATAC *1)* enjoys strong theoretical guarantees on robust policy improvement over hyperparameter that controls the pessimism degree and learning consistency for nonlinear function approximators, and *2)* has a scalable implementation that can learn with deep neural networks and large datasets.

ATAC is designed based on the concept of relative pessimism, leading to a two-player Stackelberg game formulation of offline RL. We treat the actor policy as the leader that aims to perform well under a follower critic, and adversarially train the critic to find Bellman-consistent (Xie et al., 2021) scenarios where the actor is inferior to the behavior policy. Under standard function-approximation assumptions, we prove that, when the actor attains no regret in the two-player game, ATAC produces a policy that provably outperforms the behavior policies for a large anchored range of hyperparameter choices and is optimal when the offline data covers scenarios visited by an optimal policy.

We also provide a practical implementation of ATAC based on stochastic first-order two-timescale optimization. In particular, we propose a new Bellman error surrogate, called double Q residual algorithm (DQRA) loss, which is inspired by a related idea of Wang & Ueda (2021) and combines the double Q heuristic (Fujimoto et al., 2018) and the residual algorithm (Baird, 1995) to improve the optimization stability of offline RL. We test ATAC on the D4RL benchmark (Fu et al., 2020), and ATAC consistently outperforms state-of-the-art baselines across multiple continuous-control problems. These empirical results also validate the robust policy improvement property of ATAC (Fig. 1), which makes ATAC suitable for risk sensitive applications. The code is available at `https://github.com/microsoft/ATAC`.

2. Preliminaries

Markov Decision Process We consider RL in a Markov Decision Process (MDP) $\mathcal{M}$, defined by $(\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma)$. $\mathcal{S}$ is the state space, and $\mathcal{A}$ is the action space. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition function, where $\Delta(\cdot)$ denotes the probability simplex, $R : \mathcal{S} \times \mathcal{A} \to [0, R_{\max}]$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. Without loss of generality, we assume that the initial state of the MDP, s_0, is deterministic. We use $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ to denote the learner's decision-making policy, and $J(\pi) := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t | a_t \sim \pi(\cdot|s_t)]$ to denote the expected discounted return of π, with $r_t = R(s_t, a_t)$. The goal of RL is to find a policy that maximizes $J(\cdot)$. For any policy π, we define the Q-value function as $Q^\pi(s, a) := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t | (s_0, a_0) = (s, a), a_t \sim \pi(\cdot|s_t)]$. By the boundedness of rewards, we have $0 \le Q^\pi \le \frac{R_{\max}}{1-\gamma} =: V_{\max}$. For a policy π, the Bellman operator $\mathcal{T}^\pi$ is defined as $(\mathcal{T}^\pi f)(s, a) := R(s, a) + \gamma \mathbb{E}_{s'|s,a}[f(s', \pi)]$, where $f(s', \pi) := \sum_a \pi(a'|s') f(s', a')$. In addition, we use d^π to denote the normalized and discounted state-action occupancy measure of the policy π. That is, $d^\pi(s, a) := (1-\gamma)\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}(s_t = s, a_t = a) | a_t \sim \pi(\cdot|s_t)]$. We also use $\mathbb{E}_\pi$ to denote expectations with respect to d^π.

Offline RL The goal of offline RL is to compute good policies based pre-collected offline data without environ-

ment interaction. We assume the offline data $\mathcal{D}$ consists of N i.i.d. (s, a, r, s') tuples, where $(s, a) \sim \mu$, $r = R(s, a)$, $s' \sim \mathcal{P}(\cdot|s, a)$. We also assume μ is the discounted state-action occupancy of some *behavior policy*, which we also denote as μ with abuse of notation (i.e., $\mu = d^\mu$). We will use $a \sim \mu(\cdot|s)$ to denote actions drawn from this policy, and also $(s, a, s') \sim \mu$ to denote $(s, a) \sim \mu$ and $s' \sim P(\cdot|s, a)$.

Function Approximation We assume access to a value-function class $\mathcal{F} \subseteq (\mathcal{S} \times \mathcal{A} \to [0, V_{\max}])$ to model the Q-functions of policies, and we search for good policies from a policy class $\Pi \subseteq (\mathcal{S} \to \Delta(\mathcal{A}))$. The combination of $\mathcal{F}$ and Π is commonly used in the literature of actor-critic or policy-based approaches (see, e.g., Bertsekas & Tsitsiklis, 1995; Konda & Tsitsiklis, 2000; Haarnoja et al., 2018). We now recall some standard assumptions on the expressivity of the value function class $\mathcal{F}$ which are needed for actor-critic methods, particularly in an offline setting.

Assumption 1 (Approximate Realizability)**.** For any policy $\pi \in \Pi$, $\min_{f \in \mathcal{F}} \max_{\text{admissible } \nu} \|f - \mathcal{T}^\pi f\|^2_{2,\nu} \le \varepsilon_{\mathcal{F}}$, where admissibilty ν means $\nu \in \{d^{\pi'} : \forall \pi \in \Pi\}$.

Assumption 1 is a weaker form of stating $Q^\pi \in \mathcal{F}, \forall \pi' \in \Pi$. This realizability assumption is the same as the one made by Xie et al. (2021) and is weaker than assuming a small error in ℓ_∞ norm (Antos et al., 2008).

Assumption 2 (Approximate Completeness)**.** For any $\pi \in \Pi$ and $f \in \mathcal{F}$, we have $\min_{g \in \mathcal{F}} \|g - \mathcal{T}^\pi f\|^2_{2,\mu} \le \varepsilon_{\mathcal{F},\mathcal{F}}$.

Here $\|\cdot\|_{2,\mu} \coloneqq \sqrt{\mathbb{E}_\mu[(\cdot)^2]}$ denotes the μ-weighted 2-norm.[1] Again Assumption 2 weakens the typical completeness assumption of $\mathcal{T}^\pi f \in \mathcal{F}$ for all $f \in \mathcal{F}$, which is commonly assumed in the analyses of policy optimization methods with TD-style value function learning. We require the approximation to be good *only* on the data distribution.

3. A Game Theoretic Formulation of Offline RL with Robust Policy Improvement

In this section, we introduce the idea of relative pessimism and propose a new Stackelberg game (Von Stackelberg, 2010) formulation of offline RL, which is the foundation of our algorithm ATAC. For clarity, in this section we discuss solution concepts at the population level instead of using samples. This simplification is for highlighting the uncertainty in decision making due to missing coverage in the data distribution μ that offline RL faces. We will consider the effects of finite sample approximation when we introduce ATAC in Section 4.

[1] We will use the notation $\|f\|_{2,\mathcal{D}}$ for an empirical distribution d of a dataset $\mathcal{D}$, where $\|f\|_{2,\mathcal{D}} = \sqrt{\frac{1}{|\mathcal{D}|}\sum_{(s,a)\in\mathcal{D}} f(s,a)^2}$.

3.1. A Stackelberg Game Formulation of Offline RL

Stackelberg game A Stackelberg game is a sequential game between a leader and a follower. It can be stated as a bilevel optimization problem, $\min_x g(x, y_x)$, s.t. $y_x \in \operatorname{argmin}_y h(x, y)$ where the leader and the follower are the variables x and y, respectively, and g, h are their objectives. The concept of Stackelberg game has its origins in the economics literature and has been recently applied to design *online* model-based RL (Rajeswaran et al., 2020) and *online* actor critic algorithms (Zheng et al., 2021). The use of this formalism in an offline setting here is novel to our knowledge. Stackelberg games also generalize previous minimax formulations (Xie et al., 2021), which correspond to a two-player zero-sum game with $h = -g$.

Offline RL as a Stackelberg game Inspired by the minimax offline RL concept by Xie et al. (2021) and the pessimistic policy evaluation procedure by Kumar et al. (2020), we formulate the Stackelberg game for offline RL as a bilevel optimization problem, with the learner policy $\pi \in \Pi$ as the leader and a critic $f \in \mathcal{F}$ as the follower:

$$\widehat{\pi}^* \in \operatorname*{argmax}_{\pi \in \Pi} \mathcal{L}_\mu(\pi, f^\pi) \tag{1}$$
$$\text{s.t.} \quad f^\pi \in \operatorname*{argmin}_{f \in \mathcal{F}} \mathcal{L}_\mu(\pi, f) + \beta \mathcal{E}_\mu(\pi, f)$$

where $\beta \ge 0$ is a hyperparamter, and

$$\mathcal{L}_\mu(\pi, f) \coloneqq \mathbb{E}_\mu[f(s, \pi) - f(s, a)] \tag{2}$$
$$\mathcal{E}_\mu(\pi, f) \coloneqq \mathbb{E}_\mu[((f - \mathcal{T}^\pi f)(s, a))^2]. \tag{3}$$

Intuitively, $\widehat{\pi}^*$ attempts to maximize the value predicted by f^π, and f^π performs a *relatively* pessimistic policy evaluation of a candidate π with respect to the behavior policy μ (we will show $\mathcal{L}_\mu(\pi, f)$ aims to estimate $(1 - \gamma)(J(\pi) - J(\mu))$). In the definition of f^π, $\mathcal{E}_\mu(\pi, f)$ ensures f^π's (approximate) Bellman-consistency on data and $\mathcal{L}_\mu(\pi, f)$ promotes pessimism with β being the hyperparameter that controls their relative contributions. In the rest of this section, we discuss how the relative pessimistic formulation in Eq.(1) leads to the desired property of *robust* policy improvement, and compare it to the solution concepts used in the previous offline RL works.

3.2. Relative Pessimism and Robust Policy Improvement

Our design of the Stackelberg game in Eq.(1) is motivated by the benefits of robust policy improvement in β given by relative pessimism. As discussed in the introduction, such property is particularly valuable to applying offline RL in risk-sensitive applications, because it guarantees the learned policy is no worse than the behavior policy regardless of the hyperparameter choice and allows potentially direct online performance tuning. Note that prior works (e.g., Liu et al., 2020; Xie et al., 2021) have relied on well-chosen hyperparameters to show improvement upon the behavior policy

(i.e., safe policy improvement). We adopt the term *robust* policy improvement here to distinguish from those weaker guarantees. While there are works (Laroche et al., 2019; Fujimoto et al., 2019; Kumar et al., 2019) that provide robust policy improvement in tabular problems, but their heuristic extensions to function approximation lose this guarantee.

Below we show that the solution $\widehat{\pi}^*$ in Eq.(1) is no worse than the behavior policy for *any* $\beta \geq 0$ under Assumption 1. This property is because in Eq.(1) the actor is trying to optimize a lower bound on the relative performance $(1-\gamma)(J(\pi)-J(\mu))$ for π, and this lower bound is tight (exactly zero) at the behavior policy μ, for any $\beta \geq 0$.

Proposition 3. *If Assumption 1 holds with $\varepsilon_{\mathcal{F}} = 0$ and $\mu \in \Pi$, then $\mathcal{L}_\mu(\pi, f^\pi) \leq (1-\gamma)(J(\pi)-J(\mu))\ \forall \pi \in \Pi$, for* any *$\beta \geq 0$. This implies $J(\widehat{\pi}^*) \geq J(\mu)$.*

Proof. By performance difference lemma (Kakade & Langford, 2002), $J(\pi) - J(\mu) = \frac{1}{1-\gamma}\mathbb{E}_\mu[Q^\pi(s,\pi) - Q^\pi(s,a)]$. Therefore, if $Q^\pi \in \mathcal{F}$ on states of μ, then $(1-\gamma)(J(\pi) - J(\mu)) = \mathcal{L}_\mu(\pi, Q^\pi) = \mathcal{L}_\mu(\pi, Q^\pi) + \beta\mathcal{E}(Q^\pi, \pi) \geq \mathcal{L}_\mu(\pi, f^\pi) + \beta\mathcal{E}(f^\pi, \pi) \geq \mathcal{L}_\mu(\pi, f^\pi)$, where we use $\mathcal{E}(\pi, Q^\pi) = 0$ by definition of Q^π and $\mathcal{E}(\pi, f) \geq 0$ for any $f \in \mathcal{F}$. Robust policy improvement follows, as $J(\widehat{\pi}^*) - J(\mu) \geq \mathcal{L}_\mu(\widehat{\pi}^*, f^{\widehat{\pi}^*}) \geq \mathcal{L}_\mu(\mu, f^\mu) = 0$. □

Relative vs. absolute pessimism Our formulation is inspired by the maximin objective of Xie et al. (2021), which optimizes a pessimistic estimate of $J(\pi)$ (which we call *absolute pessimism*) and learns a good policy with a well-chosen value of β. In contrast, our relative pessimism formulation optimizes the performance of π *relative* to $J(\mu)$, i.e., $J(\pi) - J(\mu)$. As Section 4.1 will show, algorithms based on both formulations enjoy similar optimality guarantees with well-chosen hyperparameters. But absolute pessimism gives policy improvement only for certain hyperparameters, while the relative approach enjoys robust policy improvement for all $\beta \geq 0$, which is practically significant.

Improvement beyond behavior policy It is clear from Proposition 3 that the objective in Eq.(1) results in the optimization of a lower bound on the value gap $(1-\gamma)(J(\pi) - J(\mu))$. On the other hand, for appropriate settings of β, it turns out that this lower bound is not too loose for any $\pi \in \Pi$ such that d^π is covered by the data distribution μ, as implicitly shown in our detailed theoretical analysis presented in the next section. Consequently, maximizing the objective Eq.(1) generally results in policies that significantly outperform μ for appropriate choices of β, as long as the data has support for at least one such policy.

Imitation learning perspective An alternative interpretation of Proposition 3 follows from examining the special case of $\beta = 0$ (i.e. not using any information of rewards and transitions). In this case, the objective Eq.(1) reduces to the maximin problem: $\max_{\pi\in\Pi} \min_{f\in\mathcal{F}} \mathcal{L}_\mu(\pi, f)$, which always yields robust policy improvement under Assumption 1. More generally, if the function class $\mathcal{F}$ is rich enough to approximate all bounded, Lipschitz functions, then the above objective with $\beta = 0$ resembles behavior cloning to match the occupancy measures of π and μ using an integral probability metric (IPM; Müller, 1997) (or equivalently, Wasserstein GAN; Arjovsky et al., 2017). With $\beta > 0$, the algorithm gets more information and thus intuitively can perform better. This perspective shows how our formulation unifies the previously disparate literature on behavior regularization and pessimism.

4. Adversarially Trained Actor Critic

We design our new model-free offline RL algorithm, Adversarially Trained Actor Critic (ATAC), based on the Stackelberg game of relative pessimism in Section 3.[2] In the following, we first describe a theoretical version of ATAC (Algorithm 1) in Section 4.1, which is based on a no-regret policy optimization oracle and a pessimistic policy evaluation oracle. We discuss its working principles and give theoretical performance guarantees. This theoretical algorithm further serves as a template that provides design principles for implementing ATAC. To show its effectiveness, in Section 4.2, we design Algorithm 2, a practical deep-learning implementation of ATAC. Algorithm 2 is a two-timescale first-order algorithm based on stochastic approximation, and uses a novel Bellman error surrogate (called double-Q residual algorithm loss) for off-policy optimization stability. Later in Section 5, we empirically demonstrate that the principally designed Algorithm 2 outperforms many state-of-the-art offline RL algorithms.

4.1. Theory of ATAC with Optimization Oracles

This section instantiates a version of the ATAC algorithm with abstract optimization oracles for the leader and follower, using the concepts introduced in Section 3. We first define the empirical estimates of $\mathcal{L}_\mu$ and $\mathcal{E}_\mu$ as follows. Given a dataset $\mathcal{D}$, we define

$$\mathcal{L}_\mathcal{D}(f, \pi) := \mathbb{E}_\mathcal{D}\left[f(s,\pi) - f(s,a)\right], \tag{4}$$

and the estimated Bellman error (Antos et al., 2008)

$$\begin{aligned}\mathcal{E}_\mathcal{D}(f,\pi) := \mathbb{E}_\mathcal{D}\left[\left(f(s,a) - r - \gamma f(s',\pi)\right)^2\right] \\ - \min_{f'\in\mathcal{F}} \mathbb{E}_\mathcal{D}\left[\left(f'(s,a) - r - \gamma f(s',\pi)\right)^2\right].\end{aligned} \tag{5}$$

4.1.1. Algorithm

Using these definitions, Algorithm 1 instantiates a version of the ATAC approach. At a high-level, the k^{th} iteration of

[2] One can also use the formulation to design model-based algorithms, by constructing f as a Q-function $\widehat{Q}^\pi_\theta$ computed from a model parameterized by θ, and using $\mathcal{E}_\mu(\pi, \widehat{Q}^\pi_\theta)$ to capture the reward and transition errors of the model θ.

Algorithm 1 ATAC (Theoretical Version)

Input: Batch data $\mathcal{D}$. coefficient β.

1: Initialize policy π_1 as the uniform policy.
2: **for** $k = 1, 2, \ldots, K$ **do**
3: Obtain the pessimistic estimation of π_k as f_k,
$f_k \leftarrow \operatorname{argmin}_{f \in \mathcal{F}_k} \mathcal{L}_{\mathcal{D}}(f, \pi_k) + \beta \mathcal{E}_{\mathcal{D}}(f, \pi_k)$.
4: Compute π_{k+1} by
$\pi_{k+1} \leftarrow \mathsf{PO}(\pi_k, f_k, \mathcal{D})$,
where PO denotes a no-regret oracle (Def. 4).
5: **end for**
6: Output $\bar{\pi} := \mathsf{Unif}(\pi_{[1:K]})$. ▷ *uniformly mix $\pi_1, \ldots, \pi_K$ at the trajectory level*

the algorithm first finds a critic f_k that is maximally pessimistic for the current actor π_k along with a regularization based on the estimated Bellman error of π_k (line 3), with a hyperparameter β trading off the two terms. The actor π_{k+1} then invokes a no-regret policy optimization oracle to update its policy, given f_k (line 4). We now discuss some of the key aspects of the algorithm.

Policy optimization with a no-regret oracle In Algorithm 1, the policy optimization step (Line 4) is conducted by calling a no-regret policy optimization oracle (PO). We now define the property we expect from this oracle.

Definition 4 (No-regret policy optimization oracle)**.** An algorithm PO is called a *no-regret policy optimization oracle* if for any sequence of functions[3] $f_1, \ldots, f_K$ with $f_k : \mathcal{S} \times \mathcal{A} \to [0, V_{\max}]$, the policies $\pi_1, \ldots, \pi_K$ produced by PO satisfy, for any comparator $\pi \in \Pi$:

$$\varepsilon_{\mathsf{opt}}^{\pi} := \frac{1}{1-\gamma} \sum_{k=1}^{K} \mathbb{E}_{\pi} \left[f_k(s, \pi) - f_k(s, \pi_k) \right] = o(K).$$

The notion of regret used in Definition 4 nearly corresponds to the standard regret definition in online learning (Cesa-Bianchi & Lugosi, 2006), except that we take an expectation over states as per the occupancy measure of the comparator. Algorithmically, a natural oracle might perform online learning with states and actions sampled from μ in the offline RL setting. This mismatch of measures between the optimization objective and regret definition is typical in policy optimization literature (see e.g. Kakade & Langford, 2002; Agarwal et al., 2021). One scenario in which we indeed have such an oracle is when PO corresponds to running a no-regret algorithm *separately in each state*[4] and the policy class is sufficiently rich to approximate the resulting iterates. There is a rich literature on such approaches using mirror-descent style methods (e.g., Neu et al., 2017; Geist et al., 2019), of which a particularly popular instance is soft policy iteration or natural policy gradient (Kakade, 2001) based on multiplicative weight updates (e.g. Even-Dar et al., 2009; Agarwal et al., 2021): $\pi_{k+1}(a|s) \propto \pi_k(a|s) \exp\left(\eta f_k(s,a)\right)$ with $\eta = \sqrt{\frac{\log|\mathcal{A}|}{2V_{\max}^2 K}}$. This oracle is used by Xie et al. (2021), which leads to the regret bound $\varepsilon_{\mathsf{opt}}^{\pi} \leq \mathcal{O}\left(\frac{V_{\max}}{1-\gamma}\sqrt{K \log|\mathcal{A}|}\right)$.

[3] $\{f_k\}_{k=1}^K$ can be generated by an adaptive adversary.

[4] The computational complexity of doing so does *not* depend on the size of the state space, since we only need to run the algorithm on states observed in the data. See (Xie et al., 2021).

4.1.2. Theoretical Guarantees

We now provide the theoretical analysis of Algorithm 1. Recall that with missing coverage, we can only hope to compete with policies whose distributions are well-covered by data, and we need a quantitative measurement of such coverage. Following Xie et al. (2021), we use $\mathscr{C}(\nu; \mu, \mathcal{F}, \pi) := \max_{f \in \mathcal{F}} \frac{\|f - \mathcal{T}^{\pi} f\|_{2,\nu}^2}{\|f - \mathcal{T}^{\pi} f\|_{2,\mu}^2}$ to measure how well a distribution of interest ν (e.g., d^{π}) is covered by the data distribution μ w.r.t. policy π and function class $\mathcal{F}$, which is a sharper measure than the more typical concentrability coefficients (Munos & Szepesvári, 2008) (e.g., $\mathscr{C}(\nu; \mu, \mathcal{F}, \pi) \leq \max_{s,a} \nu(s,a)/\mu(s,a)$).

We also use $d_{\mathcal{F},\Pi}$ to denote the joint statistical complexity of the policy class Π and $\mathcal{F}$. For example, when $\mathcal{F}$ and Π are finite, we have $d_{\mathcal{F},\Pi} = \mathcal{O}(\log |\mathcal{F}||\Pi|/\delta)$, where δ is a failure probability. Our formal proofs utilize the covering number to address infinite function classes; see Appendix B for details. In addition, we also omit the approximation error terms $\varepsilon_{\mathcal{F}}$ and $\varepsilon_{\mathcal{F},\mathcal{F}}$ in the results presented in this section for the purpose of clarity. The detailed results incorporating these terms are provided in Appendix B.

Theorem 5 (Informal)**.** *Let $|\mathcal{D}| = N$, $C \geq 1$ be any constant, $\nu \in \Delta(\mathcal{S} \times \mathcal{A})$ be an arbitrarily distribution that satisfies $\max_{k \in [K]} \mathscr{C}(\nu; \mu, \mathcal{F}, \pi_k) \leq C$, and $\pi \in \Pi$ be an arbitrary competitor policy. Then, when $\varepsilon_{\mathcal{F}} = \varepsilon_{\mathcal{F},\mathcal{F}} = 0$, choosing $\beta = \Theta\left(\sqrt[3]{\frac{V_{\max} N^2}{d_{\mathcal{F},\Pi}^2}}\right)$, with high probability:*

$$J(\pi) - J(\bar{\pi}) \leq \varepsilon_{\mathsf{opt}}^{\pi} + \mathcal{O}\left(\frac{V_{\max}\sqrt{C}(d_{\mathcal{F},\Pi})^{1/3}}{(1-\gamma)N^{1/3}}\right) + \frac{1}{K(1-\gamma)} \sum_{k=1}^{K} \langle d^{\pi} \setminus \nu,\ f_k - \mathcal{T}^{\pi_k} f_k \rangle,$$

where $(d^{\pi} \setminus \nu)(s,a) := \max(d^{\pi}(s,a) - \nu(s,a), 0)$, and $\langle d, f \rangle := \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} d(s,a) f(s,a)$ for any d and f.

At a high-level, our result shows that we can compete with any policy π using a sufficiently large dataset, as long as our optimization regret is small and the data distribution μ has a good coverage for d^{π}. In particular, choosing $\nu = d^{\pi}$ removes the off-support term, so that we always have a guarantee scaling with $\max_k \mathscr{C}(d^{\pi}, \mu, \mathcal{F}, \pi_k)$, but can benefit if other distributions ν are better covered with a small off-support mass $\|d^{\pi} \setminus \nu\|_1$. The off-support term can also be small if a small Bellman error under μ generalizes to a small error out of support, due to properties of $\mathcal{F}$.

Comparison with prior theoretical results To compare our result with prior works, we focus on the two statistical

error terms in our bound, ignoring the optimization regret. Relative to the information-theoretic bound of Xie et al. (2021), we observe a similar decomposition into a finite sample deviation term and an off-support bias. Their finite sample error decays as $N^{-1/2}$ as opposed to our $N^{-1/3}$ scaling, which arises from the use of regularization here. Indeed, we can get a $N^{-1/2}$ bound for a constrained version, but such a version is not friendly to practical implementation. Prior linear methods (Jin et al., 2021; Zanette et al., 2021; Uehara et al., 2021) have roughly similar guarantees to Xie et al. (2021), so a similar comparison holds.

Most related to Theorem 5 is the $N^{-1/5}$ bound of Xie et al. (2021, Corollary 5) for their regularized algorithm PSPI, which is supposed to be computationally tractable though no practical implementation is offered.[5] While our bound is better, we use a bounded complexity Π while their result uses an unrestricted policy class. If we were to use the same policy class as theirs, the complexity of Π would grow with optimization iterates, requiring us to carefully balance the regret and deviation terms and yielding identical guarantees to theirs. To summarize, our result is comparable to Xie et al. (2021, Corollary 5) and stated in a more general form, and we enjoy a crucial advantage of robust policy improvement as detailed below.

Robust policy improvement We now formalize the robust policy improvement of Algorithm 1, which can be viewed as the finite-sample version of Proposition 3.

Proposition 6. *Let $\bar{\pi}$ be the output of Algorithm 1. If Assumption 1 holds with $\varepsilon_{\mathcal{F}} = 0$ and $\mu \in \Pi$, with high probability,*

$$J(\mu) - J(\bar{\pi}) \leq \mathcal{O}\left(\frac{V_{\max}}{1-\gamma}\sqrt{\frac{d_{\mathcal{F},\Pi}}{N}} + \frac{\beta V_{\max}^2 d_{\mathcal{F},\Pi}}{(1-\gamma)N}\right) + \varepsilon_{\text{opt}}^{\mu}.$$

Proposition 6 provides the robust policy improvement guarantee in the finite-sample regime, under a *weaker* assumption on $\mathcal{F}$ than that in Theorem 5. In contrast to the regular safe policy improvement results in offline RL (e.g., Xie et al., 2021, Corollary 3) where the pessimistic hyperparamter is required to choose properly, the robust policy improvement from Proposition 6 could adapt to a wide range of β. As long as $\beta = o(N)$, the learned policy $\bar{\pi}$ from Algorithm 1 is guaranteed improve the behavior policy μ consistently. In fact, for such a range of β, robust policy improvement holds regardless of the quality of the learned critic. For example, when $\beta = 0$, Proposition 6 still guarantees a policy no worse than the behavior policy μ, though the critic loss does not contain the Bellman error term anymore. (In this case, ATAC performs IL). In contrast, prior works based on absolute pessimism (e.g., Xie et al., 2021) immediately output degenerate solutions when the Bellman error term is removed.

[5] Incidentally, we are able to use our empirical insights to provide a scalable implementation of PSPI; see Section 5.

Algorithm 2 ATAC (Practical Version)

Input: Batch data $\mathcal{D}$, policy π, critics f_1, f_2, constants $\beta \geq 0$, $\tau \in [0,1]$, $w \in [0,1]$

1: Initialize target networks $\bar{f}_1 \leftarrow f_1$, $\bar{f}_2 \leftarrow f_2$
2: **for** $k = 1, 2, \ldots, K$ **do**
3: Sample minibatch $\mathcal{D}_{\text{mini}}$ from dataset $\mathcal{D}$.
4: For $f \in \{f_1, f_2\}$, update critic networks
$l_{\text{critic}}(f) := \mathcal{L}_{\mathcal{D}_{\text{mini}}}(f, \pi) + \beta \mathcal{E}^{w}_{\mathcal{D}_{\text{mini}}}(f, \pi)$
$f \leftarrow \text{Proj}_{\mathcal{F}}(f - \eta_{\text{fast}} \nabla l_{\text{critic}})$
5: Update actor network
$l_{\text{actor}}(\pi) := -\mathcal{L}_{\mathcal{D}_{\text{mini}}}(f_1, \pi)$
$\pi \leftarrow \text{Proj}_{\Pi}(\pi - \eta_{\text{slow}} \nabla l_{\text{actor}})$
6: For $(f, \bar{f}) \in \{(f_i, \bar{f}_i)\}_{i=1,2}$, update target
$\bar{f} \leftarrow (1-\tau)\bar{f} + \tau f$.
7: **end for**

It is also notable that, compared with Theorem 5, Proposition 6 enjoys a better statistical rate with a proper β, i.e., $\beta \leq O(N^{1/2})$, due to the decomposition of performance difference shown in the following proof sketch.

Proof sketch Theorem 5 is established based on the following decomposition of performance difference: $\forall \pi$,

$$(1-\gamma)(J(\pi) - J(\pi_k)) \leq \mathbb{E}_{\mu}\left[f_k - \mathcal{T}^{\pi_k} f_k\right] - \mathbb{E}_{\pi}\left[f_k - \mathcal{T}^{\pi_k} f_k\right] + \mathbb{E}_{\pi}\left[f_k(s,\pi) - f_k(s,\pi_k)\right] + \widetilde{\mathcal{O}}\Big(\sqrt{\frac{V_{\max}^2}{N}} + \frac{\beta V_{\max}^2}{N}\Big). \quad (6)$$

Details of this decomposition can be found in Appendix B.2, and the proof relies on the fact that f_k is obtained by our pessimistic policy evaluation procedure. In Eq.(6), the first two terms are controlled by the Bellman error (both on-support and off-support), and the third is controlled by the optimization error. Notably, when the comparator π is the behavior policy μ, the first two terms in Eq.(6) cancel out, giving the faster rate of Proposition 6. This provides insight for why robust policy improvement does not depend on the quality of the learned critic.

4.2. A Practical Implementation of ATAC

We present a scalable deep RL version of ATAC in Algorithm 2, following the principles of Algorithm 1. With abuse of notation, we use ∇l_{actor}, ∇l_{critic} to denote taking gradients with respect to the parameters of the actor and the critic, respectively; similarly Line 6 in Algorithm 2 refers to a moving average in the parameter space. In addition, every term involving π in Algorithm 2 means a stochastic approximation based on sampling an action from π when queried. In implementation, we use adaptive gradient descent algorithm ADAM (Kingma & Ba, 2015) for updates in Algorithm 2 (i.e. $f - \eta_{\text{fast}} \nabla l_{\text{critic}}$ and $\pi - \eta_{\text{slow}} \nabla l_{\text{actor}}$).

Algorithm 2 is a two-timescale first-order algorithm (Borkar, 1997; Maei et al., 2009), where the critic is updated with a much faster rate η_{fast} than the actor with η_{slow}. This two-

timescale update is designed to mimic the oracle updates in Algorithm 1. Using $\eta_{\text{fast}} \gg \eta_{\text{slow}}$ allows us to approximately treat the critic in Algorithm 2 as the solution to the pessimistic policy evaluation step in Algorithm 1 for a given actor (Maei et al., 2009); on the other hand, the actor's gradient update rule is reminiscent of the incremental nature of no-regret optimization oracles.

4.2.1. Critic Update

The update in Line 4 of Algorithm 2 is a first-order approximation of Line 3 in Algorithm 1. We discuss the important design decisions of this practical critic update below.

Projection Each critic update performs a projected mini-batch gradient step, where the projection to $\mathcal{F}$ ensures bounded complexity for the critic. We parameterize $\mathcal{F}$ as neural networks with ℓ_2 *bounded* weights.[6] The projection is crucial to ensure stable learning across all β values. The use of projection can be traced back to the training Wasserstein GAN (Arjovsky et al., 2017) or IPM-based IL (Swamy et al., 2021). We found alternatives such as weight decay penalty to be less reliable.

Double Q residual algorithm loss Off-policy optimization with function approximators and bootstrapping faces the notorious issue of deadly triad (Sutton & Barto, 2018). Commonly this is mitigated through the use of double Q heuristic (Fujimoto et al., 2018; Haarnoja et al., 2018); however, we found that this technique alone is insufficient to enable numerically stable policy evaluation when the policy π takes very different actions[7] from the behavior policy μ. To this end, we design a new surrogate for the Bellman error $\mathcal{E}_D(f, \pi)$ for Algorithm 2, by combining the double Q heuristic and the objective of the Residual Algorithm (RA) (Baird, 1995), both of which are previous attempts to combat the deadly triad. Specifically, we design the surrogate loss as the convex combination of the temporal difference (TD) losses of the critic and its delayed targets:

$$\mathcal{E}^w_{\mathcal{D}}(f, \pi) := (1-w)\mathcal{E}^{\text{td}}_{\mathcal{D}}(f, f, \pi) + w\mathcal{E}^{\text{td}}_{\mathcal{D}}(f, \bar{f}_{\min}, \pi) \quad (7)$$

where $w \in [0, 1]$, $\mathcal{E}^{\text{td}}_{\mathcal{D}}(f, f', \pi) := \mathbb{E}_{\mathcal{D}}[(f(s,a) - r - \gamma f'(s', \pi))^2]$, and $\bar{f}_{\min}(s,a) := \min_{i=1,2} \bar{f}_i(s,a)$. We call the objective in Eq.(7), the *DQRA loss*. We found that using the DQRA loss significantly improves the optimization stability compared with just the double Q heuristic alone; see Figure 2. As a result, ATAC can perform stable optimization with higher β values and make the learner less pessimistic. This added stability of DQRA comes from that the residual error $\mathcal{E}^{\text{td}}_{\mathcal{D}}(f, f, \pi)$ is a fixed rather than a changing objective. This stabilization overcomes potential biases due to the challenges (related to double sampling) in unbiased gradient estimation of the RA objective. Similar observations were made by Wang & Ueda (2021) for online RL. In practice, we found that $w = 0.5$ works stably; using $w \approx 0$ ensures numerical stability, but has a worst-case exponentially slow convergence speed and often deteriorates neural network learning (Schoknecht & Merke, 2003; Wang & Ueda, 2021). In Section 5, we show an ablation to study the effects of w.

[6] We impose no constraint on the bias term.

[7] Divergence often happens, e.g., when π is uniform.

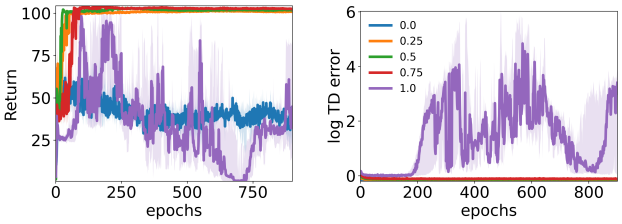


Figure 2. Ablation of the DQRA loss with different mixing weights w in Eq.(7). The plots show the policy performance and TD error across optimization epochs of ATAC with the hopper-medium-replay dataset. The stability and performance are greatly improved when $w \in (0, 1)$. For each w, the plot shows the 25^{th}, 50^{th}, 75^{th} percentiles over 10 random seeds.

4.2.2. Actor Update

The actor update aims to achieve no-regret with respect to the adversarially chosen critics. In Algorithm 2, we adopt a gradient based update (implemented as ADAM) mimicking the proximal nature of theoretical no-regret algorithms. Although ADAM has no formal no-regret guarantees for neural network learning, it works quite well in practice for RL and IL algorithms based on no-regret learning (Sun et al., 2017; Cheng et al., 2019; 2021).

Projection We set Π to be a class of policies with a minimal entropy constraint, so the projection in Line 5 ensures that the updated policy has a non-zero entropy. Soft policy iteration style theoretical algorithms naturally keep a reasonable entropy, and practically this avoids getting trapped in poor local optima. We implement the constraint by a Lagrange relaxation similar to SAC (Haarnoja et al., 2018).

Actor loss with a single critic While the critic optimization uses the double Q heuristic for numerical stability, the actor loss only uses one of the critics (we select f_1). This actor loss is similar to TD3 (Fujimoto et al., 2018), but different from SAC (Haarnoja et al., 2018) which takes $\min_{i=1,2} f_i(s,a)$ as the objective. This design choice is critical to enable ATAC's IL behavior when β is low. On the contrary, using the SAC-style loss produces instability for small β, with the actor loss oscillating in a limit cycle between the two critics.

5. Experiments

We test the effectiveness of ATAC (Algorithm 2) in terms of performance and robust policy improvement using the D4RL

	Behavior	ATAC*	ATAC	ATAC$_0^*$	ATAC$_0$	CQL	COMBO	TD3+BC	IQL	BC
halfcheetah-rand	-0.1	4.8	3.9	2.3	2.3	35.4	**38.8**	10.2	-	2.1
walker2d-rand	0.0	**8.0**	6.8	7.6	5.7	7.0	7.0	1.4	-	1.6
hopper-rand	1.2	**31.8**	17.5	31.6	18.2	10.8	17.9	11.0	-	9.8
halfcheetah-med	40.6	**54.3**	**53.3**	43.9	36.8	44.4	**54.2**	42.8	47.4	36.1
walker2d-med	62.0	**91.0**	**89.6**	**90.5**	**89.6**	74.5	75.5	79.7	78.3	6.6
hopper-med	44.2	**102.8**	85.6	**103.5**	94.8	86.6	94.9	**99.5**	66.3	29.0
halfcheetah-med-replay	27.1	49.5	48.0	49.2	47.2	46.2	**55.1**	43.3	44.2	38.4
walker2d-med-replay	14.8	**94.1**	92.5	**94.2**	89.8	32.6	56.0	25.2	73.9	11.3
hopper-med-replay	14.9	**102.8**	**102.5**	**102.7**	**102.1**	48.6	73.1	31.4	94.7	11.8
halfcheetah-med-exp	64.3	**95.5**	**94.8**	41.6	39.7	62.4	90.0	**97.9**	86.7	35.8
walker2d-med-exp	82.6	**116.3**	**114.2**	**114.5**	104.9	98.7	96.1	101.1	**109.6**	6.4
hopper-med-exp	64.7	**112.6**	**111.9**	83.0	46.5	**111.0**	**111.1**	**112.2**	91.5	**111.9**
pen-human	207.8	79.3	53.1	**106.1**	61.7	37.5	-	-	71.5	34.4
hammer-human	25.4	**6.7**	1.5	3.8	1.2	**4.4**	-	-	1.4	1.5
door-human	28.6	8.7	2.5	**12.2**	7.4	**9.9**	-	-	4.3	0.5
relocate-human	86.1	**0.3**	**0.1**	**0.5**	**0.1**	**0.2**	-	-	**0.1**	**0.0**
pen-cloned	107.7	73.9	43.7	**104.9**	68.9	39.2	-	-	37.3	56.9
hammer-cloned	8.1	2.3	1.1	**3.2**	0.4	2.1	-	-	2.1	0.8
door-cloned	12.1	**8.2**	3.7	6.0	0.0	0.4	-	-	1.6	-0.1
relocate-cloned	28.7	**0.8**	**0.2**	**0.3**	**0.0**	**-0.1**	-	-	**-0.2**	**-0.1**
pen-exp	105.7	**159.5**	136.2	**154.4**	97.7	107.0	-	-	-	85.1
hammer-exp	96.3	**128.4**	**126.9**	118.3	99.2	86.7	-	-	-	**125.6**
door-exp	100.5	**105.5**	**99.3**	**103.6**	48.3	**101.5**	-	-	-	34.9
relocate-exp	101.6	**106.5**	**99.4**	**104.0**	74.3	95.0	-	-	-	**101.3**

Table 1. Evaluation on the D4RL dataset. Algorithms with score within ϵ from the best on each domain are marked in bold, where $\epsilon = 0.1|J(\mu)|$. Baseline results are from the respective papers. For ATAC variants, we take the median score over 10 seeds.

offline RL benchmark's continuous control domains (Fu et al., 2020). More details are given in Appendix C.

Setup and hyperaparameter selection We compare ATAC (Algorithm 2) with recent offline RL algorithms CQL (Kumar et al., 2020), COMBO (Yu et al., 2021), TD3+BC (Fujimoto & Gu, 2021), IQL (Kostrikov et al., 2021), as well as the offline IL baseline, behavior cloning (BC). We also introduce an absolute pessimism version of ATAC (denoted ATAC$_0$), where we replace $\mathcal{L}_{\mathcal{D}_{\text{mini}}}(f, \pi)$ in l_{critic} of Algorithm 2 with $f(s_0, \pi)$. ATAC$_0$ can be viewed as a deep learning implementation of the theoretical algorithm PSPI from Xie et al. (2021) with the template of Algorithm 2.

In Algorithm 2, we use $\eta_{\text{fast}} = 0.0005$ and $\eta_{\text{slow}} = 10^{-3}\eta_{\text{fast}}$ based on an offline tuning heuristic, $\tau = 0.005$ from the work of Haarnoja et al. (2018), and $w = 0.5$, *across all domains*. We include an ablation for w later and further details of our setup are given in Appendix C. The regularization coefficient β is our only hyperparameter which varies across datasets, based on an online selection. Specifically, we run 100 epochs of BC for warm start; followed by 900 epochs of ATAC, where 1 epoch denotes 2K gradient updates. For each dataset, we report the median results over 10 random seeds. Since ATAC does not have guarantees on last-iterate convergence, we report also the results of both the last iterate (denoted as ATAC and ATAC$_0$) and the best checkpoint (denoted as ATAC* and ATAC$_0^*$) selected among 9 checkpoints (each was made every 100 epochs). The hyperparameter β is picked separately for ATAC, ATAC$_0$, ATAC* and ATAC$_0^*$.

Comparison with offline RL baselines Overall the experimental results in Table 1 show that ATAC and ATAC* outperform other model-free offline RL baselines consistently and model-based method COMBO mostly. Especially significant improvement is seen in *walker2d-medium, walker2d-medium-replay, hopper-medium-replay and pen-expert*, although the performance is worse than COMBO and CQL in the *halfhcheetah-rand*. It turns out that our fixed learning rate parameter does not result in sufficient convergence of ATAC on this domain. Our adaptation of PSPI (i.e. ATAC$_0$ and ATAC$_0^*$) is remarkably competitive with state-of-the-art baselines. This is the first empirical evaluation of PSPI, which further demonstrates the effectiveness of our design choices in Algorithm 2. However, ATAC$_0$ and ATAC$_0^*$ perform worse than ATAC and ATAC*, except for *pen-human, door-human, and pen-cloned*. In Appendix C we show ATAC and ATAC$_0$'s variability of performance across seeds by adding 25% and 75% quantiles of scores across 10 random seeds. (For baselines we only have scalar performance from the published results.)

Robust policy improvement We study whether the practical version of ATAC also enjoys robust policy improvement as Proposition 6 proves for the theoretical version. We show how ATAC* performs with various β values in Figure 1 on hopper. The results are consistent with the theoretical prediction in Proposition 6: ATAC robustly improves upon the behavior policy almost for all β except very large ones.

For large β, Proposition 6 shows that the finite-sample statistical error dominates the bound. ATAC does, however, not improve from the behavior policy on *-*human* and *-*cloned* even for well-tuned β; in fact, none of the offline RL algorithms does. We suspect that this is due to the failure of the realizability assumption $\mu \in \Pi$, as these datasets contain human demonstrations which can be non-Markovian. We include the variation of results across β for all datasets as well as statistics of robust policy improvement across β and iterates in Appendix C. This robust policy improvement property of ATAC means that practitioners can tune the performance of ATAC by starting with $\beta = 0$ and gradually increasing β until the performance drops, without ever deploying a policy significantly worse than the previous behavior policy.

Ablation of DQRA loss We show that the optimization stability from the DQRA loss is a key contributor to ATAC's performance by an ablation. We run ATAC with various w on *hopper-medium-replay*. When $w = 1$ (i.e. using conventional bootstrapping with double Q), the Bellman minimization part becomes unstable and the TD error $\mathcal{E}^{\text{td}}_{\mathcal{D}}(f, f, \pi)$ diverges. Using just the residual gradient ($w = 0$), while being numerical stable, leads to bad policy performance as also observed in the literature (Schoknecht & Merke, 2003; Wang & Ueda, 2021). For $w \in (0, 1)$, the stability and performance are usually significantly better than $w \in \{0, 1\}$. For simplicity, we use $w = 0.5$ in our experiments.

6. Discussion and Conclusion

We propose the concept of relative pessimism for offline RL and use it to design a new algorithm ATAC based on a Stackelberg game formulation. ATAC enjoys strong guarantees comparable to prior theoretical works, with an additional advantage of robust policy improvement due to relative pessimism. Empirical evaluation confirms the theoretical predictions and demonstrates ATAC's state-of-the-art performance on D4RL offline RL benchmarks.

ATAC shows a natural bridge between IL and offline RL. From its perspective, IL is an offline RL problem with the largest uncertainty on the value function (since IL does not have reward information), as captured by setting $\beta = 0$ in ATAC. In this case, the best policy under relative pessimism is to mimic the behavior policy exactly; otherwise, there is always a scenario within the uncertainty where the agent performs worse than the behavior policy. Only by considering the reduced uncertainty due to labeled rewards, it becomes possible for offline RL to learn a policy that strictly improves over the behavior policy. Conversely, we can view IL as the most pessimistic offline RL algorithm, which ignores the information in the data reward labels. Indeed IL does not make assumption on the data coverage, which is the core issue offline RL attempts to solve. We hope that this insightful connection can encourage future research on advancing IL and offline RL.

Finally, we remark on some limitations of ATAC. While ATAC has strong theoretical guarantees with general function approximators, it comes with a computational cost that its adversarial optimization problem (like that of Xie et al. (2021)) is potentially harder to solve than alternative offline RL approaches based on dynamic programming in a fixed pessimism MDP (Jin et al., 2021; Liu et al., 2020; Fujimoto & Gu, 2021; Kostrikov et al., 2021). For example, in our theoretical algorithm (Algorithm 1), we require having a no-regret policy optimization oracle (Definition 4), which we only know is provably time and memory efficient for linear function approximators and softmax policies (Xie et al., 2021).[8] This extra computational difficulty also manifests in the IL special case of ATAC (i.e. $\beta = 0$): ATAC reduces to IPM-minimization or Wasserstein-GAN for IL which requires harder optimization than BC based on maximum likelihood estimation, though the adversarial training version can produce a policy of higher quality. How to strike a better balance between the quality of the objective function and its computational characteristics is an open question.

Acknowledgment

NJ acknowledges funding support from ARL Cooperative Agreement W911NF-17-2-0196, NSF IIS-2112471, NSF CAREER IIS-2141781, and Adobe Data Science Research Award.

References

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Antos, A., Szepesvári, C., and Munos, R. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129, 2008.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Baird, L. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pp. 30–37. Elsevier, 1995.

Bertsekas, D. P. and Tsitsiklis, J. N. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th*

[8] When using nonlinear function approximators, the scheme there require memory linear in K.

IEEE conference on decision and control, volume 1, pp. 560–564. IEEE, 1995.

Borkar, V. S. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5):291–294, 1997.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, learning, and games*. Cambridge university press, 2006.

Chen, J. and Jiang, N. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pp. 1042–1051, 2019.

Cheng, C.-A., Yan, X., Ratliff, N., and Boots, B. Predictor-corrector policy optimization. In *International Conference on Machine Learning*, pp. 1151–1161. PMLR, 2019.

Cheng, C.-A., Kolobov, A., and Agarwal, A. Policy improvement via imitation of multiple oracles. *Advances in Neural Information Processing Systems*, 33, 2020.

Cheng, C.-A., Kolobov, A., and Swaminathan, A. Heuristic-guided reinforcement learning. *Advances in Neural Information Processing Systems*, 34:13550–13563, 2021.

Even-Dar, E., Kakade, S. M., and Mansour, Y. Online markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

Farahmand, A. M., Munos, R., and Szepesvári, C. Error propagation for approximate policy and value iteration. In *Advances in Neural Information Processing Systems*, 2010.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

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

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pp. 1587–1596. PMLR, 2018.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pp. 2052–2062, 2019.

Geist, M., Scherrer, B., and Pietquin, O. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pp. 2160–2169. PMLR, 2019.

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*, pp. 1861–1870. PMLR, 2018.

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

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pp. 267–274, 2002.

Kakade, S. M. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. Morel: Model-based offline reinforcement learning. In *NeurIPS*, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

Konda, V. R. and Tsitsiklis, J. N. Actor-critic algorithms. In *Advances in neural information processing systems*, pp. 1008–1014. Citeseer, 2000.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32:11784–11794, 2019.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191, 2020.

Laroche, R., Trichelair, P., and Des Combes, R. T. Safe policy improvement with baseline bootstrapping. In *International Conference on Machine Learning*, pp. 3652–3661. PMLR, 2019.

Liu, Y., Swaminathan, A., Agarwal, A., and Brunskill, E. Provably good batch reinforcement learning without great exploration. *Advances in Neural Information Processing Systems*, 33, 2020.

Maei, H. R., Szepesvari, C., Bhatnagar, S., Precup, D., Silver, D., and Sutton, R. S. Convergent temporal-difference learning with arbitrary smooth function approximation. In *NIPS*, pp. 1204–1212, 2009.

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.

Müller, A. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 1997.

Munos, R. Error bounds for approximate policy iteration. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, pp. 560–567, 2003.

Munos, R. and Szepesvári, C. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9 (5), 2008.

Neu, G., Jonsson, A., and Gómez, V. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

Paine, T. L., Paduraru, C., Michi, A., Gulcehre, C., Zolna, K., Novikov, A., Wang, Z., and de Freitas, N. Hyperparameter selection for offline reinforcement learning. *arXiv preprint arXiv:2007.09055*, 2020.

Rajeswaran, A., Mordatch, I., and Kumar, V. A game theoretic framework for model based reinforcement learning. In *International Conference on Machine Learning*, pp. 7953–7963. PMLR, 2020.

Schoknecht, R. and Merke, A. Td (0) converges provably faster than the residual gradient algorithm. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 680–687, 2003.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Sun, W., Venkatraman, A., Gordon, G. J., Boots, B., and Bagnell, J. A. Deeply aggrevated: Differentiable imitation learning for sequential prediction. In *International Conference on Machine Learning*, pp. 3309–3318. PMLR, 2017.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Swamy, G., Choudhury, S., Bagnell, J. A., and Wu, S. Of moments and matching: A game-theoretic framework for closing the imitation gap. In *International Conference on Machine Learning*, pp. 10022–10032. PMLR, 2021.

Uehara, M., Zhang, X., and Sun, W. Representation learning for online and offline rl in low-rank mdps. *arXiv preprint arXiv:2110.04652*, 2021.

Von Stackelberg, H. *Market structure and equilibrium*. Springer Science & Business Media, 2010.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Wang, Z. T. and Ueda, M. A convergent and efficient deep q network algorithm. *arXiv preprint arXiv:2106.15419*, 2021.

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

Xie, T. and Jiang, N. Q* approximation schemes for batch reinforcement learning: A theoretical comparison. In *Conference on Uncertainty in Artificial Intelligence*, pp. 550–559. PMLR, 2020.

Xie, T., Cheng, C.-A., Jiang, N., Mineiro, P., and Agarwal, A. Bellman-consistent pessimism for offline reinforcement learning. *Advances in neural information processing systems*, 34:6683–6694, 2021.

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

Yu, T., Kumar, A., Rafailov, R., Rajeswaran, A., Levine, S., and Finn, C. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

Zanette, A., Wainwright, M. J., and Brunskill, E. Provable benefits of actor-critic methods for offline reinforcement learning. *Advances in neural information processing systems*, 34, 2021.

Zhang, S. and Jiang, N. Towards hyperparameter-free policy selection for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Zheng, L., Fiez, T., Alumbaugh, Z., Chasnov, B., and Ratliff, L. J. Stackelberg actor-critic: Game-theoretic reinforcement learning algorithms. *arXiv preprint arXiv:2109.12286*, 2021.

A. Related Works

There is a rich literature on offline RL with function approximation when the data distribution μ is sufficiently rich to cover the state-action distribution d^π for any $\pi \in \Pi$ (Antos et al., 2008; Munos, 2003; Munos & Szepesvári, 2008; Farahmand et al., 2010; Chen & Jiang, 2019; Xie & Jiang, 2020). However, this is a prohibitive assumption in practice where the data distribution is typically constrained by the quality of available policies, safety considerations and existing system constraints which can lead it to have a significantly narrower coverage. Based on this observation, there has been a line of recent works in both the theoretical and empirical literature that systematically consider datasets with inadequate coverage.

The methods designed for learning without coverage broadly fall in one of two categories. Many works adopt the *behavior regularization* approach, where the learned policy is regularized to be close to the behavior policy in states where adequate data is not observed. On the theoretical side, some works (Laroche et al., 2019; Kumar et al., 2019; Fujimoto et al., 2018) provide *safe policy improvement* guarantees, meaning that the algorithms always do at least as well as the behavior policy, while improving upon it when possible. These and other works (Wu et al., 2019; Fujimoto & Gu, 2021) also demonstrate the benefits of this principle in comprehensive empirical evaluations.

A second class of methods follow the principle of *pessimism in the face of uncertainty*, and search for a policy with the best value under all possible scenarios consistent with the data. Some papers perform this reasoning in a model-based manner (Kidambi et al., 2020; Yu et al., 2020). In the model-free setting, Liu et al. (2020) define pessimism by truncating Bellman backups from states with limited support in the data and provide theoretical guarantees for the function approximation setting when the behavior distribution μ is known or can be easily estimated from samples, along with proof-of-concept experiments. The need to estimate μ has been subsequently removed by several recent works in both linear (Jin et al., 2021; Zanette et al., 2021) and non-linear (Xie et al., 2021; Uehara et al., 2021) settings.

Of these, the work of Xie et al. (2021) is the closest to this paper. Their approach optimizes a maximin objective where the maximization is over policies and minimization over all $f \in \mathcal{F}$ which are Bellman-consistent for that policy under the data distribution. Intuitively, this identifies an $\mathcal{F}$-induced lower bound for the value of each policy through the Bellman constraint and maximizes that lower bound. They also develop a regularized version more amenable to practical implementation, but provide no empirical validation of their approach. While the optimization of a pessimistic estimate of $J(\pi)$ results in a good policy with well-chosen hyperparameters, we argue that maximizing an alternative lower bound on the relative performance difference $J(\pi) - J(\mu)$ is nearly as good in terms of the absolute quality of the returned policy with well-chosen hyperparameters, but additionally improves upon the behavior policy for all possible choices of certain hyperparameters.

On the empirical side, several recent approaches (Kumar et al., 2020; Yu et al., 2021; Kostrikov et al., 2021) show promising empirical results for pessimistic methods. Many of these works consider policy iteration-style approaches where the policy class is implicitly defined in terms of a critic (e.g. through a softmax), whereas we allow explicit specification of both actor and critic classes. Somewhat related to our approach, the CQL algorithm (Kumar et al., 2020) trains a critic Q by maximizing the combination of a lower bound on $J(\pi_Q) - J(\mu)$, where π_Q is an implicit policy parameterized by Q, along with a Bellman error term for the current actor policy. The actor is trained with respect to the resulting critic. Lacking a clear objective like Eq.(1), this approach does not enjoy the robust policy improvement or other theoretical guarantees we establish in this paper. (We provide a detailed comparison with CQL in Appendix D) More generally, our experiments show that several elements of the theoretical design and practical implementation of our algorithm ATAC allow us to robustly outperform most of these baselines in a comprehensive evaluation.

B. Guarantees of Theoretical Algorithm

In this section, we provide the guarantees of theoretical algorithm including the the results provided in Section 4.1.

B.1. Concentration Analysis

This section provides the main results regarding $\mathcal{E}_\mathcal{D}(f, \pi)$ and its corresponding Bellman error. The results in this section are analogs of the results of Xie et al. (2021, Appendix A), but we use covering numbers to provide finer characteristics of the concentration. We provide the background of covering number as follows.

Definition 7 (ε-covering number)**.** An ε-cover of a set $\mathcal{F}$ with respect to a metric ρ is a set $\{g_1, \ldots, g_n\} \subseteq \mathcal{F}$, such that for each $g \in \mathcal{F}$, there exists some $g_i \in \{g_1, \ldots, g_n\}$ such that $\rho(g, g_i) \leq \varepsilon$. We define the ε-covering number of a set $\mathcal{F}$ under metric ρ, $\mathcal{N}(\mathcal{F}, \varepsilon, \rho)$, to be the the cardinality of the smallest ε-cover.

Further properties of covering number can be found in standard textbooks (see, e.g., Wainwright, 2019). In this paper, we will apply the ε-covering number on both function class $\mathcal{F}$ and policy class Π. For the function class, we use the following metric

$$\rho_{\mathcal{F}}(f_1, f_2) \coloneqq \|f_1 - f_2\|_\infty = \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} |f_1(s,a) - f_2(s,a)|. \tag{8}$$

We use $\mathcal{N}_\infty(\mathcal{F}, \varepsilon)$ to denote the ε-covering number of $\mathcal{F}$ w.r.t. metric $\rho_{\mathcal{F}}$ for simplicity.

Similarly, for the policy class, we define the metric as follows

$$\rho_\Pi(\pi_1, \pi_2) \coloneqq \|\pi_1 - \pi_2\|_{\infty,1} = \sup_{s\in\mathcal{S}} \|\pi_1(\cdot|s) - \pi_2(\cdot|s)\|_1, \tag{9}$$

and we use $\mathcal{N}_{\infty,1}(\Pi, \varepsilon)$ to denote the ε-covering number of Π w.r.t. metric ρ_Π for simplicity.

The following two theorems are the main results of this concentration analysis.

Theorem 8. *For any $\pi \in \Pi$, let f_π be defined as follows,*

$$f_\pi \coloneqq \operatorname*{argmin}_{f\in\mathcal{F}} \sup_{\text{admissible } \nu} \|f - \mathcal{T}^\pi f\|_{2,\nu}^2 .$$

Then, for $\mathcal{E}_\mathcal{D}(f_\pi, \pi)$ (defined in Eq.(5)), the following holds with probability at least $1-\delta$ for all $\pi \in \Pi$:

$$\mathcal{E}_\mathcal{D}(f_\pi, \pi) \le \mathcal{O}\left(\frac{V_{\max}^2 \log {|\mathcal{N}_\infty(\mathcal{F}, V_{\max}/N)||\mathcal{N}_{\infty,1}(\Pi, 1/N)|}/{\delta}}{N} + \varepsilon_\mathcal{F}\right) =: \varepsilon_r.$$

We now show that $\mathcal{E}_\mathcal{D}(f, \pi)$ could effectively estimate $\|f - \mathcal{T}^\pi f\|_{2,\mu}^2$.

Theorem 9. *With probability at least $1-\delta$, for any $\pi \in \Pi$, $f \in \mathcal{F}$,*

$$\|f - \mathcal{T}^\pi f\|_{2,\mu} - \sqrt{\mathcal{E}_\mathcal{D}(f,\pi)} \le \mathcal{O}\left(V_{\max}\sqrt{\frac{\log {|\mathcal{N}_\infty(\mathcal{F}, V_{\max}/N)||\mathcal{N}_{\infty,1}(\Pi, 1/N)|}/{\delta}}{N}} + \sqrt{\varepsilon_{\mathcal{F},\mathcal{F}}}\right). \tag{10}$$

When setting $\mathcal{E}_\mathcal{D}(f,\pi) = \varepsilon_r$, Eq.(10) implies a bound on $\|f - \mathcal{T}^\pi f\|_{2,\mu}$ which we denote as $\sqrt{\varepsilon_b}$ and will be useful later. That is,

$$\sqrt{\varepsilon_b} \coloneqq \sqrt{\varepsilon_r} + \mathcal{O}\left(V_{\max}\sqrt{\frac{\log {|\mathcal{N}_\infty(\mathcal{F}, V_{\max}/N)||\mathcal{N}_{\infty,1}(\Pi, 1/N)|}/{\delta}}{N}} + \sqrt{\varepsilon_{\mathcal{F},\mathcal{F}}}\right). \tag{11}$$

We first provide some complementary lemmas used for proving Theorems 8 and 9. The first lemma, Lemma 10, is the only place where we use concentration inequalities on $\mathcal{E}_\mathcal{D}$, and all high-probability statements regarding $\mathcal{E}_\mathcal{D}$ follow deterministically from Lemma 10.

Lemma 10. *With probability at least $1-\delta$, for any $f, g_1, g_2 \in \mathcal{F}$ and $\pi \in \Pi$,*

$$\begin{aligned}
&\Bigg| \|g_1 - \mathcal{T}^\pi f\|_{2,\mu}^2 - \|g_2 - \mathcal{T}^\pi f\|_{2,\mu}^2 \\
&- \frac{1}{N}\sum_{(s,a,r,s')\in\mathcal{D}} \left(g_1(s,a) - r - \gamma f(s',\pi)\right)^2 + \frac{1}{N}\sum_{(s,a,r,s')\in\mathcal{D}} \left(g_2(s,a) - r - \gamma f(s',\pi)\right)^2 \Bigg| \\
\le\ & \mathcal{O}\left(V_{\max}\|g_1 - g_2\|_{2,\mu}\sqrt{\frac{\log \frac{|\mathcal{N}_\infty(\mathcal{F}, \frac{V_{\max}}{N})||\mathcal{N}_{\infty,1}(\Pi, \frac{1}{N})|}{\delta}}{N}} + \frac{V_{\max}^2 \log \frac{|\mathcal{N}_\infty(\mathcal{F}, \frac{V_{\max}}{N})||\mathcal{N}_{\infty,1}(\Pi, \frac{1}{N})|}{\delta}}{N}\right).
\end{aligned}$$

Proof of Lemma 10. This proof follows a similar approach as the proof of Xie et al. (2021, Lemma A.4), but ours is established based on a more refined concentration analysis via covering number. We provide the full detailed proof here for completeness. By a standard calculation,

$$\begin{aligned}
&\frac{1}{N}\sum_{(s,a,r,s')\in\mathcal{D}} \left(g_1(s,a) - r - \gamma f(s',\pi)\right)^2 - \frac{1}{N}\sum_{(s,a,r,s')\in\mathcal{D}} \left(g_2(s,a) - r - \gamma f(s',\pi)\right)^2 \\
=\ & \frac{1}{N}\sum_{(s,a,r,s')\in\mathcal{D}} \Big(\left(g_1(s,a) - r - \gamma f(s',\pi)\right)^2 - \left(g_2(s,a) - r - \gamma f(s',\pi)\right)^2\Big)
\end{aligned}$$

$$= \frac{1}{N} \sum_{(s,a,r,s') \in \mathcal{D}} \left((g_1(s,a) - g_2(s,a)) \left(g_1(s,a) + g_2(s,a) - 2r - 2\gamma f(s', \pi) \right) \right). \tag{12}$$

Similarly, letting $\mu \times (\mathcal{P}, R)$ denote the distribution $(s,a) \sim \mu, r = R(s,a), s' \sim \mathcal{P}(\cdot|s,a)$, we have

$$\begin{aligned}
&\mathbb{E}_{\mu \times (\mathcal{P},R)} \left[\left(g_1(s,a) - r - \gamma f(s', \pi) \right)^2 \right] - \mathbb{E}_{\mu \times (\mathcal{P},R)} \left[\left(g_2(s,a) - r - \gamma f(s', \pi) \right)^2 \right] \\
&\overset{\text{(a)}}{=} \mathbb{E}_{\mu \times (\mathcal{P},R)} \left[(g_1(s,a) - g_2(s,a)) \left(g_1(s,a) + g_2(s,a) - 2r - 2\gamma f(s', \pi) \right) \right] \\
&= \mathbb{E}_{\mu} \left[\mathbb{E} \left[(g_1(s,a) - g_2(s,a)) \left(g_1(s,a) + g_2(s,a) - 2r - 2\gamma f(s', \pi) \right) | s, a \right] \right] \\
&= \mathbb{E}_{\mu} \left[(g_1(s,a) - g_2(s,a)) \left(g_1(s,a) + g_2(s,a) - 2 \left(\mathcal{T}^{\pi} f \right)(s,a) \right) \right] \qquad (13) \\
&\overset{\text{(b)}}{=} \mathbb{E}_{\mu} \left[\left(g_1(s,a) - \left(\mathcal{T}^{\pi} f \right)(s,a) \right)^2 \right] - \mathbb{E}_{\mu} \left[\left(g_2(s,a) - \left(\mathcal{T}^{\pi} f \right)(s,a) \right)^2 \right], \qquad (14)
\end{aligned}$$

where (a) and (b) follow from the similar argument to Eq.(12).

By using Eq.(12) and Eq.(14), we know

$$\begin{aligned}
&\mathbb{E}_{\mu \times (\mathcal{P},R)} \left[\frac{1}{N} \sum_{(s,a,r,s') \in \mathcal{D}} \left(g_1(s,a) - r - \gamma f(s', \pi) \right)^2 - \frac{1}{N} \sum_{(s,a,r,s') \in \mathcal{D}} \left(g_2(s,a) - r - \gamma f(s', \pi) \right)^2 \right] \\
&= \mathbb{E}_{\mu} \left[\left(g_1(s,a) - \left(\mathcal{T}^{\pi} f \right)(s,a) \right)^2 \right] - \mathbb{E}_{\mu} \left[\left(g_2(s,a) - \left(\mathcal{T}^{\pi} f \right)(s,a) \right)^2 \right].
\end{aligned}$$

Now, let $\mathcal{F}_{\varepsilon_1}$ be an ε_1-cover of $\mathcal{F}$ and Π_{ε_2} be an ε_2-cover of Π, so that we know: i) $|\mathcal{F}_{\varepsilon_1}| = \mathcal{N}_{\infty}(\mathcal{F}, \varepsilon_1)$, $|\Pi_{\varepsilon_2}| = \mathcal{N}_{\infty,1}(\Pi, \varepsilon_2)$; ii) there exist $\widetilde{f}, \widetilde{g}_1, \widetilde{g}_2 \in \mathcal{F}_{\varepsilon_1}$ and $\widetilde{\pi} \in \Pi_{\varepsilon_2}$, such that $\|f - \widetilde{f}\|_{\infty}, \|g_1 - \widetilde{g}_1\|_{\infty}, \|g_2 - \widetilde{g}_2\|_{\infty} \leq \varepsilon_1$ and $\|\pi - \widetilde{\pi}\|_{\infty,1} \leq \varepsilon_2$, where $\|\cdot\|_{\infty}$ and $\|\cdot\|_{\infty,1}$ are defined in Eq.(8) and Eq.(9).

Then, with probability at least $1 - \delta$, for all $f, g_1, g_2 \in \mathcal{F}$, $\pi \in \Pi$, and the corresponding $\widetilde{f}, \widetilde{g}_1, \widetilde{g}_2, \widetilde{\pi}$,

$$\begin{aligned}
&\left| \mathbb{E}_{\mu} \left[\left(\widetilde{g}_1(s,a) - \left(\mathcal{T}^{\widetilde{\pi}} \widetilde{f} \right)(s,a) \right)^2 \right] - \mathbb{E}_{\mu} \left[\left(\widetilde{g}_2(s,a) - \left(\mathcal{T}^{\widetilde{\pi}} \widetilde{f} \right)(s,a) \right)^2 \right] \right. \\
&\left. - \frac{1}{N} \sum_{(s,a,r,s') \in \mathcal{D}} \left(\widetilde{g}_1(s,a) - r - \gamma \widetilde{f}(s', \widetilde{\pi}) \right)^2 + \frac{1}{N} \sum_{(s,a,r,s') \in \mathcal{D}} \left(\widetilde{g}_2(s,a) - r - \gamma \widetilde{f}(s', \widetilde{\pi}) \right)^2 \right| \\
&= \left| \mathbb{E}_{\mu} \left[\left(\widetilde{g}_1(s,a) - \left(\mathcal{T}^{\widetilde{\pi}} \widetilde{f} \right)(s,a) \right)^2 \right] - \mathbb{E}_{\mu} \left[\left(\widetilde{g}_2(s,a) - \left(\mathcal{T}^{\widetilde{\pi}} \widetilde{f} \right)(s,a) \right)^2 \right] \right. \\
&\left. - \frac{1}{N} \sum_{(s,a,r,s') \in \mathcal{D}} \left((\widetilde{g}_1(s,a) - \widetilde{g}_2(s,a)) \left(\widetilde{g}_1(s,a) + \widetilde{g}_2(s,a) - 2r - 2\gamma \widetilde{f}(s', \widetilde{\pi}) \right) \right) \right| \\
&\leq \sqrt{\frac{4 \mathbb{V}_{\mu \times (\mathcal{P},R)} \left[(\widetilde{g}_1(s,a) - \widetilde{g}_2(s,a)) \left(\widetilde{g}_1(s,a) + \widetilde{g}_2(s,a) - 2r - 2\gamma \widetilde{f}(s', \widetilde{\pi}) \right) \right] \log \frac{|\mathcal{N}_{\infty}(\mathcal{F}, \varepsilon_1)| |\mathcal{N}_{\infty,1}(\Pi, \varepsilon_2)|}{\delta}}{N}} \\
&+ \frac{2 V_{\max}^2 \log \frac{|\mathcal{N}_{\infty}(\mathcal{F}, \varepsilon_1)| |\mathcal{N}_{\infty,1}(\Pi, \varepsilon_2)|}{\delta}}{3N},
\end{aligned}$$

where the first equation follows from Eq.(13) and the last inequality follows from the Bernstein's inequality and union bounding over $\mathcal{F}_{\varepsilon_1}$ and Π_{ε_2}.

We now upper bound the variance term inside the squareroot of the above expression:

$$\begin{aligned}
&\mathbb{V}_{\mu \times (\mathcal{P},R)} \left[(\widetilde{g}_1(s,a) - \widetilde{g}_2(s,a)) \left(\widetilde{g}_1(s,a) + \widetilde{g}_2(s,a) - 2r - 2\gamma \widetilde{f}(s', \widetilde{\pi}) \right) \right] \\
&\leq \mathbb{E}_{\mu \times (\mathcal{P},R)} \left[(\widetilde{g}_1(s,a) - \widetilde{g}_2(s,a))^2 \left(\widetilde{g}_1(s,a) + \widetilde{g}_2(s,a) - 2r - 2\gamma \widetilde{f}(s', \widetilde{\pi}) \right)^2 \right] \\
&\leq 4 V_{\max}^2 \mathbb{E}_{\mu} \left[(\widetilde{g}_1(s,a) - \widetilde{g}_2(s,a))^2 \right].
\end{aligned}$$

where the last inequality follows from the fact of $|\widetilde{g}_1(s,a) + \widetilde{g}_2(s,a) - 2r - 2\gamma \widetilde{f}(s', \widetilde{\pi})| \leq 2V_{\max}$. Therefore, w.p. $1 - \delta$,

$$\left| \left\| \widetilde{g}_1 - \mathcal{T}^{\widetilde{\pi}} \widetilde{f} \right\|_{2,\mu}^2 - \left\| \widetilde{g}_2 - \mathcal{T}^{\widetilde{\pi}} \widetilde{f} \right\|_{2,\mu}^2 \right.$$

$$- \frac{1}{N} \sum_{(s,a,r,s') \in \mathcal{D}} \left(\widetilde{g}_1(s,a) - r - \gamma \widetilde{f}(s', \widetilde{\pi}) \right)^2 + \frac{1}{N} \sum_{(s,a,r,s') \in \mathcal{D}} \left(\widetilde{g}_2(s,a) - r - \gamma \widetilde{f}(s', \widetilde{\pi}) \right)^2 \Bigg|$$

$$\leq 4V_{\max} \|\widetilde{g}_1 - \widetilde{g}_2\|_{2,\mu} \sqrt{\frac{\log \frac{|\mathcal{N}_\infty(\mathcal{F}, \varepsilon_1)||\mathcal{N}_{\infty,1}(\Pi, \varepsilon_2)|}{\delta}}{N}} + \frac{2V_{\max}^2 \log \frac{|\mathcal{N}_\infty(\mathcal{F}, \varepsilon_1)||\mathcal{N}_{\infty,1}(\Pi, \varepsilon_2)|}{\delta}}{3N}.$$

By definitions of $\widetilde{f}, \widetilde{g}_1, \widetilde{g}_2$ and $\widetilde{\pi}$, we know for any (s, a, r, s') tuple,

$$\Big| \left(g_1(s,a) - r - \gamma f(s', \pi) \right)^2 + \left(g_2(s,a) - r - \gamma f(s', \pi) \right)^2$$

$$- \left(\widetilde{g}_1(s,a) - r - \gamma \widetilde{f}(s', \widetilde{\pi}) \right)^2 + \left(\widetilde{g}_2(s,a) - r - \gamma \widetilde{f}(s', \widetilde{\pi}) \right)^2 \Big| = \mathcal{O}(V_{\max} \varepsilon_1 + V_{\max}^2 \varepsilon_2),$$

and

$$\begin{aligned} \|g_1 - g_2\|_{2,\mu} &= \|\widetilde{g}_1 - \widetilde{g}_2 + (g_1 - \widetilde{g}_1) - (g_2 - \widetilde{g}_2)\|_{2,\mu} \\ &\leq \|\widetilde{g}_1 - \widetilde{g}_2\|_{2,\mu} + \|g_1 - \widetilde{g}_1\|_{2,\mu} + \|g_2 - \widetilde{g}_2\|_{2,\mu} \\ &\leq \|\widetilde{g}_1 - \widetilde{g}_2\|_{2,\mu} + 2\varepsilon_1. \end{aligned}$$

These implies

$$\Big| \|g_1 - \mathcal{T}^\pi f\|_{2,\mu}^2 - \|g_2 - \mathcal{T}^\pi f\|_{2,\mu}^2$$

$$- \frac{1}{N} \sum_{(s,a,r,s') \in \mathcal{D}} \left(g_1(s,a) - r - \gamma f(s', \pi) \right)^2 + \frac{1}{N} \sum_{(s,a,r,s') \in \mathcal{D}} \left(g_2(s,a) - r - \gamma f(s', \pi) \right)^2 \Big|$$

$$\lesssim V_{\max} \|g_1 - g_2\|_{2,\mu} \sqrt{\frac{\log \frac{|\mathcal{N}_\infty(\mathcal{F}, \varepsilon_1)||\mathcal{N}_{\infty,1}(\Pi, \varepsilon_2)|}{\delta}}{N}} + \frac{V_{\max}^2 \log \frac{|\mathcal{N}_\infty(\mathcal{F}, \varepsilon_1)||\mathcal{N}_{\infty,1}(\Pi, \varepsilon_2)|}{\delta}}{N}$$

$$+ V_{\max} \varepsilon_1 \sqrt{\frac{\log \frac{|\mathcal{N}_\infty(\mathcal{F}, \varepsilon_1)||\mathcal{N}_{\infty,1}(\Pi, \varepsilon_2)|}{\delta}}{N}} + V_{\max} \varepsilon_1 + V_{\max}^2 \varepsilon_2.$$

($x \lesssim y$ means $x \leq C \cdot y$ for some absolute constant C)

Choosing $\varepsilon_1 = \mathcal{O}(\frac{V_{\max}}{N})$ and $\varepsilon_2 = \mathcal{O}(\frac{1}{N})$ completes the proof. □

Lemma 11. *For any $\pi \in \Pi$, let f_π and g be defined as follows,*

$$f_\pi \coloneqq \operatorname*{argmin}_{f \in \mathcal{F}} \sup_{\text{admissible } \nu} \|f - \mathcal{T}^\pi f\|_{2,\nu}^2$$

$$g \coloneqq \operatorname*{argmin}_{g' \in \mathcal{F}} \frac{1}{N} \sum_{(s,a,r,s') \in \mathcal{D}} \left(g'(s,a) - r - \gamma f_\pi(s', \pi) \right)^2.$$

Then, with high probability,

$$\|f_\pi - g\|_{2,\mu} \leq \mathcal{O}\left(V_{\max} \sqrt{\frac{\log \frac{|\mathcal{N}_\infty(\mathcal{F}, \frac{V_{\max}}{N})||\mathcal{N}_{\infty,1}(\Pi, \frac{1}{N})|}{\delta}}{N}} + \sqrt{\varepsilon_{\mathcal{F}}} \right).$$

Proof of Lemma 11. The proof of this lemma is obtained exactly the same as Xie et al. (2021, Proof of Lemma A.5), we we only need to change the use of Xie et al. (2021, Lemma A.4) to Lemma 10. This completes the proof. □

We now ready to prove Theorem 8 and Theorem 9. Note that the proofs of Theorem 8 and Theorem 9 follow similar approaches as the proof of Xie et al. (2021, Theorem A.1, Theorem A.2), and we provide the full detailed proof here for completeness.

Proof of Theorem 8. This proof is obtained by exactly the same strategy of Xie et al. (2021, Proof of Theorem A.1), but we we change to change the corresponding lemmas to the new ones provided above. The correspondence of those lemmas are as follows: (Xie et al., 2021, Lemma A.4) → Lemma 10; (Xie et al., 2021, Lemma A.5) → Lemma 11. This completes the proof. □

Proof of Theorem 9. This proof is obtained by the same exactly same strategy of Xie et al. (2021, Proof of Theorem A.2), but we we change to change the corresponding lemmas to the new ones provided above. The correspondence of those lemmas are as follows: (Xie et al., 2021, Lemma A.4) $\rightarrow$ Lemma 10; (Xie et al., 2021, Lemma A.5) $\rightarrow$ Lemma 11. This completes the proof. □

B.2. Decomposition of Performance Difference

This section proves Eq.(6). We provide a more general version of Eq.(6) with its proof as follows.

Lemma 12. *Let π be an arbitrary competitor policy, $\widehat{\pi} \in \Pi$ be some learned policy, and f be an arbitrary function over $\mathcal{S} \times \mathcal{A}$. Then we have,*

$$J(\pi) - J(\widehat{\pi}) = \frac{1}{1-\gamma}\left(\mathbb{E}_\mu\left[\left(f - \mathcal{T}^{\widehat{\pi}} f\right)(s,a)\right] + \mathbb{E}_\pi\left[\left(\mathcal{T}^{\widehat{\pi}} f - f\right)(s,a)\right] + \mathbb{E}_\pi\left[f(s,\pi) - f(s,\widehat{\pi})\right] + \mathcal{L}_\mu(\widehat{\pi}, f) - \mathcal{L}_\mu(\widehat{\pi}, Q^{\widehat{\pi}})\right).$$

Proof of Lemma 12. Let $R^{f,\widehat{\pi}}(s,a) \coloneqq f(s,a) - \gamma \mathbb{E}_{s'|s,a}[f(s',\widehat{\pi})]$ be a fake reward function given f and $\widehat{\pi}$. We use the subscript "$(\cdot)_{R^{f,\widehat{\pi}}}$" to denote functions or operators under the true dynamics but the fake reward $R^{f,\widehat{\pi}}$. Since $f(s,a) = (\mathcal{T}^\pi_{R^{f,\widehat{\pi}}} f)(s,a)$, we know $f \equiv Q^\pi_{R^{f,\widehat{\pi}}}$.

We perform a performance decomposition:

$$J(\pi) - J(\widehat{\pi}) = \left(J(\pi) - J(\mu)\right) - \left(J(\widehat{\pi}) - J(\mu)\right)$$

and rewrite the second term as

$$\begin{aligned}
(1-\gamma)\left(J(\widehat{\pi}) - J(\mu)\right) &= \mathcal{L}_\mu(\widehat{\pi}, Q^{\widehat{\pi}}) \\
&= \Delta(\widehat{\pi}) + \mathcal{L}_\mu(\widehat{\pi}, f) && (\Delta(\widehat{\pi}) \coloneqq \mathcal{L}_\mu(\widehat{\pi}, Q^{\widehat{\pi}}) - \mathcal{L}_\mu(\widehat{\pi}, f)) \\
&= \Delta(\widehat{\pi}) + \mathbb{E}_\mu[f(s,\widehat{\pi}) - f(s,a)] \\
&= \Delta(\widehat{\pi}) + (1-\gamma)(J_{R^{f,\widehat{\pi}}}(\widehat{\pi}) - J_{R^{f,\widehat{\pi}}}(\mu)) \\
&\qquad \text{(by performance difference lemma (Kakade \& Langford, 2002))} \\
&= \Delta(\widehat{\pi}) + (1-\gamma) Q^{\widehat{\pi}}_{R^{f,\widehat{\pi}}}(s_0, \widehat{\pi}) - \mathbb{E}_\mu[R^{\widehat{\pi},f}(s,a)] \\
&= \Delta(\widehat{\pi}) + (1-\gamma) f(s_0, \widehat{\pi}) - \mathbb{E}_\mu[R^{\widehat{\pi},f}(s,a)]. && (\text{by } f(\cdot,\cdot) \equiv Q^\pi_{R^{f,\widehat{\pi}}}(\cdot,\cdot))
\end{aligned}$$

Therefore,

$$(1-\gamma)(J(\pi) - J(\widehat{\pi})) = \underbrace{(1-\gamma)\left(J(\pi) - f(d_0, \widehat{\pi})\right)}_{\text{(I)}} + \underbrace{\left(\mathbb{E}_\mu[R^{\widehat{\pi},f}(s,a)] - (1-\gamma)J(\mu)\right)}_{\text{(II)}} - \Delta(\widehat{\pi}).$$

We first analyze (II). We can expand it by the definition of $R^{\widehat{\pi},f}$ as follows

$$\begin{aligned}
\text{(II)} &= \mathbb{E}_\mu[R^{\widehat{\pi},f}(s,a)] - (1-\gamma)J(\mu) \\
&= \mathbb{E}_\mu[R^{\widehat{\pi},f}(s,a) - R(s,a)] \\
&= \mathbb{E}_\mu[(f - \mathcal{T}^{\widehat{\pi}} f)(s,a)].
\end{aligned}$$

We now write (I) as

$$\begin{aligned}
\text{(I)} &= (1-\gamma)\left(J(\pi) - f(s_0, \widehat{\pi})\right) \\
&= \underbrace{(1-\gamma)J(\pi) - \mathbb{E}_{d^\pi}[R^{\widehat{\pi},f}(s,a)]}_{\text{(Ia)}} + \underbrace{\mathbb{E}_{d^\pi}[R^{\widehat{\pi},f}(s,a)] - (1-\gamma) f(s_0, \widehat{\pi})}_{\text{(Ib)}}.
\end{aligned}$$

We analyze each term above in the following.

$$\begin{aligned}
\text{(Ib)} &= \mathbb{E}_{d^\pi}[R^{\widehat{\pi},f}(s,a)] - (1-\gamma) f(s_0, \widehat{\pi}) \\
&= \mathbb{E}_{d^\pi}[f(s,\pi) - f(s,\widehat{\pi})].
\end{aligned}$$

On the other hand, we can write

$$\text{(Ia)} = (1-\gamma)J(\pi) - \mathbb{E}_{d^\pi}[R^{\widehat{\pi},f}(s,a)]$$

$$= \mathbb{E}_{d^\pi}[R(s,a) - R^{\widehat{\pi},f}(s,a)]$$
$$= \mathbb{E}_{d^\pi}[(\mathcal{T}^{\widehat{\pi}} f - f)(s,a)].$$

Combine them all, we have

$$\begin{aligned}
&J(\pi) - J(\widehat{\pi}) \\
&= \frac{1}{1-\gamma}\left((\text{Ia}) + (\text{Ib}) + (\text{II}) - \Delta(\widehat{\pi})\right) \\
&= \frac{1}{1-\gamma}\left(\mathbb{E}_\mu\left[\left(f - \mathcal{T}^{\widehat{\pi}} f\right)(s,a)\right] + \mathbb{E}_\pi\left[\left(\mathcal{T}^{\widehat{\pi}} f - f\right)(s,a)\right] + \mathbb{E}_\pi\left[f(s,\pi) - f(s,\widehat{\pi})\right] + \mathcal{L}_\mu(\widehat{\pi}, f) - \mathcal{L}_\mu(\widehat{\pi}, Q^{\widehat{\pi}})\right).
\end{aligned}$$

This completes the proof. □

We now prove a general version of Eq.(6) using Lemma 12, which takes into account the approximation errors in the realizability and completeness assumptions (Assumption 1 and Assumption 2).

Lemma 13 (General Version of Eq.(6))**.** *Let π be an arbitrary competitor policy. Also let π_k and f_k be obtained by Algorithm 1 for $k \in [K]$. Then with high probability, for any $k \in [K]$,*

$$\begin{aligned}
(1-\gamma)\left(J(\pi) - J(\pi_k)\right) \le{}& \mathbb{E}_\mu\left[f_k - \mathcal{T}^{\pi_k} f_k\right] + \mathbb{E}_\pi\left[\mathcal{T}^{\pi_k} f_k - f_k\right] + \mathbb{E}_\pi\left[f_k(s,\pi) - f_k(s,\pi_k)\right] \\
&+ \mathcal{O}\left(V_{\max}\sqrt{\frac{\log |\mathcal{N}_\infty(\mathcal{F}, V_{\max}/N)||\mathcal{N}_{\infty,1}(\Pi, 1/N)|/\delta}{N}} + \sqrt{\varepsilon_{\mathcal{F}}}\right) + \beta \cdot \mathcal{O}\left(\frac{V_{\max}^2 \log |\mathcal{N}_\infty(\mathcal{F}, V_{\max}/N)||\mathcal{N}_{\infty,1}(\Pi, 1/N)|/\delta}{N} + \varepsilon_{\mathcal{F}}\right).
\end{aligned}$$

Proof of Lemma 13. By Lemma 12, we have

$$J(\pi) - J(\pi_k) = \frac{\mathbb{E}_\mu\left[f_k - \mathcal{T}^{\pi_k} f_k\right]}{1-\gamma} + \frac{\mathbb{E}_\pi\left[\mathcal{T}^{\pi_k} f_k - f_k\right]}{1-\gamma} + \frac{\mathbb{E}_\pi\left[f_k(s,\pi) - f_k(s,\pi_k)\right]}{1-\gamma} + \frac{\mathcal{L}_\mu(\pi_k, f_k) - \mathcal{L}_\mu(\pi_k, Q^{\pi_k})}{1-\gamma}.$$

We now bound the term of $\mathcal{L}_\mu(\pi_k, f_k) - \mathcal{L}_\mu(\pi_k, Q^{\pi_k})$.

$$\begin{aligned}
f_\pi &\coloneqq \operatorname*{argmin}_{f \in \mathcal{F}} \sup_{\text{admissible } \nu} \|f - \mathcal{T}^\pi f\|_{2,\nu}^2, \ \forall \pi \in \Pi \\
\varepsilon_{\mathsf{stat}} &\coloneqq \mathcal{O}\left(\frac{V_{\max}^2 \log |\mathcal{N}_\infty(\mathcal{F}, V_{\max}/N)||\mathcal{N}_{\infty,1}(\Pi, 1/N)|/\delta}{N}\right), \\
\varepsilon_r &\coloneqq \varepsilon_{\mathsf{stat}} + \mathcal{O}\left(\varepsilon_{\mathcal{F}}\right).
\end{aligned}$$

Then, by Theorem 8, we know that with high probability, for any $k \in [K]$,

$$\mathcal{E}_{\mathcal{D}}(\pi_k, f_{\pi_k}) \le \varepsilon_r. \tag{15}$$

For $|\mathcal{L}_\mu(\pi_k, Q^{\pi_k}) - \mathcal{L}_\mu(\pi_k, f_{\pi_k})|$, we have,

$$\begin{aligned}
\mathcal{L}_\mu(\pi_k, Q^{\pi_k}) &= \mathbb{E}_\mu\left[Q^{\pi_k}(s,\pi_k) - Q^{\pi_k}(s,a)\right] \\
&= (1-\gamma)\left(J(\pi_k) - J(\mu)\right) \\
&= (1-\gamma)\left(f_{\pi_k}(s_0,\pi_k) - J(\mu)\right) + (1-\gamma)\left(J(\pi_k) - f_{\pi_k}(s_0,\pi_k)\right) \\
&= \mathbb{E}_\mu\left[f_{\pi_k}(s,\pi_k) - (\mathcal{T}^{\pi_k} f_{\pi_k})(s,a)\right] + \mathbb{E}_{d^{\pi_k}}\left[(\mathcal{T}^{\pi_k} f_{\pi_k})(s,a) - f_{\pi_k}(s,a)\right] \\
&\qquad \text{(by the extension of performance difference lemma (see, e.g., Cheng et al., 2020, Lemma 1))} \\
&= \mathcal{L}_\mu(\pi_k, f_{\pi_k}) + \mathbb{E}_\mu\left[f_{\pi_k}(s,a) - (\mathcal{T}^{\pi_k} f_{\pi_k})(s,a)\right] + \mathbb{E}_{d^{\pi_k}}\left[(\mathcal{T}^{\pi_k} f_{\pi_k})(s,a) - f_{\pi_k}(s,a)\right] \\
\implies |\mathcal{L}_\mu(\pi_k, Q^{\pi_k}) - \mathcal{L}_\mu(\pi_k, f_{\pi_k})| &\le \|f_{\pi_k} - \mathcal{T}^{\pi_k} f_{\pi_k}\|_{2,\mu} + \|\mathcal{T}^{\pi_k} f_{\pi_k} - f_{\pi_k}\|_{2,d^{\pi_k}} \\
&\le \mathcal{O}(\sqrt{\varepsilon_{\mathcal{F}}}),
\end{aligned} \tag{16}$$

where the last step is by Assumption 1. Also, by applying standard concentration inequalities on $\mathcal{L}_{\mathcal{D}}$ (the failure probability will be split evenly with that on $\mathcal{E}_{\mathcal{D}}$ from Lemma 10):

$$|\mathcal{L}_\mu(\pi_k, f_k) - \mathcal{L}_{\mathcal{D}}(\pi_k, f_k)| + |\mathcal{L}_\mu(\pi_k, f_{\pi_k}) - \mathcal{L}_{\mathcal{D}}(\pi_k, f_{\pi_k})| \le \sqrt{\varepsilon_{\mathsf{stat}}}, \ \forall k \in [K]. \tag{17}$$

Therefore,

$$\begin{aligned}
&\mathcal{L}_\mu(\pi_k, f_k) - \mathcal{L}_\mu(\pi_k, Q^{\pi_k}) \\
&\le \mathcal{L}_\mu(\pi_k, f_k) + \beta\mathcal{E}_{\mathcal{D}}(\pi_k, f_k) - \mathcal{L}_\mu(\pi_k, Q^{\pi_k}) && (\mathcal{E}_{\mathcal{D}}(\cdot) \ge 0) \\
&\le \mathcal{L}_\mu(\pi_k, f_k) + \beta\mathcal{E}_{\mathcal{D}}(\pi_k, f_k) - \mathcal{L}_\mu(\pi_k, f_{\pi_k}) - \beta\mathcal{E}_{\mathcal{D}}(\pi_k, f_{\pi_k}) + \mathcal{O}(\sqrt{\epsilon_{\mathcal{F}}}) + \beta\varepsilon_r && \text{(by Eq.(15) and Eq.(16))}
\end{aligned}$$

$$
\begin{aligned}
&\leq \mathcal{L}_{\mathcal{D}}(\pi_k, f_k) + \beta \mathcal{E}_{\mathcal{D}}(\pi_k, f_k) - \mathcal{L}_{\mathcal{D}}(\pi_k, f_{\pi_k}) - \beta \mathcal{E}_{\mathcal{D}}(\pi_k, f_{\pi_k}) \\
&\quad + \mathcal{O}(\sqrt{\epsilon_{\mathcal{F}}}) + \sqrt{\varepsilon_{\mathsf{stat}}} + \beta \cdot \mathcal{O}\left(\varepsilon_{\mathsf{stat}} + \varepsilon_{\mathcal{F}}\right) && \text{(by Eq.(17))} \\
&\leq \mathcal{O}\left(\sqrt{\varepsilon_{\mathcal{F}}}\right) + \sqrt{\varepsilon_{\mathsf{stat}}} + \beta \cdot \mathcal{O}\left(\varepsilon_{\mathsf{stat}} + \varepsilon_{\mathcal{F}}\right) && \text{(by the optimality of } f_k\text{)} \\
&\leq \mathcal{O}\left(\sqrt{\varepsilon_{\mathcal{F}}} + V_{\max}\sqrt{\frac{\log |\mathcal{N}_\infty(\mathcal{F}, V_{\max}/N)||\mathcal{N}_{\infty,1}(\Pi, 1/N)|/\delta}{N}}\right) + \beta \cdot \mathcal{O}\left(\frac{V_{\max}^2 \log |\mathcal{N}_\infty(\mathcal{F}, V_{\max}/N)||\mathcal{N}_{\infty,1}(\Pi, 1/N)|/\delta}{N} + \varepsilon_{\mathcal{F}}\right).
\end{aligned}
$$

This completes the proof. □

B.3. Performance Guarantee of the Theoretical Algorithm

This section proves a general version of Theorem 5 using Lemma 12, which relies on the approximate realizability and completeness assumptions (Assumption 1 and Assumption 2).

Theorem 14 (General Version of Theorem 5)**.** *Under the same condition as Theorem 5, let $C > 0$ be any constant, ν be an arbitrarily distribution that satisfies $\mathscr{C}(\nu; \mu, \mathcal{F}, \pi_k) \leq C$, $\varepsilon_{\mathsf{stat}} \coloneqq \mathcal{O}\left(\frac{V_{\max}^2 \log |\mathcal{N}_\infty(\mathcal{F}, V_{\max}/N)||\mathcal{N}_{\infty,1}(\Pi, 1/N)|/\delta}{N}\right)$, and π be an arbitrary competitor policy. Then, we choose $\beta = \mathcal{O}\left(\frac{V_{\max}^{1/3}}{(\varepsilon_{\mathcal{F}} + \varepsilon_{\mathsf{stat}})^{2/3}}\right)$ and with probability at least $1 - \delta$,*

$$
\begin{aligned}
&J(\pi) - J(\bar{\pi}) \\
&\leq \mathcal{O}\left(\frac{\sqrt{C}\left(\sqrt{\varepsilon_{\mathcal{F}}} + \sqrt{\varepsilon_{\mathcal{F},\mathcal{F}}} + \sqrt{\varepsilon_{\mathsf{stat}}} + \left(V_{\max}\varepsilon_{\mathcal{F}} + V_{\max}\varepsilon_{\mathsf{stat}}\right)^{1/3}\right)}{1-\gamma}\right) + \frac{\langle d^\pi \setminus \nu,\ f_k - \mathcal{T}^{\pi_k} f_k \rangle}{1-\gamma}.
\end{aligned}
$$

Proof of Theorem 14. Over this proof, let

$$
\varepsilon_{\mathsf{stat}} \coloneqq \mathcal{O}\left(\frac{V_{\max}^2 \log |\mathcal{N}_\infty(\mathcal{F}, V_{\max}/N)||\mathcal{N}_{\infty,1}(\Pi, 1/N)|/\delta}{N}\right).
$$

By the definition of $\bar{\pi}$, we have

$$
\begin{aligned}
&J(\pi) - J(\bar{\pi}) \\
&= \frac{1}{K}\sum_{k=1}^{K}\left(J(\pi) - J(\pi_k)\right) \\
&\leq \frac{1}{K}\sum_{k=1}^{K}\Bigg(\underbrace{\frac{\mathbb{E}_\mu\left[f_k - \mathcal{T}^{\pi_k} f_k\right]}{1-\gamma}}_{\text{(I)}} + \underbrace{\frac{\mathbb{E}_\pi\left[\mathcal{T}^{\pi_k} f_k - f_k\right]}{1-\gamma}}_{\text{(II)}} + \underbrace{\frac{\mathbb{E}_\pi\left[f_k(s,\pi) - f_k(s,\pi_k)\right]}{1-\gamma}}_{\text{(III)}} + \sqrt{\varepsilon_{\mathcal{F}}} + \sqrt{\varepsilon_{\mathsf{stat}}} + \beta \cdot \mathcal{O}(\varepsilon_{\mathcal{F}} + \varepsilon_{\mathsf{stat}})\Bigg).
\end{aligned}
$$

(by Lemma 13)

By the same argument of Xie et al. (2021, Proof of Theorem 4.1), we know for any $k \in [K]$,

$$
\text{(I)} \leq \frac{\sqrt{\varepsilon_b} + \sqrt{V_{\max}/\beta}}{1-\gamma} \qquad (\varepsilon_b \text{ is defined in Equation (11))}
$$

and

$$
\text{(II)} \leq \frac{2\sqrt{C}(\sqrt{\varepsilon_b} + \sqrt{V_{\max}/\beta})}{1-\gamma} + \frac{\langle d^\pi \setminus \nu,\ f_k - \mathcal{T}^{\pi_k} f_k \rangle}{1-\gamma},
$$

where $C \geq 1$ can be selected arbitrarily and ν is an arbitrarily distribution that satisfies $\mathscr{C}(\nu; \mu, \mathcal{F}, \pi_k) \leq C$.

Also, using the property of the no-regret oracle, we have

$$
\frac{1}{K}\sum_{k=1}^{K}\text{(III)} = o(1)
$$

Note that $\sqrt{\varepsilon_b} = \mathcal{O}(\sqrt{\varepsilon_{\mathcal{F}}} + \sqrt{\varepsilon_{\mathcal{F},\mathcal{F}}} + \sqrt{\varepsilon_{\mathsf{stat}}})$. Then, combine them all, we obtain,

$$
\begin{aligned}
&J(\pi) - J(\bar{\pi}) \\
&\leq \mathcal{O}\left(\frac{\sqrt{C}\left(\sqrt{\varepsilon_{\mathcal{F}}} + \sqrt{\varepsilon_{\mathcal{F},\mathcal{F}}} + \sqrt{\varepsilon_{\mathsf{stat}}} + \sqrt{V_{\max}/\beta}\right)}{1-\gamma} + \beta(\varepsilon_{\mathcal{F}} + \varepsilon_{\mathsf{stat}})\right) + \frac{1}{K}\sum_{k=1}^{K}\frac{\langle d^\pi \setminus \nu,\ f_k - \mathcal{T}^{\pi_k} f_k \rangle}{1-\gamma}.
\end{aligned}
$$

Algorithm 3 ATAC (Detailed Practical Version)

Input: Batch data $\mathcal{D}$, policy π, critics f_1, f_2, constants $\beta \geq 0$, $\tau \in [0,1]$, $w \in [0,1]$, entropy lower bound $\text{Entropy}_{\min}$

1: Initialize target networks $\bar{f}_1 \leftarrow f_1$, $\bar{f}_2 \leftarrow f_2$
2: Initialize Lagrange multiplier $\alpha \leftarrow 1$
3: **for** $k = 1, 2, \ldots, K$ **do**
4: Sample minibatch $\mathcal{D}_{\text{mini}}$ from dataset $\mathcal{D}$.
5: For $f \in \{f_1, f_2\}$, update critic networks
$l_{\text{critic}}(f) \coloneqq \mathcal{L}_{\mathcal{D}_{\text{mini}}}(f, \pi) + \beta \mathcal{E}^{w}_{\mathcal{D}_{\text{mini}}}(f, \pi)$
\# $f \leftarrow \text{Proj}_{\mathcal{F}}(f - \eta_{\text{fast}} \nabla l_{\text{critic}})$
$f \leftarrow \text{ADAM}(f, \nabla l_{\text{critic}}, \eta_{\text{fast}})$
$f \leftarrow \text{ClipWeightL2}(f)$
6: Update actor network
\# $l_{\text{actor}}(\pi) \coloneqq -\mathcal{L}_{\mathcal{D}_{\text{mini}}}(f_1, \pi)$
\# $\pi \leftarrow \text{Proj}_{\Pi}(\pi - \eta_{\text{slow}} \nabla l_{\text{actor}})$
$\tilde{l}_{\text{actor}}(\pi, \alpha) = -\mathcal{L}_{\mathcal{D}_{\text{mini}}}(f_1, \pi) - \alpha(\mathbb{E}_{\mathcal{D}_{\text{mini}}}[\pi \log \pi] + \text{Entropy}_{\min})$
$\pi \leftarrow \text{ADAM}(\pi, \nabla_{\pi} l_{\text{actor}}, \eta_{\text{slow}})$
$\alpha \leftarrow \text{ADAM}(\alpha, -\nabla_{\alpha} \tilde{l}_{\text{actor}}, \eta_{\text{fast}})$
$\alpha \leftarrow \max\{0, \alpha\}$
7: For $(f, \bar{f}) \in \{(f_i, \bar{f}_i)\}_{i=1,2}$, update target networks
$\bar{f} \leftarrow (1-\tau)\bar{f} + \tau f$.
8: **end for**

Therefore, we choose $\beta = \Theta\left(\frac{V_{\max}^{1/3}}{(\varepsilon_{\mathcal{F}} + \varepsilon_{\text{stat}})^{2/3}}\right)$, and obtain

$$J(\pi) - J(\bar{\pi}) \leq \mathcal{O}\left(\frac{\sqrt{C}\left(\sqrt{\varepsilon_{\mathcal{F}}} + \sqrt{\varepsilon_{\mathcal{F},\mathcal{F}}} + \sqrt{\varepsilon_{\text{stat}}} + (V_{\max}\varepsilon_{\mathcal{F}} + V_{\max}\varepsilon_{\text{stat}})^{1/3}\right)}{1-\gamma}\right) + \frac{1}{K}\sum_{k=1}^{K} \frac{\langle d^{\pi} \setminus \nu,\ f_k - \mathcal{T}^{\pi_k} f_k \rangle}{1-\gamma}.$$

This completes the proof. □

C. Experiment Details

C.1. Implementation Details

We provide a more detailed version of our practical algorithm Algorithm 2 in Algorithm 3, which shows how the actor and critic updates are done with ADAM. As mentioned in Section 4.2, the projection in $\pi \leftarrow \text{Proj}_{\Pi}(\pi - \eta_{\text{slow}} \nabla l_{\text{actor}})$ of the pseudo-code Algorithm 2 is done by a further Lagrange relaxation through introducing a Lagrange multiplier $\alpha \geq 0$. We update α in the fast timescale η_{fast}, so the policy entropy $\mathbb{E}_{\mathcal{D}}[-\pi \log \pi]$ can be maintained above a threshold $\text{Entropy}_{\min}$, roughly following the path of the projected update in Line 5 in the pseudo code in Algorithm 2. $\text{Entropy}_{\min}$ is set based on the heuristic used in SAC (Haarnoja et al., 2018).

In implementation, we use separate 3-layer fully connected neural networks to realize the policy and the critics, where each hidden layer has 256 neurons and ReLU activation and the output layer is linear. The policy is Gaussian, with the mean and the standard deviation predicted by the neural network. We impose an l_2 norm constraint of 100 for the weight (not the bias) in each layer of the critic networks.

The first-order optimization is implemented by ADAM (Kingma & Ba, 2015) with a minibatch size $|\mathcal{D}_{\text{mini}}| = 256$, and the two-timescale stepsizes are set as $\eta_{\text{fast}} = 0.0005$ and $\eta_{\text{slow}} = 10^{-3}\eta_{\text{fast}}$. These stepsizes η_{fast} and η_{slow} were selected *offline* with a heuristic: Since ATAC with $\beta = 0$ is IPM-IL, we did a grid search (over $\eta_{\text{fast}} \in \{5e-4, 5e-5, 5e-6\}$ and $\eta_{\text{slow}} = \{5e-5, 5e-6, 5e-7\}$, on the *hopper-medium* and *hopper-expert* datasets) and selected the combination that attains the lowest ℓ_2 IL error after 100 epochs.

We set $w = 0.5$ in Eq.(7), as we show in the ablation (Figure 3) that either $w = 0$ and $w = 1$ leads to bad numerical stability and/or policy performance. We use $\tau = 0.005$ for target network update from the work of Haarnoja et al. (2018). The discount is set to the common $\gamma = 0.99$.

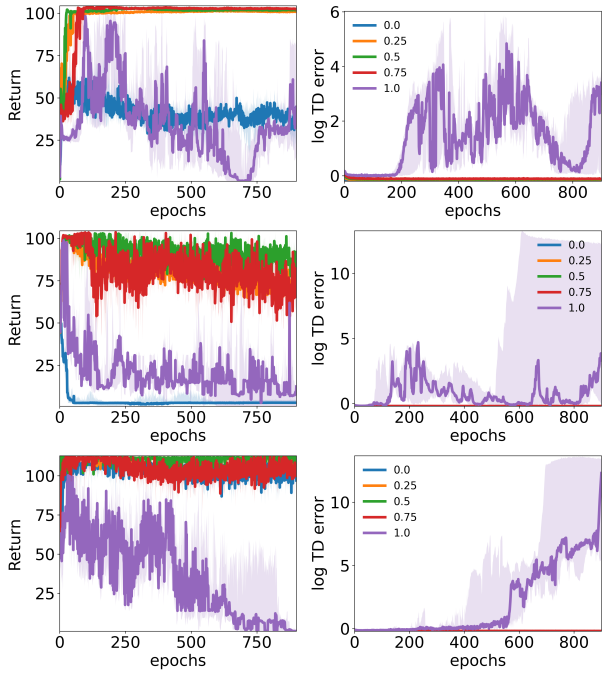


Figure 3. Ablation of the DQRA loss with different mixing weights w in Eq.(7). The plots show the policy performance and TD error across optimization epochs of ATAC with the *hopper-medium-replay*, *hopper-medium*, and *hopper-medium-expert* datasets from top to buttom. The stability and performance are greatly improved when $w \in (0, 1)$. For each w, the plot shows the 25^{th}, 50^{th}, 75^{th} percentiles over 10 random seeds.

The regularization coefficient β is our only hyperparameter that varies across datasets based on an online selection. We consider β in $\in \{0, 4^{-4}, 4^{-3}, 4^{-2}, 4^{-1}, 1, 4, 4^2, 4^3, 4^4\}$. For each β, we perform ATAC training with 10 different seeds: for each seed, we run 100 epochs of BC for warm start and 900 epochs of ATAC, where 1 epoch denotes 2K gradient updates. During the warmstart, the critics are optimized to minimize the Bellman surrogate $\mathcal{E}^w_{\mathcal{D}_{\text{mini}}}(f, \pi)$ except for $\beta = 0$.

Since ATAC does not have guarantees on last-iterate convergence, we report also the results of both the last iterate (denoted as ATAC and ATAC$_0$) and the best checkpoint (denoted as ATAC* and ATAC$^*_0$) selected among 9 checkpoints (each was made every 100 epochs).

We argue that the online selection of β and few checkpoints are reasonable for ATAC, as ATAC theory provides robust policy improvement guarantees. While the assumptions made in the theoretical analysis does not necessarily apply to the practical version of ATAC, empirically we found that ATAC does demonstrate robust policy improvement properties in the D4RL benchmarks that we experimented with, which we will discuss more below.

C.2. Detailed Experimental Results

We used a selection of the Mujoco datasets (v2) and Adroit datasets (v1) from D4RL as our benchmark environments. For each evaluation, we roll out the mean part of the Gaussian policy for 5 rollouts and compute the Monte Carlo return. For each dataset, we report the statistical results over 10 random seeds in Table 2.

Compared with the summary we provided in the main text (Table 1), Table 2 includes also the confidence interval which shows how much the 25^{th} and the 75^{th} percentiles of performance deviate from the median (i.e. the 50^{th} percentile). In addition, Table 2 also provides the selected hyperparamter β for each method.

Overall, we see that confidence intervals are small for ATAC, except for larger variations happening in *hopper-rand*, *pen-human*, and *hammer-human*. Therefore, the performance improvement of ATAC from other offline RL baselines and behavior policies is significant. We also see that ATAC most of the time picks $\beta = 64$ for the Mujoco datasets, except for the halfcheetah domain, and has a tendency of picking smaller β as the dataset starts to contain expert trajectories (i.e. in *-exp datasets). This is reasonable, since when the behavior policy has higher performance, an agent requires less information from the reward to perform well; in the extreme of learning with trajectories of the optimal policy, the learner can be optimal

	Behavior	ATAC*	CI	β	ATAC	CI	β	$ATAC_0^*$	CI	β	$ATAC_0$	CI	β
halfcheetah-rand	-0.1	4.8	[-0.5, 0.5]	16.0	3.9	[-1.2, 0.5]	4.0	2.3	[-0.0, 0.3]	64.0	2.3	[-0.1, 0.3]	64.0
walker2d-rand	0.0	8.0	[-0.9, 0.4]	64.0	6.8	[-0.4, 1.1]	4.0	7.6	[-0.2, 0.3]	16.0	5.7	[-0.1, 0.1]	16.0
hopper-rand	1.2	31.8	[-7.2, 0.1]	64.0	17.5	[-11.2, 13.2]	16.0	31.6	[-0.9, 0.6]	64.0	18.2	[-14.0, 12.7]	16.0
halfcheetah-med	40.6	54.3	[-0.8, 0.2]	4.0	53.3	[-0.4, 0.1]	4.0	43.9	[-4.9, 0.6]	16.0	36.8	[-1.4, 1.2]	0.0
walker2d-med	62.0	91.0	[-0.5, 0.3]	64.0	89.6	[-0.2, 0.2]	64.0	90.5	[-0.4, 0.8]	64.0	89.6	[-1.2, 0.3]	64.0
hopper-med	44.2	102.8	[-0.5, 0.9]	64.0	85.6	[-7.2, 6.6]	16.0	103.5	[-0.8, 0.2]	16.0	94.8	[-11.4, 4.9]	4.0
halfcheetah-med-replay	27.1	49.5	[-0.3, 0.1]	16.0	48.0	[-0.6, 0.2]	64.0	49.2	[-0.3, 0.4]	64.0	47.2	[-0.2, 0.4]	64.0
walker2d-med-replay	14.8	94.1	[-0.2, 0.4]	64.0	92.5	[-4.5, 1.0]	16.0	94.2	[-1.2, 1.1]	64.0	89.8	[-4.1, 2.0]	16.0
hopper-med-replay	14.9	102.8	[-0.3, 0.3]	16.0	102.5	[-0.4, 0.2]	16.0	102.7	[-1.0, 1.0]	1.0	102.1	[-0.3, 0.5]	16.0
halfcheetah-med-exp	64.3	95.5	[-0.2, 0.1]	0.062	94.8	[-0.4, 0.1]	0.062	41.6	[-0.1, 2.4]	0.0	39.7	[-1.3, 1.7]	0.0
walker2d-med-exp	82.6	116.3	[-0.7, 0.5]	64.0	114.2	[-7.4, 0.7]	16.0	114.5	[-1.5, 0.8]	64.0	104.9	[-8.1, 8.0]	64.0
hopper-med-exp	64.7	112.6	[-0.3, 0.2]	1.0	111.9	[-0.3, 0.3]	1.0	83.0	[-18.0, 12.8]	1.0	46.5	[-16.6, 15.2]	0.0
pen-human	207.8	79.3	[-14.2, 16.5]	0.004	53.1	[-37.2, 21.0]	0.004	106.1	[-32.2, 7.3]	0.004	61.7	[-7.0, 27.6]	0.004
hammer-human	25.4	6.7	[-3.6, 8.2]	0.016	1.5	[-0.2, 0.1]	64.0	3.8	[-1.9, 0.9]	4.0	1.2	[-0.3, 0.7]	64.0
door-human	28.6	8.7	[-2.1, 0.1]	0.0	2.5	[-1.9, 1.5]	0.0	12.2	[-3.6, 6.7]	16.0	7.4	[-7.2, 1.3]	16.0
relocate-human	86.1	0.3	[-0.2, 0.7]	0.25	0.1	[-0.1, 0.1]	64.0	0.5	[-0.2, 1.2]	4.0	0.1	[-0.0, 0.0]	1.0
pen-cloned	107.7	73.9	[-5.2, 5.7]	0.0	43.7	[-28.2, 24.4]	0.0	104.9	[-13.5, 13.0]	0.016	68.9	[-49.0, 16.4]	0.062
hammer-cloned	8.1	2.3	[-0.2, 3.3]	16.0	1.1	[-0.4, 0.2]	0.016	3.2	[-1.6, 0.6]	4.0	0.4	[-0.2, 0.2]	0.25
door-cloned	12.1	8.2	[-4.5, 5.1]	0.062	3.7	[-2.8, 1.0]	0.016	6.0	[-5.5, 1.1]	4.0	0.0	[-0.0, 0.0]	0.0
relocate-cloned	28.7	0.8	[-0.6, 0.8]	0.062	0.2	[-0.1, 0.1]	0.004	0.3	[-0.1, 0.8]	16.0	0.0	[-0.0, 0.0]	4.0
pen-exp	105.7	159.5	[-8.4, 1.8]	1.0	136.2	[-5.4, 18.7]	0.062	154.4	[-4.1, 4.0]	4.0	97.7	[-66.7, 8.8]	0.062
hammer-exp	96.3	128.4	[-0.5, 0.2]	0.016	126.9	[-0.5, 0.3]	0.016	118.3	[-20.3, 9.9]	0.062	99.2	[-41.0, 5.6]	4.0
door-exp	100.5	105.5	[-1.6, 0.3]	0.0	99.3	[-24.1, 4.6]	0.25	103.6	[-6.5, 0.8]	64.0	48.3	[-39.3, 31.2]	64.0
relocate-exp	101.6	106.5	[-1.5, 1.0]	0.016	99.4	[-12.2, 1.9]	0.004	104.0	[-2.7, 3.1]	64.0	74.3	[-1.4, 7.1]	16.0

Table 2. Experimental results of ATAC and $ATAC_0$ on the D4RL dataset and its confidence interval. We report the median score and the 25^{th} and 75^{th} percentiles, over 10 random seeds.

	ATAC*	ATAC	CQL	TD3+BC
halfcheetah-rand	2.3	2.3	**35.4**	10.2
walker2d-rand	**8.2**	6.5	7.0	1.4
hopper-rand	**12.1**	**12.0**	10.8	11.0
halfcheetah-med	**42.9**	**42.6**	**44.4**	**42.8**
walker2d-med	**84.0**	**83.0**	74.5	79.7
hopper-med	53.3	33.5	86.6	**99.5**
halfcheetah-med-replay	43.3	41.7	**46.2**	43.3
walker2d-med-replay	**33.7**	21.8	32.6	25.2
hopper-med-replay	39.2	29.5	**48.6**	31.4
halfcheetah-med-exp	**108.4**	**107.5**	62.4	97.9
walker2d-med-exp	**111.8**	109.1	98.7	101.1
hopper-med-exp	**112.8**	**112.5**	**111.0**	**112.2**

Table 3. Results of mujoco-v0 dataset. We grayed out the results of hopper-v0 because these datasets have bug (see D4RL github).

just by IL.

We also include extra ablation results on the effectiveness of DQRA loss in stabilizing learning in Figure 3, which includes two extra hopper datasets compared with Figure 2. Similar to the results in the main paper, we see that $w = 1$ is unstable, $w = 0$ is stable but under-performing, while using $w \in (0, 1)$ strikes a balance between the two. Our choice $w = 0.5$ has the best performance in these three datasets and is numerically stable. We also experimented with the max-aggregation version recently proposed by Wang & Ueda (2021). It does address the instability issue seen in the typical bootstrapped version $w = 1$, but its results tend to be noisier compared with $w = 0.5$ as it makes the optimization landscape more non-smooth.

Lastly, we include experimental results of ATAC on D4RL mujoco-v0 datasets in Table 3. We used v2 instead of v0 in the main results, because 1) hopper-v0 has a bug (see D4RL github; for this reason they are grayed out in Table 3), and 2) some baselines we compare ATAC with also used v2 (or they didn't specify and we suspect so). Here we include these results for completeness.

C.3. Robust Policy Improvement

We study empirically the robust policy improvement property of ATAC. First we provide an extensive validation on how ATAC* performs with different β on all datasets in Figure 4 and Figure 5, which are the complete version of Figure 1. In these figures, we plot the results of ATAC* (relative pessimism) and ATAC$^*_0$ (absolute pessimism) (which is a deep learning implementation of PSPI (Xie et al., 2021)) in view of the behavior policy's performance. These results show similar trends as we have observed in Figure 1. ATAC can robustly improve from the behavior policy over a wide range of β values. In particular, we see the performance degrades below the behavior policy only for large βs, because of the following reasons. When $\beta \to 0$ ATAC converges to the IL mode, which can recover the behavior policy performance if the realizability assumption is satisfied. On the other hand, when β is too large, Proposition 6 shows that the statistical error will start to dominate and therefore lead to substandard performance. This robust policy improvement property means that practitioners of ATAC can online tune its performance by starting with $\beta = 0$ and the gradually increasing β until the performance drop, without ever dropping below the performance of behavior policy much.

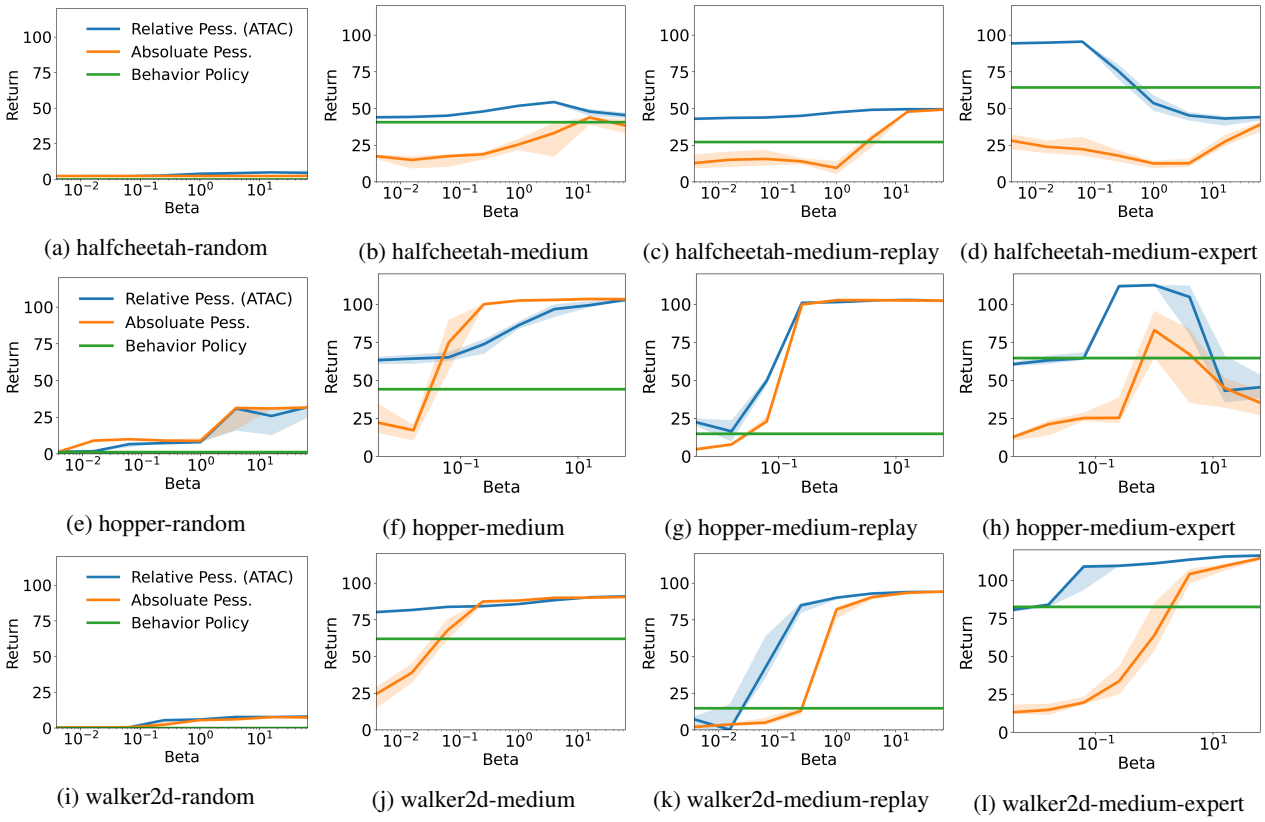
Figure 4 and Figure 5 show the robustness of ATAC* which uses the best checkpoint. Below in Table 4 we validate further whether safe policy improvement holds across iterates. To this end, we define a robust policy improvement score

$$\text{score}_{\text{RPI}}(\pi) := \frac{J(\pi) - J(\mu)}{|J(\mu)|} \tag{18}$$

which captures how a policy π performs relatively to the behavior policy μ. Table 4 shows the percentiles of the robust policy improvement score for each dataset, over *all* the β choices, random seeds, and iterates from the 100^{th} epoch to the 900^{th} epoch of ATAC training. Overall, we see that in most datasets (excluding *-human and *-clone datasets which do not satisfy our theoretical realizability assumption), more than 50% of iterates generated by ATAC across all the experiments are better than the behavior policy. For others, more than 60% of iterates are within 80 % of the behavior policy's performance. This robustness result is quite remarkable as it includes iterates where ATAC has not fully converged as well as bad choices of β.

	10^{th}	20^{th}	30^{th}	40^{th}	50^{th}	60^{th}	70^{th}	80^{th}	90^{th}	100^{th}
halfcheetah-rand	**0.9**	**1.0**	**1.0**	**1.0**	**1.0**	**1.1**	**1.4**	**1.5**	**1.8**	**5.5**
walker2d-rand	**1.9**	**2.3**	**3.5**	**5.6**	**16.5**	**64.6**	**134.0**	**139.1**	**159.2**	**519.1**
hopper-rand	**0.1**	**0.2**	**1.3**	**3.5**	**6.7**	**10.6**	**11.4**	**12.6**	**23.2**	**63.3**
halfcheetah-med	-0.1	**0.0**	**0.1**	**0.1**	**0.1**	**0.1**	**0.2**	**0.3**	**0.3**	**0.4**
walker2d-med	**0.1**	**0.2**	**0.3**	**0.3**	**0.3**	**0.4**	**0.4**	**0.4**	**0.4**	**0.5**
hopper-med	**0.2**	**0.2**	**0.3**	**0.4**	**0.4**	**0.6**	**0.7**	**0.9**	**1.1**	**1.4**
halfcheetah-med-replay	**0.6**	**0.6**	**0.6**	**0.7**	**0.7**	**0.8**	**0.8**	**0.9**	**0.9**	**1.0**
walker2d-med-replay	-1.0	-1.0	-1.0	-0.2	**3.5**	**4.4**	**4.8**	**5.0**	**5.2**	**5.5**
hopper-med-replay	-1.0	-0.9	-0.9	**1.1**	**5.4**	**6.0**	**6.0**	**6.1**	**6.1**	**6.2**
halfcheetah-med-exp	-0.6	-0.5	-0.4	-0.3	-0.2	**-0.0**	**0.2**	**0.4**	**0.5**	**0.5**
walker2d-med-exp	-0.1	-0.1	**0.0**	**0.3**	**0.3**	**0.3**	**0.3**	**0.4**	**0.4**	**0.4**
hopper-med-exp	-0.6	-0.4	-0.2	-0.2	-0.1	-0.1	**0.0**	**0.6**	**0.7**	**0.8**
pen-human	-1.0	-1.0	-1.0	-0.9	-0.9	-0.9	-0.8	-0.8	-0.7	-0.2
hammer-human	-1.1	-1.1	-1.1	-1.1	-1.0	-1.0	-1.0	-1.0	-1.0	**0.9**
door-human	-1.1	-1.1	-1.1	-1.1	-1.1	-1.1	-1.1	-1.0	-0.9	-0.1
relocate-human	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-0.9
pen-cloned	-1.0	-1.0	-1.0	-0.9	-0.9	-0.8	-0.7	-0.6	-0.5	**0.5**
hammer-cloned	-1.3	-1.3	-1.3	-1.3	-1.3	-1.2	-1.2	-1.1	-1.0	**9.2**
door-cloned	-1.2	-1.2	-1.2	-1.2	-1.2	-1.2	-1.1	-1.0	-0.8	**1.4**
relocate-cloned	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-0.7
pen-exp	-0.4	-0.2	-0.1	**0.0**	**0.1**	**0.2**	**0.2**	**0.3**	**0.4**	**0.6**
hammer-exp	-1.0	-1.0	-1.0	-0.8	-0.6	-0.2	**0.1**	**0.3**	**0.3**	**0.4**
door-exp	-1.0	-0.8	-0.6	-0.4	-0.2	-0.2	**-0.0**	**0.0**	**0.0**	**0.1**
relocate-exp	-0.4	-0.3	-0.3	-0.2	-0.2	-0.1	-0.1	**-0.0**	**0.0**	**0.1**

Table 4. The robust policy improvement scores of ATAC. We report for each dataset, the percentiles of iterates over all 9 choices of β, 10 seeds, and 800 epochs (from the 100^{th} to the 900^{th} epochs). In most datasets (excluding *-human and *-clone datasets which likely do not satisfy our theoretical realizability assumption), more than 50% of iterates generated by ATAC across all seeds and βs are better than the behavior policy. For others, more than 60% of iterates are within 80% of the behavior policy's performance.

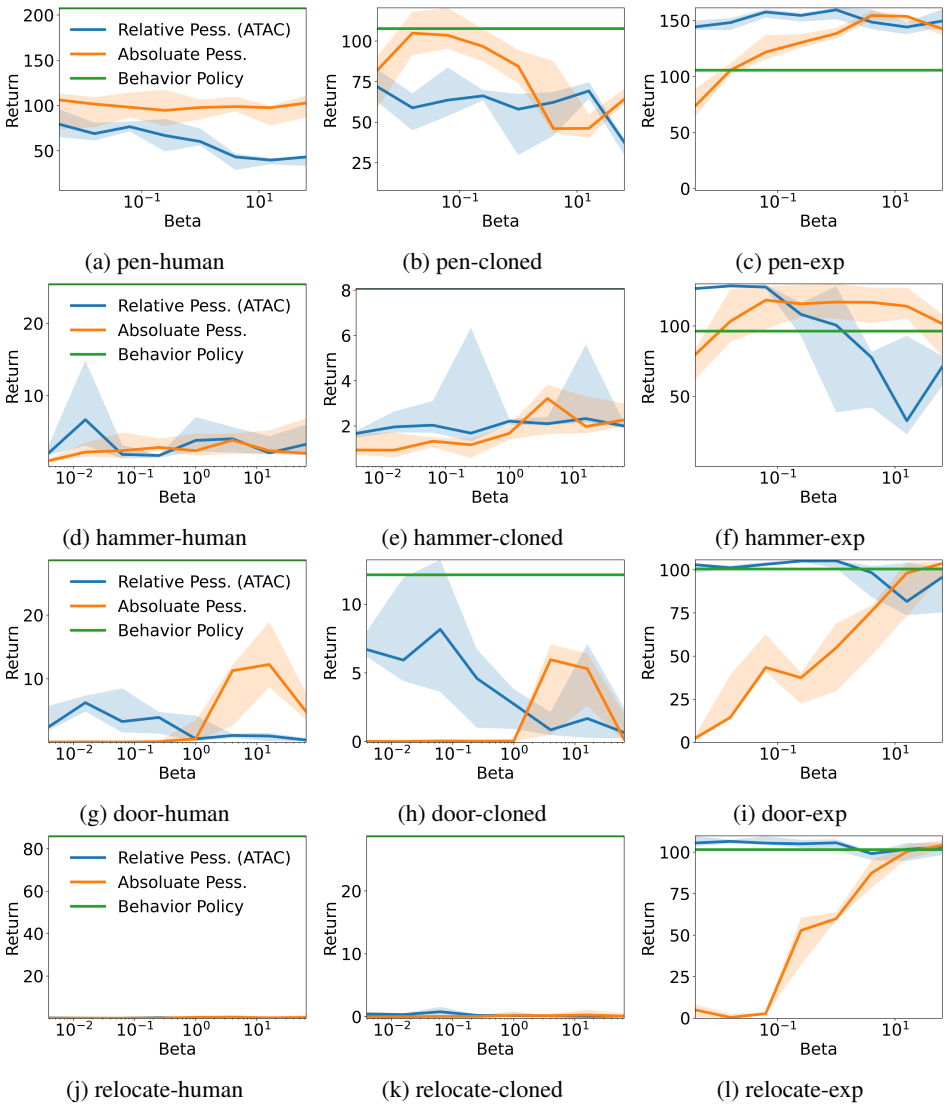


(a) halfcheetah-random (b) halfcheetah-medium (c) halfcheetah-medium-replay (d) halfcheetah-medium-expert

(e) hopper-random (f) hopper-medium (g) hopper-medium-replay (h) hopper-medium-expert

(i) walker2d-random (j) walker2d-medium (k) walker2d-medium-replay (l) walker2d-medium-expert

Figure 4. Robust Policy Improvement of ATAC in the Mujoco domains. ATAC based on *relative* pessimism improves from behavior policies over a wide range of hyperparameters that controls the degree of pessimism. On the contrary, *absolute* pessimism does not have this property and needs well-tuned hyperparameters to ensure safe policy improvement. The plots show the 25^{th}, 50^{th}, 75^{th} percentiles over 10 random seeds.



(a) pen-human (b) pen-cloned (c) pen-exp

(d) hammer-human (e) hammer-cloned (f) hammer-exp

(g) door-human (h) door-cloned (i) door-exp

(j) relocate-human (k) relocate-cloned (l) relocate-exp

Figure 5. Robust Policy Improvement of ATAC in the Adroit domains. ATAC based on *relative* pessimism improves from behavior policies over a wide range of hyperparameters that controls the degree of pessimism for the *-exp datasets. On the contrary, *absolute* pessimism does not have this property and needs well-tuned hyperparameters to ensure safe policy improvement. For *-human and *-cloned datasets, robust policy improvement is not observed empirically, likely because human demonstrators cannot be modeled by Markovian Gaussian policies (i.e. $\mu \notin \Pi$). The plots show the 25^{th}, 50^{th}, 75^{th} percentiles over 10 random seeds.

D. Comparison between ATAC and CQL

We compare ATAC with CQL (Kumar et al., 2020) in details, since they share a similar pessimistic policy evaluation procedure. In a high level, there are several major differences at the conceptual level:

1. **(Conceptual Algorithm)** ATAC describes an explicit solution concept, whereas CQL does not have a clear objective but is described as an iterative procedure. Since the convergence property and fixed point of CQL is unclear for general setups, we cannot always compare ATAC and CQL.
2. **(Maximin vs Minimax)** ATAC decouples the policy and the critic, whereas CQL aims to derive the policy from a critic. Specifically, ATAC uses a maximin formulation that finds policies performing well even for the worst case critic, whereas CQL uses a minimax formulation that finds the optimal policy for the worst case critic. In general, maximin and minimax leadto different policies.
3. **(Robust Policy Improvement)** Because of the difference between maximin and minimax in the second point, ATAC recovers behavior cloning when the Bellman term is turned off but CQL doesn't. This property is crucial to establishing the robust policy improvement property of ATAC.

ATAC and CQL also differ noticeably in the implementation design. ATAC uses the novel DQRA loss, projections, and two-timescale update; on the other hand, CQL adds an inner policy maximization, uses standard double-Q bootstrapping, and more similar step sizes for the critic and the actor.

Given such differences in both abstract theoretical reasoning and practical implementations, ATAC and CQL are two fundamentally different approaches to general offline RL, though it is likely there are special cases where the two produce the same policy (e.g. bandit problems with linear policies and critics).

Below we discuss the core differences between the two algorithms in more details.

D.1. Conceptual Algorithm

First we compare the two algorithms at the conceptual level, ignoring the finite-sample error. **ATAC has a clear objective and an accompanying iterative algorithm to find approximate solutions, whereas CQL is described directly as an iterative algorithm whose fixed point property is not established in general.**

Specifically, recall that ATAC aims to find the solution to the Stackelberg

$$\begin{aligned} &\widehat{\pi}^\star \in \operatorname*{argmax}_{\pi\in\Pi} \mathbb{E}_\mu[f(s,\pi) - f(s,a)] \\ \text{s.t.} \quad &f^\pi \in \operatorname*{argmin}_{f\in\mathcal{F}} \mathbb{E}_\mu[f(s,\pi) - f(s,a)] + \beta\mathbb{E}_\mu[((f - \mathcal{T}^\pi f)(s,a))^2] \end{aligned} \tag{19}$$

and we show that an approximate solution to the above can be found by a no-regret reduction in Algorithm 1.

On the other hand, CQL (specifically CQL ($\mathcal{R}$) in Eq.(3) of (Kumar et al., 2020)) performs the update below[9]

$$f_{k+1} \leftarrow \operatorname*{argmin}_{f\in\mathcal{F}} \max_{\pi\in\Pi} \alpha\mathbb{E}_\mu[f(s,\pi) - f(s,a)] - \mathcal{R}(\pi) + \mathbb{E}_\mu[((f - \mathcal{T}^{\pi_k} f_k)(s,a))^2] \tag{20}$$

Kumar et al. (2020) propose this iterative procedure as an approximation of a pessimistic policy iteration scheme, which alternates between pessimistic policy evaluation and policy improvement with respect to the pessimistic critic:

> *We could alternate between performing full off-policy evaluation for each policy iterate, π^k, and one step of policy improvement. However, this can be computationally expensive. Alternatively, since the policy π^k is typically derived from the Q-function, we could instead choose $\mu(a|s)$ to approximate the policy that would maximize the current Q-function iterate, thus giving rise to an online algorithm.* (Kumar et al., 2020).

Note μ in the quote above corresponds to π in the inner maximization in (20). When presenting this conceptual update rule, Kumar et al. (2020) however do not specify exactly how π_k is updated but only provide properties on the policy $\exp(f_k(s,a)/Z(s))$. Thus, below we will suppose CQL aims to find policies of quality similar to $\exp(f_k(s,a)/Z(s))$.

[9] Assume the data is collected by the behavior policy μ.

D.2. Maximin vs. Minimax

Although it is unclear what the fixed point of CQL is in general, we still can see ATAC and CQL aim to find very different policies. **ATAC decouples the policy and the critic to find a robust policy, whereas CQL aims to derive the policy from a critic function.**. This observation is reflected below.

1. ATAC is based on a maximin formulation, whereas CQL is based on minimax formulation.
2. ATAC updates policies by a no regret routine, where each policy is slow updated and determined by all the critics generated in the past iterations, whereas CQL is more akin to a policy iteration algorithm, where each policy is derived by a single critic.

We can see this difference concretely, if we specialize the two algorithms to bandit problems. In this case, CQL is no longer iterative and has a clear objective. Specifically, if we let $\alpha = \frac{1}{\beta}$, the two special cases can be written as

$$\widehat{\pi}^\star \in \operatorname*{argmax}_{\pi\in\Pi} \min_{f\in\mathcal{F}} \mathbb{E}_\mu[f(s,\pi) - f(s,a)] + \beta\mathbb{E}_\mu[((f-r)(s,a))^2] \tag{ATAC}$$

$$\widehat{f}^\star \leftarrow \operatorname*{argmin}_{f\in\mathcal{F}} \max_{\pi\in\Pi} \mathbb{E}_\mu[f(s,\pi) - f(s,a)] + \beta\mathbb{E}_\mu[((f-r)(s,a))^2] - \beta\mathcal{R}(\pi) \tag{CQL}$$

If we further ignore the extra regularization term $\mathcal{R}(\pi)$ (as that can often be absorbed into the policy class), then the main difference between the two approaches, in terms of solution concepts, is clearly the order of max and min. It is well known maximin and minimax gives different solutions in general, unless when the objective is convex-concave (with respect to the policy and critic parameterizations). For example, in this bandit special case, suppose the states and actions are tabular; the objective is convex-concave when Π and $\mathcal{F}$ contains *all* tabular functions, but convex-concave objective is lost when $\mathcal{F}$ contains a finite set of functions. In the latter scenario, CQL and ATAC would give very different policies, and CQL would not enjoy the nice properties of ATAC.

D.3. Robust Policy Improvement

We now illustrate concretely how the difference between ATAC and CQL affects the robust policy improvement property. For simplicity, we only discuss in population level.

By Proposition 3 and Proposition 6, we know $\widehat{\pi}^\star$, the learned policy from ATAC, provably improves behavior policy μ under a wide range of β choice of Eq.(19), including $\beta = 0$. In other word, as long as $\mu \in \Pi$, ATAC has $J(\widehat{\pi}^\star) \geq J(\mu)$ even if $\beta = 0$ in Eq.(19).

However, the following argument shows that: In CQL, if $\pi_f \in \Pi, \forall f \in \mathcal{F}$ and $\mathcal{F}$ contains constant functions, then setting $\beta = 0$ cannot guarantee policy improvement over μ, even when $\mu \in \Pi$, where π_f denotes the greedy policy with respect to f.

Based on what's shown before, the corresponding CQL update rule with $\beta = 0$ can be written as

$$f_{k+1} \leftarrow \operatorname*{argmin}_{f\in\mathcal{F}} \max_{\pi\in\Pi} \mathbb{E}_\mu[f(s,\pi) - f(s,a)].$$

We now prove that f_{k+1} is constant across actions in every state on the support of μ for any k:

1. $\min_{f\in\mathcal{F}} \max_{\pi\in\Pi} \mathbb{E}_\mu[f(s,\pi) - f(s,a)] = 0$, by
$$0 = \min_{f\in\mathcal{F}} \mathbb{E}_\mu[f(s,\pi) - f(s,a)]\Big|_{\pi=\mu} \leq \min_{f\in\mathcal{F}} \max_{\pi\in\Pi} \mathbb{E}_\mu[f(s,\pi) - f(s,a)] \leq \max_{\pi\in\Pi} \mathbb{E}_\mu[f(s,\pi) - f(s,a)]\Big|_{f\equiv 0} = 0.$$
2. For any $f' \in \mathcal{F}$, if there exists $(s_1, a_1) \in \mathcal{S}\times\mathcal{A}$ such that $\mu(s_1) > 0$ and $f'(s_1,a_1) > \max_{a\in\mathcal{A}\setminus a_1} f'(s_1,a)$, then $\max_{\pi\in\Pi} \mathbb{E}_\mu[f'(s,\pi) - f'(s,a)] \geq \mathbb{E}_\mu[f'(s,\pi_{f'}) - f'(s,a)] \geq \mu(s_1)(f'(s_1,a_1) - f'(s_1,\mu)) \geq \mu(s_1)(1-\mu(a_1|s_1))(f'(s_1,a_1) - \max_{a\in\mathcal{A}\setminus a_1} f'(s_1,a)) > 0$.
3. Combining the two bullets above, we obtain that f_{k+1} for all k must have $f_{k+1}(s_1,a_1) = f_{k+1}(s_1,a_2)$ for all $(s_1,a_1,a_2) \in \mathcal{S}\times\mathcal{A}\times\mathcal{A}$ such that $\mu(s_1) > 0$, i.e., f_{k+1} is constant across actions in every $s \in \mathcal{S}$ in the support of μ.

Therefore, for CQL with $\beta = 0$, the policies are updated with per-state constant functions, leading to arbitrary learned policies and failing to provide the safe policy improvement guarantee over the behavior policy μ.