An Implicit Trust Region Approach to Behavior Regularized Offline Reinforcement Learning

Zhe Zhang[1,2], **Xiaoyang Tan**[1,2*]

[1]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
[2]MIIT Key Laboratory of Pattern Analysis and Machine Intelligence
zhangzhe@nuaa.edu.cn, x.tan@nuaa.edu.cn

Abstract

We revisit behavior regularization, a popular approach to mitigate the extrapolation error in offline reinforcement learning (RL), showing that current behavior regularization may suffer from unstable learning and hinder policy improvement. Motivated by this, a novel reward shaping-based behavior regularization method is proposed, where the log-probability ratio between the learned policy and the behavior policy is monitored during learning. We show that this is equivalent to an implicit but computationally lightweight trust region mechanism, which is beneficial to mitigate the influence of estimation errors of the value function, leading to more stable performance improvement. Empirical results on the popular D4RL benchmark verify the effectiveness of the presented method with promising performance compared with some state-of-the-art offline RL algorithms.

Introduction

Deep reinforcement learning (deep RL) has achieved great success in various fields, *e.g.,* game playing (Silver et al. 2017; Usunier et al. 2016), robotic manipulation (Levine et al. 2016; Zhu et al. 2017), natural language processing (Dhingra et al. 2016; Yu et al. 2017) and so on. A striking characteristic of deep RL is its low sample efficiency and large request for online interactions with the environment. And this drawback also limits its broader applications to real-world scenes, especially for those where online data may be costly and dangerously collected, *e.g.,* autonomous driving (Gao, Sun, and Xiao 2019; Kiran et al. 2021), healthcare (Kosorok and Moodie 2015; Ling et al. 2017). The offline RL setting has recently been proposed to address the above realistic issue. In particular, offline RL seeks to learn the agent from an offline dataset collected by the behavior policy in advance while without any online data. Similar to off-policy algorithms, offline RL involves learning from behaviors generated by a policy different than the new policy. Unfortunately, the direct employment of the common off-policy methods (*e.g.,* SAC (Haarnoja et al. 2018), TD3 (Fujimoto, Hoof, and Meger 2018)) often fails to achieve the same level of performance as in the online setting.

Current studies (Fujimoto, Meger, and Precup 2019; Levine et al. 2020) mainly attribute the performance degradation to the *extrapolation error* from out-of-distribution (OOD) actions. More specifically, the Q-values of OOD actions are easily overestimated due to the *distribution shift* between the learned policy and the behavior policy (or the offline dataset). Such overestimated errors could lead to some unexpected decision-making. Hence how to mitigate the impacts of these over-optimistic OOD actions is a critical consideration for the design of offline RL. One way to achieve this goal is to penalize Q-values for OOD actions (Kumar et al. 2020; Yu et al. 2021). The resulting conservative Q-values are beneficial to alleviate the erroneously optimistic value estimation and thus cast an implicit policy constraint. However, these methods usually suffer from overly conservative value functions. In contrast, some other algorithms (Fujimoto and Gu 2021; Ghasemipour, Schuurmans, and Gu 2021; Kumar et al. 2019; Wu, Tucker, and Nachum 2019) directly impose explicit policy constraints on the policy evaluation/improvement steps so as to avoid taking potentially dangerous OOD actions. Both kinds of these methods can achieve conservative but safe policy learning, which is especially suitable for the offline RL setting.

As a typical policy constraint method, behavior regularized actor-critic (BRAC) framework (Wu, Tucker, and Nachum 2019) adopts the distance between the learned policy and the behavior policy either as the value penalty (BRAC-vp) for policy evaluation or as the policy regularization (BRAC-pr) for policy search. Nevertheless, we found that this distance constraint alone is not enough for policy improvement, as the learned policy could just wander about in the feasible region of the policy space while satisfying the constraint (i.e., staying close enough to the behavior policy, see Figure 1 for an illustration). It is worth mentioning that this problem generally exists in many current policy constraint-based offline RL methods besides BRAC.

To address this issue, an extra constraint that ensures monotonic policy improvement is necessary. For this we propose to take the distance between two successive policies into account during learning. While a naive implementation of this idea may be too computationally heavy, we present a simple but effective strategy by reshaping the immediate reward in the BRAC framework with a log-probability ratio between the learned and behavior policies. Interestingly, we

[*]Corresponding Author

show this modification naturally leads to an **i**mplicit **T**rust-**R**egion **P**olicy **O**ptimization (iTRPO) to the BRAC method. In other words, the proposed reward shaping scheme is capable of imposing the behavior regularization between consecutive policies as well as the original constraint of BRAC simultaneously. We further analyze the error propagation of the presented method, revealing that it is able to alleviate the errors accumulated in the performance bound. Experimental results on the popular D4RL benchmark demonstrate that our method yields competitive and promising performance compared to several state-of-the-art algorithms.

Background

Without loss of generality, we consider the RL problem within the framework of the Markov Decision Process (MDP) in this paper. Specifically, any MDP can be defined as a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$, where $\mathcal{S}, \mathcal{A}$ denote the state and action space, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ and $r : \mathcal{S} \times \mathcal{A} \to [R_{\min}, R_{\max}]$ represent the Markov transition probability function and reward function respectively, and $\gamma \in (0, 1)$ is the discounted factor. The goal of RL is to learn a policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ that can maximize the corresponding expected discounted cumulative return: $\eta(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^\pi, a\sim\pi}\left[r(s, a)\right]$, where d^π is the stationary state distribution and $\tau = \{s_0, a_0, \cdots\}$ represents the sampled trajectory. To evaluate the quality of policy, the Bellman operator is usually used to estimate its Q-value via bootstrapping:

$$\mathcal{T}^\pi Q(s, a) = r(s, a) + \gamma \mathbb{E}_{s'\sim P, a'\sim\pi}\left[Q(s', a')\right] \tag{1}$$

Offline Reinforcement Learning

In online RL, a critical assumption is that an agent can interact with the environment almost infinitely while learning the policy. So the estimation error can be corrected through this online trial and error. Unlike online setting, offline RL considers evaluating and learning the policy π from a fixed dataset $\mathcal{D} = \{s_i, a_i, r_i, s_{i+1}\}_{i=1}^N$ generated in advance by the unknown behavior policy π_β. Traditional off-policy methods such as SAC (Haarnoja et al. 2018) or TD3 (Fujimoto, Hoof, and Meger 2018) usually fail in this setting due to the *distribution shift* between the learned policy π and behavior policy π_β. This is because the OOD action a' rarely visited by policy π_β may be easily overestimated and chosen to construct the bootstrapped target using Eq.(1), and thus accumulate and propagate extrapolation errors in estimated value function.

Behavior Regularization in Offline RL

To mitigate the extrapolation error caused by *out-of-distribution* actions, a direct solution is to constrain the learned policy to be close to the behavior policy. Some prior works achieve this behavior regularization by restricting the policy action space to the actions provided by the offline dataset. For example, BCQ (Fujimoto, Meger, and Precup 2019) imposes the constrained space on *policy improvement* through random perturbations of actions sampled by the approximated behavior policy. Instead, EMaQ (Ghasemipour, Schuurmans, and Gu 2021) simplifies BCQ by removing the heuristic perturbation network and defining a distribution-constrained operator for *policy evaluation*.

While in some other works, behavior regularization is alternatively regarded as a distance penalty used for policy evaluation/improvement process, such as KL-divergence (Wu, Tucker, and Nachum 2019), Maximum Mean Discrepancy (Kumar et al. 2019), Fisher-divergence (Kostrikov et al. 2021), and Mean Square Error (Fujimoto and Gu 2021). A typical framework of this idea is the *Behavior Regularized Actor-Critic* (BRAC) algorithm (Wu, Tucker, and Nachum 2019). Specifically, the *value penalty* variant of BRAC adds the divergence function between learned policy π and behavior policy π_β into the target of Q-value update:

$$\min_{Q_\theta} \mathop{\mathbb{E}}_{\substack{(s,a,r,s')\sim\mathcal{D}\\ a'\sim\pi_\phi(\cdot|s')}} \Bigg[\bigg(Q_\theta(s,a) - r(s,a) - \gamma\Big(Q_{\bar\theta}(s',a') - \alpha D_{\mathrm{KL}}[\pi_\phi(\cdot|s')\|\pi_\beta(\cdot|s')]\Big)\bigg)^2\Bigg] \tag{2}$$

Where $Q_{\bar\theta}$ represents the target Q function and the next Q-value $Q_{\bar\theta}(s', a')$ is penalized by a KL-divergence term[1] $D_{\mathrm{KL}}[\pi_\phi(\cdot|s')\|\pi_\beta(\cdot|s')]$. Meanwhile, the divergence function can be used as the penalty function for constrained policy optimization, which is called the *policy regularization* variant of BRAC:

$$\max_{\pi_\phi} \mathop{\mathbb{E}}_{s\sim\mathcal{D}}\left[\mathop{\mathbb{E}}_{a\sim\pi_\phi(\cdot|s)}[Q_\theta(s,a)] - \alpha D_{\mathrm{KL}}[\pi_\phi(\cdot|s)\|\pi_\beta(\cdot|s)]\right] \tag{3}$$

Behavior regularization imposed by BRAC is useful for the new policy in handling OOD actions. However, this regularization is not the only factor that may have influence on the performance bound in offline RL. Intuitively, the new policy by BRAC may just wander around but never approach the local optimal policy π^*, even when the behavior regularization is satisfied (see Figure 1 (left) for an illustration).

Method

In this section, we first analyze the performance lower bound in offline RL and then propose our solutions motivated by this theoretical insight.

Performance Lower Bound in Offline RL

As previously mentioned, BRAC considers behavior regularization only consisting of the constraint between π and π_β when implementing the policy improvement (shown in Eq.(3)). But in fact, we found that only this regularization was not enough for the policy improvement guarantee. The following Corollary 1 provides a piece of theoretical evidence.

[1] In fact, there are several choices for the divergence function, *e.g.,* Maximum Mean Discrepancy, Wasserstein Distance and so on. Here we choose the KL-divergence for the convenience of subsequent derivation.

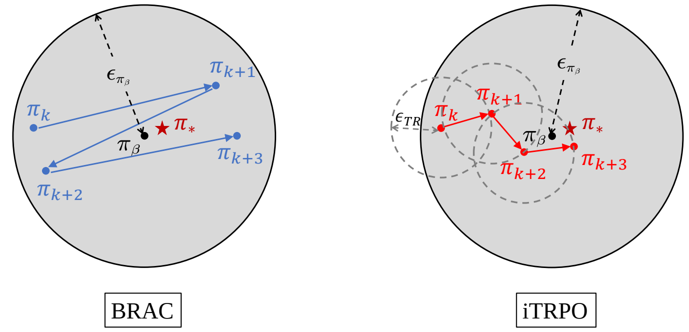


Figure 1: An illustration to compare BRAC (left) with iTRPO (right) methods. $\{\pi_k, \pi_{k+1}, \cdots\}$ is the sequence of updated policy from timestep k. The local optimal policy π_* achieves the largest policy improvement around π_β within a range ϵ_{π_β}. And ϵ_{TR} represents the scaling range of the trust region used for conservative policy updates.

Corollary 1. *Let* $D^{\max}_{\mathrm{KL}}(\pi, \tilde{\pi}) = \max_s D_{\mathrm{KL}}[\pi \| \tilde{\pi}](s)$ *and* $D^{\exp}_{\mathrm{KL}}(\pi, \tilde{\pi}) = \mathbb{E}_{s \sim d^{\tilde{\pi}}}[D_{\mathrm{KL}}[\pi \| \tilde{\pi}](s)]$. *And denote the expected advantage of* π *over* π_k *at state* s *by* $\bar{A}_{\pi,\pi_k}(s) = \mathbb{E}_{a \sim \pi(a|s)}\left[A^{\pi_k}(s,a)\right]$[2]*, and then define* $\eta(\pi) = \eta(\pi_k) + \frac{1}{1-\gamma}\mathbb{E}_{s \sim d^{\pi}}\left[\bar{A}_{\pi,\pi_k}(s)\right]$, $L_{\pi_\beta}(\pi) = \eta(\pi_k) + \frac{1}{1-\gamma}\mathbb{E}_{s \sim d^{\pi_\beta}}\left[\bar{A}_{\pi,\pi_k}(s)\right]$. $\forall k \geq 0$*, we have:*

$$\eta(\pi) \geq L_{\pi_\beta}(\pi) - C \cdot \left(D^{\max}_{\mathrm{KL}}(\pi, \pi_k) \cdot D^{\exp}_{\mathrm{KL}}(\pi, \pi_\beta)\right)^{\frac{1}{2}}$$

where $C = \frac{2\varepsilon\gamma}{(1-\gamma)^2}$ *and* $\varepsilon = \max_{s,a}\left|A^{\pi_k}(s,a)\right|$.

The proof can be found in Appendix A.1[3]. Note that, due to the lack of on-policy data, we usually have to utilize offline data and optimize the substituted objective $L_{\pi_\beta}(\pi)$ for a policy improvement in offline RL setting. And according to the lower bound of $\eta(\pi)$ shown in Corollary 1, if we want to achieve an actual policy improvement on $\eta(\pi)$, in addition to maximizing $L_{\pi_\beta}(\pi)$, it's also necessary to minimize the KL divergence between the learned policy and the behavior policy, *i.e.*, $D^{\exp}_{\mathrm{KL}}(\pi, \pi_\beta)$, as well as the last updated policy, *i.e.*, $D^{\max}_{\mathrm{KL}}(\pi, \pi_k)$.

However, according to the objectives in Eq.(2-3), BRAC only considers the behavior regularization that constrains π to be close to π_β; thus it doesn't guarantee an increasing lower bound on policy performance.

Implicit Trust Region Approach

Motivated by the above finding, we propose our solution to search for an actual improvement on the lower performance bound by supplementing the original regularization used in BRAC. For convenience, we first rewrite the objectives of the BRAC method as their vector forms.

With a slight abuse of notations, we denote probability transition matrix and policy vector by $\pi \in \Delta^{\mathcal{S}}_{\mathcal{A}}$ and $P \in \Delta^{\mathcal{S}\times\mathcal{A}}_{\mathcal{S}}$ respectively, in which Δ_X is the set of probability simplex over the set X and Y^X represents the set of applications from X to the set Y. Let $Q, r \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ be the Q-value and reward vector over state-action space where we further define a component-wise dot product as $\langle u, v \rangle = \left(\sum_a u(s,a)v(s,a)\right)_s \in \mathbb{R}^{|\mathcal{S}|}$ which is useful for the expectation over action probability, *e.g.*, $\mathbb{E}_{a\sim\pi}[Q(s,a)] = \langle \pi, Q \rangle$. Analogously, we can also define the expectation of the next Q-value as $P\langle \pi, Q \rangle = \left(\sum_{s'} P(s'|s,a) \sum_{a'} \pi(a'|s')Q(s',a')\right)_{s,a} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$. And the KL divergence between any two policies can be represented as $D_{\mathrm{KL}}\left[\pi_1 \| \pi_2\right] = \langle \pi_1, \log\pi_1 - \log\pi_2 \rangle$. Then we can combine and rewrite both variants (Eq.(2-3)) of BRAC as the Value Iteration (BRAC-VI) scheme:

$$\begin{cases} \pi_{k+1} = \arg\max\limits_{\pi} \left\langle \pi, Q_k - \alpha \log \frac{\pi}{\pi_\beta} \right\rangle \\ Q_{k+1} = r + \gamma P \left\langle \pi_{k+1}, Q_k - \alpha \log \frac{\pi_{k+1}}{\pi_\beta} \right\rangle \end{cases} \tag{4}$$

Inspired by the prior studies on Mirror Descent VI (MD-VI) (Vieillard et al. 2020; Vieillard, Pietquin, and Geist 2020), we propose to supplement behavior regularization by reshaping the immediate reward instead of adding another KL regularization. Specifically, within the BRAC framework, we reshape the immediate reward $r(s,a)$ with a scaling log-probability ratio between the learned policy π and the behavior policy π_β, *i.e.*, $\tau\alpha \log \frac{\pi_k(a|s)}{\pi_\beta(a|s)}$, for all offline samples. Accordingly, we can obtain the modified BRAC-VI objective written as:

$$\begin{cases} \pi_{k+1} = \arg\max\limits_{\pi} \left\langle \pi, Q_k - \alpha \log \frac{\pi}{\pi_\beta} \right\rangle \\ Q_{k+1} = r + \tau\alpha \log \frac{\pi_{k+1}}{\pi_\beta} \\ \qquad\qquad + \gamma P \left\langle \pi_{k+1}, Q_k - \alpha \log \frac{\pi_{k+1}}{\pi_\beta} \right\rangle \end{cases} \tag{5}$$

Where $\alpha \in \mathbb{R}^+$ controls the strength of value penalty and $\tau \in [0, 1]$ is a scaling factor that will be shown to balance two different behavior regularization later.

As the only difference between the original BRAC-VI in Eq.(4) and our modified one in Eq.(5), the reward shaping term $\tau\alpha \log \frac{\pi_{k+1}}{\pi_\beta}(a|s)$ for the current state s could somewhat offset the value penalty imposed on the next state, *i.e.*, $-\alpha \log \frac{\pi_{k+1}}{\pi_\beta}(a'|s')$, which seems to be opposite of what we intended. Nevertheless, the following Theorem 1 shows our solution can implicitly supplement behavior regularization necessary for increasing performance bound.

[2] $A^\pi(s,a)$ represents the advantage function of policy π.

[3] All Appendices can be found in https://github.com/JackZhangY/parnec.tan.publication

Theorem 1. *For any timestep $k \geq 0$, the above modified BRAC-VI scheme denoted by* Eq.(5) *is equivalent to:*

$$\begin{cases} \pi_{k+1} = \arg\max_{\pi} \langle \pi, Q'_k \rangle - \tau\alpha D_{\mathrm{KL}}\left[\pi \| \pi_k\right] \\ \qquad\qquad - (1-\tau)\alpha D_{\mathrm{KL}}\left[\pi \| \pi_\beta\right] \\ Q'_{k+1} = r + \gamma P\Big(\langle \pi_{k+1}, Q'_k \rangle - \tau\alpha D_{\mathrm{KL}}\left[\pi_{k+1} \| \pi_k\right] \\ \qquad\qquad - (1-\tau)\alpha D_{\mathrm{KL}}\left[\pi_{k+1} \| \pi_\beta\right] \Big) \end{cases}$$

Where $Q'_k \triangleq Q_k - \tau\alpha \log \frac{\pi_k}{\pi_\beta}$ is the implicit iterated Q-value.

The detailed proof can be found in Appendix A.2. As shown in Theorem 1, our modified BRAC-VI implicitly imposes the extra behavior regularization that constrains the KL divergence between two successive policies, in addition to the original KL regularization that limits the learned policy not far away from the behavior policy. And both the different behavior regularization will serve as the value penalty for the implicit Q-value (Q'_k) iteration and the policy regularization for the policy update, respectively. This result implies the goal of minimizing both KL divergence shown in Corollary 1 can be achieved by this simple reward shaping within the BRAC framework, beneficial to increase the lower performance bound.

Like prior *trust region*-based works in online RL, *e.g.,* TRPO (Schulman et al. 2015) and PPO (Schulman et al. 2017), our method can be viewed to an **i**mplicit **T**rust **R**egion **P**olicy **O**ptimization to the BRAC algorithms. So we dub our proposed method as **iTRPO** algorithm. We provide an illustrated comparison between BRAC and iTRPO methods in Figure 1. In contrast to the BRAC method, our iTRPO additively constrains the distance between successive policies within a range ϵ_{TR}, which is beneficial for step-by-step optimization to the local optimal policy π_*, instead of oscillating within a reasonable range that satisfies the distance constraint from behavior policy π_β. The algorithmic implementations of our iTRPO refer to Appendix B&C.

Theoretical Analysis

In this section, We mainly implement error propagation analysis on the proposed iTRPO method under the condition of approximated VI (Perolat et al. 2015). Following prior works (Geist, Scherrer, and Pietquin 2019; Vieillard et al. 2020), the MDP regularized by both a KL divergence and a scaled entropy is defined as follows:

Definition 1. *For a MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$, the KL-Entropy-Regularized operator is denoted by:*

$$T^{\eta,\mu}_{\pi,\pi'} Q = r + \gamma P(\langle \pi, Q \rangle - \eta D_{\mathrm{KL}}\left[\pi \| \pi'\right] + \mu \mathcal{H}(\pi)) \quad (6)$$

Where $\mathcal{H}(\pi) = \langle \pi, -\log \pi \rangle$ is the entropy of iterated policy. η and μ are the scaling coefficients of the KL divergence and entropy term. In particular, the operator with only entropy regularization is abbreviated as $T^{0,\mu}_{\pi} Q$.

Theorem 1 has shown that our proposed method actually imposes an extra KL regularization on the BRAC framework. While according to Definition 1, we can further view our iTRPO from the perspective of entropy-regularized MDP defined in the following Lemma:

Lemma 1. *When $k \to \infty$, then our proposed implicit trust region approach to offline RL can be viewed as a VI scheme for an entropy-regularized MDP as follow:*

$$\begin{cases} \pi_{k+1} = \arg\max_{\pi} \langle \pi, h_k \rangle + \lambda \mathcal{H}(\pi) \\ h_{k+1} \triangleq \hat{T}^{0,\lambda}_{\pi_{k+1}} h_k = \hat{r} + \gamma P\big(\langle \pi_{k+1}, h_k \rangle + \lambda \mathcal{H}(\pi_{k+1}) \big) \end{cases}$$

Where $\lambda = \alpha(1-\tau)$ is the scaling coefficient of the entropy term and the modified reward function is denoted by $\hat{r} = r + \alpha(1-\tau)\log \pi_\beta$ and the implicit value function also satisfies $h_k = (1-\tau)\sum_{j=0}^{k} \tau^{k-j} \big(Q'_j + \alpha(1-\tau)\log \pi_\beta \big)$.

We refer to Appendix A.3.1 for a detailed explanation and proof. Lemma 1 provides a new explanation of our iTRPO method, namely our modification in Eq.(5) transforms the original KL regularization in BRAC framework into an entropy-regularized MDP whose reward function is reshaped by the behavior policy π_β. Intuitively, the reshaped reward $\hat{r} \triangleq r + \alpha(1-\tau)\log \pi_\beta$ will reduce the immediate rewards of OOD samples more than in-distribution samples, which can achieve a similar pessimism in prior offline RL works (Shi et al. 2022; Jin, Yang, and Wang 2021; Uehara and Sun 2021; Bai et al. 2022). Meanwhile, the entropy-regularized MDP is also beneficial to the exploration in offline RL. Besides, this also implies the final convergence to the entropy-regularized optimal value function V^*_λ that could be found in (Cayci, He, and Srikant 2021; Zhan et al. 2021)

As mentioned above, extrapolation errors are considered to be mainly responsible for performance degeneration. So we're going to analyze the influence of estimation errors on the performance bound. Specifically, we carry out the theoretical analysis of our iTRPO within the AVI (Approximate Value Iteration) framework in which an error term ϵ_{k+1} is taken into account for each Q-value iteration in Eq.(5):

$$Q_{k+1} \triangleq r + \tau\alpha \log \frac{\pi_{k+1}}{\pi_\beta} + \gamma P \left\langle \pi_{k+1}, Q_k - \alpha \log \frac{\pi_{k+1}}{\pi_\beta} \right\rangle + \epsilon_{k+1} \quad (7)$$

Like previous offline RL works (Kumar et al. 2019; Munos 2003), we also define a similar concentrability coefficient in Assumption 1.

Assumption 1 (Concentrability). *Let d^π_ρ denote the stationary state-action visitation distribution for any policy π from the initial state distribution ρ. Suppose that there exists a coefficient function $c(t)$ such that for any $t \geq 0$ and $s, a \in \mathcal{S} \times \mathcal{A}$:*

$$d^\pi_\rho (P_{\pi^*})^t(s,a) \leq c(t) d^{\mathcal{D}}(s,a)$$

Where $P_{\pi^*} \in \Delta^{\mathcal{S}\times\mathcal{A}}_{\mathcal{S}\times\mathcal{A}}$ is the transition operator on state-action pairs induced by the optimal policy π^* and $d^{\mathcal{D}}$ is the visitation distribution where the offline dataset is sampled.

Note that if $c(t)$ can be bounded by a finite constant, the above assumption actually requires $d^{\mathcal{D}}$ to cover all possible policies' state-action distributions. According to the above results and assumptions, we provide the performance bound of the policies learned by our iTRPO algorithm in the following Theorem 2 (proven in Appendix A.3.2):

Theorem 2. *Following Lemma 1, we define* $\hat{r}_{\max} \triangleq \|r + \alpha(1-\tau)\log\pi_\beta\|_\infty$. *The sequence of policies learned by our iTRPO with coefficients* (α, τ) *using the given dataset* $\{s_i, a_i, r_i, s_{i+1}\}_{i=1}^{N} \sim d^{\mathcal{D}}$ *is denoted by* $\{\pi_0, \cdots, \pi_{k+1}\}$. *Given the estimation error* ϵ_{k+1} *in the AVI framework, we define* $E_{k+1} = (1-\tau)\sum_{i=1}^{k+1}\tau^{k+1-i}\epsilon_i$ *as the moving average of all past iteration errors. With these notations, the performance bound of* π_{k+1} *can be represented as:*

$$
\begin{aligned}
&V^*_\lambda(\rho) - V^{\pi_{k+1}}_\lambda(\rho) \\
\le&\frac{2}{1-\gamma}\sum_{j=1}^{k}\gamma^{k-j}\sqrt{c(k-j)}\big\|E_j\big\|_{2,d^{\mathcal{D}}} \\
&+\frac{2\gamma^k}{1-\gamma}\left(\frac{1}{1-\gamma}+\sum_{j=1}^{k}\left(\frac{\tau}{\lambda}\right)^j\right)(\hat{r}_{\max}+\lambda\log|\mathcal{A}|)
\end{aligned}
$$

According to the above theoretical result, we can see that the errors accumulated in the performance bound are mainly determined by both the concentrability of behavior policy and the moving average of errors E_{k+1}. The former fact actually emphasizes the coverage of offline dataset $\mathcal{D}$, avoiding a potentially infinite $c(k)$, while the latter one represents a smoother sum of all past errors, which can mitigate the influence of some large single-step errors.

Related Work

Trust region-based methods are commonly used in online RL for the monotonic policy improvement guarantee (Schulman et al. 2015, 2017). Due to their 'on-policyness', these methods are difficult to be extended to offline RL. To address this issue, AlgaeDICE (Nachum et al. 2019) learns the importance weights to reweight the objective of each offline sample. BPPO (Zhuang et al. 2023) applies PPO method to train with the offline dataset. Though BPPO can achieve outstanding performance, this direct use of on-policy methods for offline samples lacks theoretical justification and guarantee. While (Queeney, Paschalidis, and Cassandras 2021) develops an off-policy variant of the PPO algorithm with principled sample reuse, which can balance the stability and sample efficiency.

In contrast, we utilize simple reward shaping to achieve the implicit trust region optimization in offline RL. The proposed method is inspired by the Munchausen VI (MVI) scheme (Vieillard, Pietquin, and Geist 2020), but Table 1 shows the substantial differences from application backgrounds to practical effects between both methods. Firstly, MVI is an algorithm used for the online RL setting, while our iTRPO is designed to address the issues in offline RL. Secondly, MVI reshapes the reward by a log-policy term $\alpha\tau\log\pi$ within an entropy-regularized MDP. Instead, the BRAC framework within which our iTRPO is built can be seen as a KL-regularized MDP, and its shaping reward is the log-probability ratio between the learned policy and behavior policy, *i.e.,* $\alpha\tau\log\frac{\pi}{\pi_\beta}$. Last but not least, these two methods are derived from different motivations and achieve different effects. MVI actually aims to balance the stable learning and exploration ability in online RL by the KL and entropy-regularization, respectively. In contrast, our iTRPO seeks an increasing performance bound by both different KL-regularization in offline RL.

	MVI	**iTRPO(ours)**
Setting	online RL	offline RL
Basic MDP	Entropy-regularized MDP	KL-regularized MDP
Shaping reward	$\alpha\tau\ln\pi$	$\alpha\tau\ln\frac{\pi}{\pi_\beta}$
Implicit effect	Entropy-reg vs. KL-reg	Both KL-reg

Table 1: Comparison between MVI and our method.

Experiments

In this section, we verify the effectiveness and feasibility of our iTRPO method by comparing it with some strong baselines on the popular D4RL benchmark (Fu et al. 2020). Besides the promising performance, we further show some significant properties of the presented iTRPO method.

Experimental Setup

Baselines. In the D4RL benchmark, we compare our iTRPO method with some recent state-of-the-art baseline methods. These including algorithms attempt to solve the existing issues in offline RL from different perspectives: *a.* **BEAR** (Kumar et al. 2019) that imposes policy constraint through the MMD distance; *b.* **UWAC** (Wu et al. 2021) that reweights the TD-error according to the uncertainty estimation; *c.* **IQL** (Kostrikov, Nair, and Levine 2022) that implicitly implements Q-learning without querying on OOD actions; *d.* **TD3-BC** (Fujimoto and Gu 2021) that regularizes the policy optimization via a simple BC constraint; *e.* **CQL** (Kumar et al. 2020) that avoids OOD actions by minimizing their Q-values of them; *f.* **PBRL** (Bai et al. 2022) that achieves pessimistic bootstrapping by the uncertainty from ensemble Q-functions.

Datasets. We mainly conduct our experiments on the MuJoCo Locomotion tasks, consisting of a total of 15 combinations of 3 environments (halfcheetah, hopper, walker2d) across 5 different types of datasets (random, medium, medium-replay, medium-expert, expert). Note that we use the latest bug-fixed '-v2' datasets for the performance comparison, so we retrain CQL, IQL, and TD3-BC for their complete results on the '-v2' datasets. We train all these methods for 1 million gradient steps and take the average of the final 10 evaluations as the shown performance. As for the other algorithms, we directly take their score performance reported in previous papers (Bai et al. 2022). Besides, we also evaluate these methods on the more challenging AntMaze domain, which consists of 6 sparse-reward tasks. All the details on these reimplementations can also be found in Appendix C.

Task Name	BEAR	UWAC	IQL	TD3-BC	CQL	PBRL	**iTRPO**
halfcheetah-r	2.3±0.0	2.3±0.0	12.1±2.8	11.8±0.5	23.2±1.0	11.0±5.8	**27.4±0.3**
hopper-r	3.9±2.3	2.7±0.3	8.4±0.8	9.1±1.6	8.5±0.5	26.8±9.3	**31.6±0.1**
walker2d-r	**12.8±10.2**	2.0±0.4	6.3±1.2	1.4±1.0	5.5±1.0	8.1±4.4	5.7±4.7
halfcheetah-m	43.2±0.2	42.2±0.4	47.1±0.1	48.2±0.1	49.1±0.1	**57.9±1.5**	56.2±1.1
hopper-m	51.8±4.0	50.9±4.4	63.6±2.9	59.0±0.7	67.5±2.4	75.3±31.2	**98.5±0.6**
walker2d-m	-0.2±0.1	75.4±3.0	80.2±2.8	83.8±0.3	83.1±0.6	**89.6±0.7**	85.0±0.4
halfcheetah-m-r	36.3±3.1	35.9±3.7	44.3±0.3	44.6±0.1	45.5±0.2	45.1±8.0	**55.0±0.5**
hopper-m-r	52.2±19.3	25.3±1.7	95.4±4.9	65.4±14.0	95.5±0.8	100.6±1.0	**101.2±0.7**
walker2d-m-r	7.0±7.8	23.6±6.9	69.9±8.4	80.3±1.8	82.5±2.1	77.7±14.5	**94.2±3.1**
halfcheetah-m-e	46.0±4.7	42.7±0.3	89.2±1.2	91.7±1.9	87.7±5.2	92.3±1.1	**94.4±0.3**
hopper-m-e	50.6±25.3	44.9±8.1	101.6±7.4	101.6±2.0	104.6±2.2	110.8±0.8	**110.8±0.3**
walker2d-m-e	22.1±44.9	96.5±9.1	109.6±0.6	110.1±0.3	109.5±0.1	110.1±0.3	**110.6±0.4**
halfcheetah-e	92.7±0.6	92.9±0.6	94.8±0.2	96.7±0.5	**98.2±1.3**	92.4±1.7	95.2±0.1
hopper-e	54.6±21.0	110.5±0.5	109.1±2.0	110.2±1.9	107.7±2.4	110.5±0.4	**111.3±0.1**
walker2d-e	106.6±6.8	108.4±0.4	109.3±0.2	110.2±0.1	109.4±0.1	108.3±0.3	**111.8±1.9**
Average	38.8±10.0	50.4±2.7	69.4±1.8	68.3±1.8	71.8±1.3	74.4±5.3	**79.3±0.9**

Table 2: Normalized score comparison of all mentioned methods above on 15 MuJoCo Locomotion tasks, where r = random, m = medium, m-r = medium-replay, m-e = medium-expert, e = expert. We report the mean and standard deviation of score performance over 4 random seeds and the final 10 evaluations. We bold the highest scores.

Task Name	BEAR	TD3-BC	PLAS	CQL	IQL	SPOT	**iTRPO**
Antmaze-umaze-v2	73.0	73.0±34.0	62.0±16.7	82.6±5.7	89.6±4.2	**93.5±2.4**	92.7±1.4
Antmaze-umaze-diverse-v2	61.0	47.0±7.3	45.4±7.9	10.2±6.7	65.6±8.3	40.7±5.1	**86.5±4.3**
Antmaze-medium-play-v2	0.0	0.0±0.0	31.4±21.5	59.0±1.6	76.4±2.7	74.7±4.6	**77.9±5.3**
Antmaze-medium-diverse-v2	8.0	0.2±0.4	20.6±27.7	46.6±24.0	72.8±7.0	**79.1±5.6**	76.3±1.8
Antmaze-large-play-v2	0.0	0.0±0.0	2.2±3.8	16.4±17.1	42.0±3.8	35.3±8.3	**50.9±4.9**
Antmaze-large-diverse-v2	0.0	0.0±0.0	3.0±6.7	3.2±4.1	46.0±4.5	36.3±13.7	**50.6±4.7**
Total	142.0	120.2	164.6	218.0	392.4	359.6	**418.5**

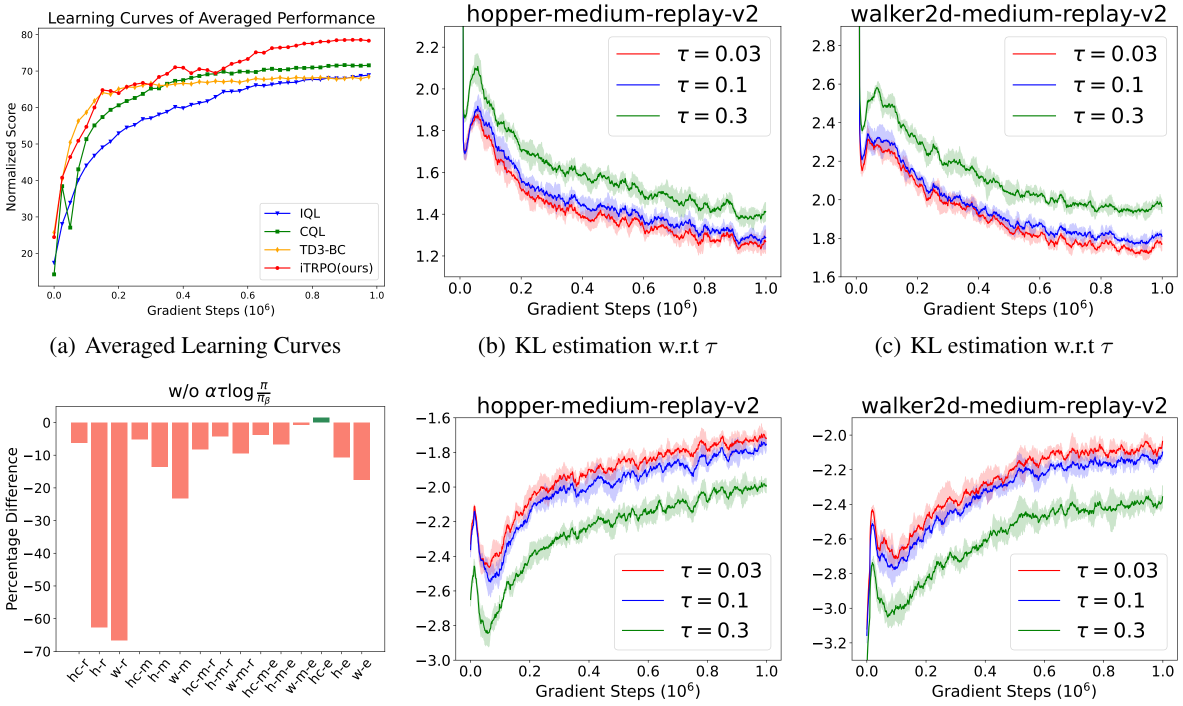
Table 3: Normalized score comparison on some Antmaze tasks. We report the mean and standard deviation of score performance over 4 random seeds. We bold the highest scores.

Performance Comparison

For Mujoco Locomotion tasks, we summarize the average normalized scores of our iTRPO method with all mentioned baselines in Table 2, where 0 score represents a random policy and 100 score corresponds to an expert policy. As the Table shows, the conservative-based methods (CQL, PBRL) generally perform better than the policy constraint-based ones (BEAR, UWAC, IQL, TD3-BC). However, our iTRPO method, which also belongs to the policy constraint-based algorithms, still significantly outperforms the conservative-based ones. Quantitatively, iTRPO achieves the highest performing scores on 11 out of 15 tasks and yields a total average score of **79.3** on all these tasks, which exceeds the second-best result of 74.4 achieved by the PBRL algorithm. Besides, our method owns a more stable convergence performance with a reduced variance (±0.9) among different tasks. Compared with the including policy constraint-based methods, our iTRPO method achieves around ~ 10.0 improvements in averaged scores. Especially for the non-optimal datasets, such as 'random', 'medium-replay', and 'medium-expert', iTRPO can achieve a more obvious performance improvement.

We also demonstrate the learning curves of average scores in Figure 2(a). And the separate learning curves for each task can be found in Appendix B. Compared with IQL and CQL algorithms, our iTRPO method has better performance and higher sample efficiency. Meanwhile, TD3-BC has a more rapid performance improvement at the early stage of learning ($< 10^5$ gradient steps), but its premature convergence also leads to inferior performance than ours.

While Table 3 shows the performance comparison across Antmaze tasks. We can see that our iTRPO method can still achieve a significant improvement compared with the SOTA algorithm, *i.e.,* IQL and SPOT(Wu et al. 2022). Especially for the difficult tasks, *e.g.* 'large-medium' and 'large-diverse', our iTRPO method obtains the best results, verifying its effectiveness for more challenging tasks. More details and learning curves can be found in Appendix C.



(a) Averaged Learning Curves

(b) KL estimation w.r.t τ

(c) KL estimation w.r.t τ

(d) Ablation Study

(e) Shaping reward w.r.t τ

(f) Shaping reward w.r.t τ

Figure 2: Performance Comparison & Property analysis.

Property Analysis

In this subsection, we record and show how the shaping reward term $\log\frac{\pi}{\pi_\beta}$ changes during the learning process and its effects on behavior regularization. Besides, we conduct the ablation study on the shaping reward, verifying its importance in performance improvement.

Effects on behavior regularization. As mentioned above, the shaping reward $\tau\alpha\log\frac{\pi}{\pi_\beta}$ implicitly leads to two different behavior regularizations, and the scaling factor τ can achieve the trade-off between them. So we estimate $D_{\mathrm{KL}}[\pi\|\pi_\beta]$ with respect to different τ, showing the varying strength of behavior regularization in Figure 2(b) & 2(c). We observe that the KL divergence between π and π_β would increase as the coefficient τ increases. This implies a larger τ would weaken the strength of behavior regularization that constrains the learned policy to be close to the behavior policy, which also satisfies the fact stated in Theorem 1, *i.e.,* the implicit coefficient of $D_{\mathrm{KL}}[\pi\|\pi_\beta]$ term is $1-\tau$.

Shaping reward. Now, we focus on the detailed changes in shaping rewards. Figure 2(e) & 2(f) show its averaged estimation on sampled data $(s,a)\in\mathcal{D}$. We can observe that the shaping rewards will increase as the learning progress, which means the learned policy π is approaching the behavior policy according to the definition $\log\frac{\pi}{\pi_\beta}$. Similar to the above findings, a larger τ would lead to smaller shaping rewards which suggests a bigger deviation from the behavior in the offline dataset $\mathcal{D}$. Besides, we note that the mean of shaping rewards is usually negative, and thus our iTRPO can learn a more conservative value function than the original BRAC method.

Ablation study. To verify the importance of the shaping reward in our iTRPO method, we perform an ablation study on the extra term $\alpha\tau\log\frac{\pi}{\pi_\beta}$. Figure 2(d) shows the percentage difference of the performance without the reward shaping. We can see that, without the reshaping term, the original BRAC-style method can lead to performance degeneration compared with our iTRPO method in most tasks. Especially for the low-quality datasets, *i.e.,* 'random', 'medium', the implicit trust region optimization provided by our iTRPO method plays an important role in policy improvement.

Conclusion

We found in this paper that behavior regularization alone is not enough to guarantee policy improvement, and extra behavior regularization that constrains the distance between successive policies is also of importance in searching good policy under the setting of offline RL. A simple reward-shaping solution that achieves an effect of implicit trust region policy optimization is introduced, yielding promising performance on the popular benchmark. Theoretical analysis of error propagation shows that the proposed method can mitigate the influence of estimation errors. Considering that behavior regularization has become a popular scheme in offline RL, our work highlights the importance in seeking effective solutions that yield both safe and reliable policy improvement in this context.

Acknowledgments

This work is partially supported by National Key R&D program of China (2021ZD0113203), National Science Foundation of China (61976115)

References

Bai, C.; Wang, L.; Yang, Z.; Deng, Z.; Garg, A.; Liu, P.; and Wang, Z. 2022. Pessimistic Bootstrapping for Uncertainty-Driven Offline Reinforcement Learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Cayci, S.; He, N.; and Srikant, R. 2021. Linear convergence of entropy-regularized natural policy gradient with linear function approximation. *arXiv preprint arXiv:2106.04096*.

Dhingra, B.; Li, L.; Li, X.; Gao, J.; Chen, Y.-N.; Ahmed, F.; and Deng, L. 2016. Towards end-to-end reinforcement learning of dialogue agents for information access. *arXiv preprint arXiv:1609.00777*.

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

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

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

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

Gao, Z.; Sun, T.; and Xiao, H. 2019. Decision-making method for vehicle longitudinal automatic driving based on reinforcement Q-learning. *International Journal of Advanced Robotic Systems*, 16(3): 1729881419853185.

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

Ghasemipour, S. K. S.; Schuurmans, D.; and Gu, S. S. 2021. Emaq: Expected-max q-learning operator for simple yet effective offline and online rl. In *International Conference on Machine Learning*, 3682–3691. PMLR.

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

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

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

Kosorok, M. R.; and Moodie, E. E. 2015. *Adaptive treatment strategies in practice: planning trials and analyzing data for personalized medicine*. SIAM.

Kostrikov, I.; Fergus, R.; Tompson, J.; and Nachum, O. 2021. Offline reinforcement learning with fisher divergence critic regularization. In *International Conference on Machine Learning*, 5774–5783. PMLR.

Kostrikov, I.; Nair, A.; and Levine, S. 2022. Offline Reinforcement Learning with Implicit Q-Learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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

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

Levine, S.; Finn, C.; Darrell, T.; and Abbeel, P. 2016. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1): 1334–1373.

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

Ling, Y.; Hasan, S. A.; Datla, V.; Qadir, A.; Lee, K.; Liu, J.; and Farri, O. 2017. Diagnostic inferencing via improving clinical concept extraction with deep reinforcement learning: A preliminary study. In *Machine Learning for Healthcare Conference*, 271–285. PMLR.

Munos, R. 2003. Error bounds for approximate policy iteration. In *ICML*, volume 3, 560–567. Citeseer.

Nachum, O.; Dai, B.; Kostrikov, I.; Chow, Y.; Li, L.; and Schuurmans, D. 2019. Algaedice: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*.

Perolat, J.; Scherrer, B.; Piot, B.; and Pietquin, O. 2015. Approximate dynamic programming for two-player zero-sum Markov games. In *International Conference on Machine Learning*, 1321–1329. PMLR.

Queeney, J.; Paschalidis, Y.; and Cassandras, C. G. 2021. Generalized proximal policy optimization with sample reuse. *Advances in Neural Information Processing Systems*, 34: 11909–11919.

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

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shi, L.; Li, G.; Wei, Y.; Chen, Y.; and Chi, Y. 2022. Pessimistic q-learning for offline reinforcement learning: Towards optimal sample complexity. In *International Conference on Machine Learning*, 19967–20025. PMLR.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of go without human knowledge. *nature*, 550(7676): 354–359.

Uehara, M.; and Sun, W. 2021. Pessimistic model-based offline reinforcement learning under partial coverage. *arXiv preprint arXiv:2107.06226*.

Usunier, N.; Synnaeve, G.; Lin, Z.; and Chintala, S. 2016. Episodic exploration for deep deterministic policies: An application to starcraft micromanagement tasks. *arXiv preprint arXiv:1609.02993*.

Vieillard, N.; Kozuno, T.; Scherrer, B.; Pietquin, O.; Munos, R.; and Geist, M. 2020. Leverage the average: an analysis of KL regularization in reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 12163–12174.

Vieillard, N.; Pietquin, O.; and Geist, M. 2020. Munchausen reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 4235–4246.

Wu, J.; Wu, H.; Qiu, Z.; Wang, J.; and Long, M. 2022. Supported Policy Optimization for Offline Reinforcement Learning. In *NeurIPS*.

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

Wu, Y.; Zhai, S.; Srivastava, N.; Susskind, J. M.; Zhang, J.; Salakhutdinov, R.; and Goh, H. 2021. Uncertainty Weighted Actor-Critic for Offline Reinforcement Learning. In *International Conference on Machine Learning*, 11319–11328. PMLR.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

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

Zhan, W.; Cen, S.; Huang, B.; Chen, Y.; Lee, J. D.; and Chi, Y. 2021. Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence. *arXiv preprint arXiv:2105.11066*.

Zhu, Y.; Mottaghi, R.; Kolve, E.; Lim, J. J.; Gupta, A.; Fei-Fei, L.; and Farhadi, A. 2017. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, 3357–3364. IEEE.

Zhuang, Z.; Lei, K.; Liu, J.; Wang, D.; and Guo, Y. 2023. Behavior Proximal Policy Optimization. *arXiv preprint arXiv:2302.11312*.