Supplementary Material

A Theoretical Proofs

This section provides complete proof of all the theoretical evidence mentioned in the main text.

A.1 Proof of Corollary 1

Here we define the total variation distance mentioned in the following: $D_{\mathrm{TV}}[p\|q] = \frac{1}{2}\sum_i |p_i - q_i|$.

Corollary 1. *We denote the expected advantage of* π *over* π_k *at state* s *by* $\bar{A}_{\pi,\pi_k}(s) = \mathbb{E}_{a\sim\pi(a|s)}\left[A_{\pi_k}(s,a)\right]$*, 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:*

$$\eta(\pi) \geq L_{\pi_\beta}(\pi) - \frac{2\varepsilon\gamma}{(1-\gamma)^2}\cdot\max_s\sqrt{D_{\mathrm{KL}}[\pi\|\pi_k](s)}\cdot\sqrt{\mathbb{E}_{s\sim d^{\pi_\beta}}\left[D_{\mathrm{KL}}[\pi\|\pi_\beta](s)\right]}$$

where $\varepsilon = \max_{s,a}\left|A_{\pi_k}(s,a)\right|$*.*

Proof. According to the definitions, we have:

$$\begin{aligned}
|\eta(\pi) - L_{\pi_\beta}(\pi)| &= \frac{1}{1-\gamma}\left|\sum_s\left(d^{\pi}(s) - d^{\pi_\beta}(s)\right)\bar{A}_{\pi,\pi_k}(s)\right| \\
&\leq \frac{1}{1-\gamma}\|d^{\pi} - d^{\pi_\beta}\|_1\cdot\left\|\bar{A}_{\pi,\pi_k}(s)\right\|_\infty \\
&\overset{(a)}{\leq} \frac{2\gamma}{(1-\gamma)^2}\mathbb{E}_{s\sim d^{\pi_\beta}}\left[D_{\mathrm{TV}}[\pi\|\pi_\beta](s)\right]\cdot\left\|\bar{A}_{\pi,\pi_k}(s)\right\|_\infty \\
&\overset{(b)}{\leq} \frac{4\gamma}{(1-\gamma)^2}\max_{s,a}\left|A_{\pi_k}(s,a)\right|\cdot\max_s D_{\mathrm{TV}}[\pi\|\pi_k](s)\cdot\mathbb{E}_{s\sim d^{\pi_\beta}}\left[D_{\mathrm{TV}}[\pi\|\pi_\beta](s)\right] \\
&\overset{(c)}{\leq} \frac{2\varepsilon\gamma}{(1-\gamma)^2}\cdot\max_s\sqrt{D_{\mathrm{KL}}[\pi\|\pi_k](s)}\cdot\sqrt{\mathbb{E}_{s\sim d^{\pi_\beta}}\left[D_{\mathrm{KL}}[\pi\|\pi_\beta](s)\right]} \qquad (1)
\end{aligned}$$

Note that the inequality (a) relies on [1, Lemma 3] stated as:

$$\|d^{\pi'} - d^{\pi}\|_1 \leq \frac{2\gamma}{1-\gamma}\mathbb{E}_{s\sim d^{\pi}}\left[D_{\mathrm{TV}}[\pi'\|\pi](s)\right]$$

And the inequality (b) depends on [12, Lemma 1] which shows:

$$\left|\bar{A}_{\pi,\pi_k}(s)\right| \leq 2\max_a\left|A_{\pi_k}(s,a)\right|\cdot D_{\mathrm{TV}}[\pi\|\pi_k](s)$$

So we further have:

$$\left\|\bar{A}_{\pi,\pi_k}(s)\right\|_\infty = \max_s\left|\bar{A}_{\pi,\pi_k}(s)\right| \leq 2\max_{s,a}\left|A_{\pi_k}(s,a)\right|\cdot\max_s D_{\mathrm{TV}}[\pi\|\pi_k](s)$$

As for the last inequality (c), we make use of the definition $\varepsilon = \max_{s,a}\left|A_{\pi_k}(s,a)\right|$ and the Pinsker's inequality: $D_{\mathrm{TV}}[p\|q] \leq \sqrt{\frac{1}{2}D_{\mathrm{KL}}[p\|q]}$ for any two probability distributions p, q.

Finally, according to Eq.(1), we can obtain the lower bound of $\eta(\pi)$ described in Corollary 1 □

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

A.2 Proof of Theorem 1

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

$$\begin{cases} \pi_{k+1} = \arg\max_{\pi} \langle \pi, Q'_k \rangle - \tau\alpha D_{\mathrm{KL}}\left[\pi \| \pi_k\right] - (1-\tau)\alpha D_{\mathrm{KL}}\left[\pi \| \pi_\beta\right] & \text{(2a)} \\ Q'_{k+1} = r + \gamma P\Big(\langle \pi_{k+1}, Q'_k \rangle - \tau\alpha D_{\mathrm{KL}}\left[\pi_{k+1} \| \pi_k\right] - (1-\tau)\alpha D_{\mathrm{KL}}\left[\pi_{k+1} \| \pi_\beta\right]\Big) & \text{(2b)} \end{cases}$$

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

Proof. Recalling the definition of modified BRAC-VI:

$$\begin{cases} \pi_{k+1} = \arg\max_{\pi} \left\langle \pi, Q_k - \alpha \log \frac{\pi}{\pi_\beta} \right\rangle & \text{(3a)} \\ Q_{k+1} = r + \tau\alpha \log \frac{\pi_{k+1}}{\pi_\beta} + \gamma P \left\langle \pi_{k+1}, Q_k - \alpha \log \frac{\pi_{k+1}}{\pi_\beta} \right\rangle & \text{(3b)} \end{cases}$$

Firstly, we define the term Q'_k for any timestep $k \geq 0$ as

$$Q'_k \triangleq Q_k - \tau\alpha \log \frac{\pi_k}{\pi_\beta} \tag{4}$$

Then we can rewrite the Q-value iteration in Eq.(3b) as follows:

$$\begin{aligned} Q_{k+1} - \tau\alpha \log \frac{\pi_{k+1}}{\pi_\beta} &= r + \gamma P \left\langle \pi_{k+1}, Q_k - \alpha \log \frac{\pi_{k+1}}{\pi_\beta} \right\rangle \\ \iff Q'_{k+1} &= r + \gamma P \left\langle \pi_{k+1}, Q_k - \alpha \log \frac{\pi_{k+1}}{\pi_\beta} \right\rangle \\ &= r + \gamma P \left\langle \pi_{k+1}, Q_k - \tau\alpha \log \frac{\pi_k}{\pi_\beta} + \tau\alpha \log \frac{\pi_k}{\pi_\beta} - \alpha \log \frac{\pi_{k+1}}{\pi_\beta} \right\rangle \\ &= r + \gamma P \left\langle \pi_{k+1}, Q'_k + \tau\alpha \log \frac{\pi_k}{\pi_\beta} - \alpha \log \frac{\pi_{k+1}}{\pi_\beta} \right\rangle \\ &= r + \gamma P \left\langle \pi_{k+1}, Q'_k + \tau\alpha \log \frac{\pi_k}{\pi_\beta} - \tau\alpha \log \frac{\pi_{k+1}}{\pi_\beta} + \tau\alpha \log \frac{\pi_{k+1}}{\pi_\beta} - \alpha \log \frac{\pi_{k+1}}{\pi_\beta} \right\rangle \\ &= r + \gamma P \left\langle \pi_{k+1}, Q'_k - \tau\alpha \log \frac{\pi_{k+1}}{\pi_k} - (1-\tau)\alpha \log \frac{\pi_{k+1}}{\pi_\beta} \right\rangle \\ &= r + \gamma P\Big(\langle \pi_{k+1}, Q'_k \rangle - \tau\alpha D_{\mathrm{KL}}\left[\pi_{k+1} \| \pi_k\right] - (1-\tau)\alpha D_{\mathrm{KL}}\left[\pi_{k+1} \| \pi_\beta\right]\Big) \end{aligned} \tag{5}$$

Similarly, we can rewrite the policy improvement in Eq.(3a) like the above derivations and get the results as follows:

$$\begin{aligned} &\arg\max_{\pi} \left\langle \pi, Q_k - \alpha \log \frac{\pi}{\pi_\beta} \right\rangle \\ \Rightarrow &\arg\max_{\pi} \left\langle \pi, Q_k - \tau\alpha \log \frac{\pi_k}{\pi_\beta} + \tau\alpha \log \frac{\pi_k}{\pi_\beta} - \alpha \log \frac{\pi}{\pi_\beta} \right\rangle \\ \Rightarrow &\arg\max_{\pi} \left\langle \pi, Q'_k + \tau\alpha \log \frac{\pi_k}{\pi_\beta} - \tau\alpha \log \frac{\pi}{\pi_\beta} + \tau\alpha \log \frac{\pi}{\pi_\beta} - \alpha \log \frac{\pi}{\pi_\beta} \right\rangle \\ \Rightarrow &\arg\max_{\pi} \langle \pi, Q'_k \rangle - D_{\mathrm{KL}}\left[\pi \| \pi_k\right] - (1-\tau)\alpha D_{\mathrm{KL}}\left[\pi \| \pi_\beta\right] \end{aligned} \tag{6}$$

Combining Eq.(5) and Eq.(6), we get the final result in Theorem 1:

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

□

A.3 Proof of Theorem 2

A.3.1 Preliminary Lemmas

Before starting the proof of Theorem 2, we introduce some useful notations and lemmas for the convenience of follow-up. Firstly, we introduce the KL-Entropy-Regularized operator as the following definition states:

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

$$T^{\eta,\mu}_{\pi,\pi'} Q = r + \gamma P(\langle \pi, Q \rangle - \eta D_{\mathrm{KL}}\left[\pi \| \pi'\right] + \mu \mathcal{H}(\pi)) \tag{7}$$

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

Then we further rewrite the implicit scheme in Theorem 1 as shown in the following Lemma:

Lemma 1. *For any $\tau \in (0, 1)$, the implicit VI scheme presented in Theorem 1 can further be written as the following scheme:*

$$\begin{cases} \pi_{k+1} = \arg\max_{\pi} \langle \pi, h_k \rangle + \alpha(1-\tau)\mathcal{H}(\pi) & \text{(8a)} \\ Q'_{k+1} = T^{\tau\alpha, \alpha(1-\tau)}_{\pi_{k+1},\pi_k}\big(Q'_k + \alpha(1-\tau)\log \pi_\beta\big) & \text{(8b)} \\ h_{k+1} = \tau h_k + (1-\tau)\big(Q'_{k+1} + \alpha(1-\tau)\log \pi_\beta\big) & \text{(8c)} \end{cases}$$

Proof. It's easy to find out that the Q-value iteration in Eq.(2b) can be represented as:

$$\begin{aligned} Q'_{k+1} &= r + \gamma P\Big(\big\langle \pi_{k+1}, Q'_k \big\rangle - \tau\alpha D_{\mathrm{KL}}\left[\pi_{k+1} \| \pi_k\right] - (1-\tau)\alpha D_{\mathrm{KL}}\left[\pi_{k+1} \| \pi_\beta\right]\Big) \\ &= r + \gamma P\Big(\big\langle \pi_{k+1}, Q'_k \big\rangle - \tau\alpha D_{\mathrm{KL}}\left[\pi_{k+1} \| \pi_k\right] + \alpha(1-\tau)\big(\mathcal{H}(\pi_{k+1}) + \langle \pi_{k+1}, \log \pi_\beta \rangle\big)\Big) \\ &= r + \gamma P\Big(\big\langle \pi_{k+1}, Q'_k + \alpha(1-\tau)\log \pi_\beta \big\rangle - \tau\alpha D_{\mathrm{KL}}\left[\pi_{k+1} \| \pi_k\right] + \alpha(1-\tau)\mathcal{H}(\pi_{k+1})\Big) \\ &= T^{\tau\alpha, \alpha(1-\tau)}_{\pi_{k+1},\pi_k}\big(Q'_k + \alpha(1-\tau)\log \pi_\beta\big) \end{aligned} \tag{9}$$

Then we consider the policy update step in Eq.(2a), we first rewrite it as the entropy-regularized optimization:

$$\begin{aligned} \pi_{k+1} &= \arg\max_{\pi} \big\langle \pi, Q'_k \big\rangle - \tau\alpha D_{\mathrm{KL}}\left[\pi \| \pi_k\right] - (1-\tau)\alpha D_{\mathrm{KL}}\left[\pi \| \pi_\beta\right] \\ &= \arg\max_{\pi} \big\langle \pi, Q'_k \big\rangle - \tau\alpha \big\langle \pi, \log \frac{\pi}{\pi_k} \big\rangle - \alpha(1-\tau)\big\langle \pi, \log \frac{\pi}{\pi_\beta} \big\rangle \\ &= \arg\max_{\pi} \big\langle \pi, Q'_k + \alpha(1-\tau)\log \pi_\beta + \tau\alpha \log \pi_k \big\rangle + \alpha\mathcal{H}(\pi) \end{aligned} \tag{10}$$

According to Legendre-Fenchel transform (or convex conjugate), we know that the solution of entropy-regularized policy optimization is the *softmax*-like policy:

$$\arg\max_{\pi} \langle \pi, Q \rangle + \mathcal{H}(\pi) = \frac{\exp Q}{\langle \mathbf{1}, \exp Q \rangle} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|} \tag{11}$$

Therefore the solution to Eq.(10) satisfies $\pi_{k+1} \propto \exp \frac{Q'_k + \alpha(1-\tau)\log \pi_\beta + \tau\alpha \log \pi_k}{\alpha}$, By repeating recursion, we get the relationship between π_{k+1} with $\{Q'_0, \cdots, Q'_k\}$:

$$\begin{aligned} \log \pi_{k+1} &= \frac{Q'_k + \alpha(1-\tau)\log \pi_\beta}{\alpha} + \tau \log \pi_k + \mathrm{const} \\ &= \frac{Q'_k + \alpha(1-\tau)\log \pi_\beta}{\alpha} + \tau \frac{Q'_{k-1} + \alpha(1-\tau)\log \pi_\beta}{\alpha} + \tau^2 \log \pi_{k-1} + \mathrm{const} \\ &\cdots \\ &= \frac{1}{\alpha(1-\tau)}\left((1-\tau)\sum_{j=0}^{k} \tau^{k-j}\Big(Q'_j + \alpha(1-\tau)\log \pi_\beta\Big)\right) + \mathrm{const} \end{aligned} \tag{12}$$

Where const refers to a constant that is irrelevant to any action and used to normalize the action probability. We denote the moving average of past biased Q-values $Q'_j + \alpha(1-\tau)\log\pi_\beta$ by h_k, and with $h_{-1} = 0$:

$$\begin{aligned} h_k &= (1-\tau)\sum_{j=0}^{k}\tau^{k-j}\Big(Q'_j + \alpha(1-\tau)\log\pi_\beta\Big) \\ &= \tau h_{k-1} + (1-\tau)(Q'_k + \alpha(1-\tau)\log\pi_\beta)) \end{aligned} \tag{13}$$

Plugging the result in Eq.(13) into Eq.(12), we can get:

$$\pi_{k+1} \propto \exp\frac{h_k}{\alpha(1-\tau)}$$

Recalling the optimization solution in Eq.(11), the original policy optimization is equivalent to:

$$\pi_{k+1} = \arg\max_{\pi}\langle\pi, h_k\rangle + \alpha(1-\tau)\mathcal{H}(\pi) \tag{14}$$

Combining Eq.(9), Eq.(13) and Eq.(14), we can get the final results shown in Lemma 1. □

Lemma 2. *The relationship between Q-value Q'_k and h-value h_k in Lemma 1 can be represented as:*

$$Q'_{k+1} = T^{\tau\alpha,\alpha(1-\tau)}_{\pi_{k+1},\pi_k}\big(Q'_k + \alpha(1-\tau)\log\pi_\beta\big) = \frac{1}{1-\tau}\Big(T^{0,\alpha(1-\tau)}_{\pi_{k+1}}h_k - \tau T^{0,\alpha(1-\tau)}_{\pi_k}h_{k-1}\Big) \tag{15}$$

Proof. According to Lemma 1, we know that π_k is actually a softmax policy w.r.t $h_{k-1}/\alpha(1-\tau)$, and then we have:

$$\log\pi_k = \frac{h_{k-1}}{\alpha(1-\tau)} - \log\left\langle\mathbf{1}, \exp\frac{h_{k-1}}{\alpha(1-\tau)}\right\rangle \tag{16}$$

Plugging Eq.(16) into the Eq.(8b) yields

$$\begin{aligned} Q'_{k+1} &= T^{\tau\alpha,\alpha(1-\tau)}_{\pi_{k+1},\pi_k}\big(Q'_k + \alpha(1-\tau)\log\pi_\beta\big) \\ &= r + \gamma P\Big(\big\langle\pi_{k+1}, Q'_k + \alpha(1-\tau)\log\pi_\beta + \tau\alpha\log\pi_k\big\rangle - \alpha\big\langle\pi_{k+1}, \log\pi_{k+1}\big\rangle\Big) \\ &= r + \frac{\gamma}{1-\tau}P\Bigg(\Big\langle\pi_{k+1}, (1-\tau)\Big(Q'_k + \alpha(1-\tau)\log\pi_\beta\Big) + \tau h_{k-1}\Big\rangle + \alpha(1-\tau)\mathcal{H}(\pi_{k+1}) \\ &\qquad - \tau\alpha(1-\tau)\log\left\langle\mathbf{1}, \exp\frac{h_{k-1}}{\alpha(1-\tau)}\right\rangle\Bigg) \\ &\overset{(a)}{=} \frac{r}{1-\tau} - \frac{\tau r}{1-\tau} + \frac{\gamma}{1-\tau}P\Bigg(\big\langle\pi_{k+1}, h_k\big\rangle + \alpha(1-\tau)\mathcal{H}(\pi_{k+1}) \\ &\qquad - \tau\Big(\big\langle\pi_k, h_{k-1}\big\rangle + \alpha(1-\tau)\mathcal{H}(\pi_k)\Big)\Bigg) \\ &\overset{(b)}{=} \frac{1}{1-\tau}\Big(T^{0,\alpha(1-\tau)}_{\pi_{k+1}}h_k - \tau T^{0,\alpha(1-\tau)}_{\pi_k}h_{k-1}\Big) \end{aligned} \tag{17}$$

Note that (a) utilizes Eq.(13) and the fact that $\max_\pi\langle\pi, Q\rangle + \alpha\mathcal{H}(\pi) = \alpha\log\langle\mathbf{1}, \exp Q/\alpha\rangle \in \mathbb{R}^{|\mathcal{S}|}$. And the definition in Eq.(1) is used for the final equation (b). □

In practice, the iterated Q-value is usually approximated by parameterized functions (*e.g.,* Neural Network), which can accumulate errors into the updated Q-value function. Approximate Value Iteration (AVI) is a common method to analyze the influence of cumulative errors. Specifically, AVI introduces an explicit error term ϵ_k into the update of Q-value. Within the framework of AVI, the approximated Q-value in Eq.(8b) can be defined as:

$$Q'_{k+1} = T^{\tau\alpha,\alpha(1-\tau)}_{\pi_{k+1},\pi_k}\big(Q'_k + \alpha(1-\tau)\log\pi_\beta\big) + \epsilon_{k+1} \tag{18}$$

Then we have the approximated h-value in Eq.(8c) as follows:

Lemma 3. *Assume the initial Q-value Q'_0, following the approximated Q-value update in* Eq.(18) *within the implicit VI scheme in* Eq.(8) *and denote the negative moving average of the past approximated error term by $E_{k+1} = -(1-\tau)\sum_{j=1}^{k+1} \tau^{k+1-j}\epsilon_j = \tau E_k - (1-\tau)\epsilon_{k+1}$ (with $E_0 = 0$). Then for any $k \geq 0$, we have that:*

$$h_{k+1} = T^{0,\alpha(1-\tau)}_{\pi_{k+1}} h_k - E_{k+1} + \alpha(1-\tau)(1-\tau^{k+1}) \log \pi_\beta - \tau^{k+1}\big(T^{0,\alpha(1-\tau)}_{\pi_0} h_{-1} - h_0\big) \quad (19)$$

Proof. Recalling the definition of h_{k+1} in Eq.(13) and using Lemma 2, it then holds that:

$$\begin{aligned}
h_{k+1} &= (1-\tau)\sum_{j=0}^{k+1} \tau^{k+1-j}\Big(Q'_j + \alpha(1-\tau)\log \pi_\beta\Big) \\
&= (1-\tau)\sum_{j=1}^{k+1} \tau^{k+1-j}\Big(Q'_j + \alpha(1-\tau)\log_\beta\Big) + (1-\tau)\tau^{k+1}\Big(Q'_0 + \alpha(1-\tau)\log \pi_\beta\Big) \\
&= (1-\tau)\sum_{j=0}^{k} \tau^{k-j}\Big(Q'_{j+1} + \alpha(1-\tau)\log_\beta\Big) + (1-\tau)\tau^{k+1}\Big(Q'_0 + \alpha(1-\tau)\log \pi_\beta\Big) \\
&= (1-\tau)\sum_{j=0}^{k} \tau^{k-j}\Big(\frac{1}{1-\tau}\Big(T^{0,\alpha(1-\tau)}_{\pi_{j+1}} h_j - \tau T^{0,\alpha(1-\tau)}_{\pi_j} h_{j-1}\Big) + \epsilon_{j+1} + \alpha(1-\tau)\log \pi_\beta\Big) \\
&\quad + (1-\tau)\tau^{k+1}\Big(Q'_0 + \alpha(1-\tau)\log \pi_\beta\Big) \\
&= T^{0,\alpha(1-\tau)}_{\pi_{k+1}} h_k - \tau^{k+1} T^{0,\alpha(1-\tau)}_{\pi_0} h_{-1} + (1-\tau)\sum_{j=0}^{k} \tau^{k-j}\epsilon_{j+1} + (1-\tau)\sum_{j=0}^{k} \tau^{k-j}\alpha(1-\tau)\log \pi_\beta \\
&\quad + \tau^{k+1}(1-\tau)\Big(Q'_0 + \alpha(1-\tau)\log \pi_\beta\Big) \\
&\overset{(a)}{=} T^{0,\alpha(1-\tau)}_{\pi_{k+1}} h_k - \tau^{k+1} T^{0,\alpha(1-\tau)}_{\pi_0} h_{-1} + (1-\tau)\sum_{j=0}^{k} \tau^{k-j}\epsilon_{j+1} + \alpha(1-\tau)(1-\tau^{k+1})\log \pi_\beta + \tau^{k+1} h_0 \\
&\overset{(b)}{=} T^{0,\alpha(1-\tau)}_{\pi_{k+1}} h_k - \tau^{k+1} T^{0,\alpha(1-\tau)}_{\pi_0} h_{-1} - E_{k+1} + \alpha(1-\tau)(1-\tau^{k+1})\log \pi_\beta + \tau^{k+1} h_0 \\
&= T^{0,\alpha(1-\tau)}_{\pi_{k+1}} h_k - E_{k+1} + \alpha(1-\tau)(1-\tau^{k+1})\log \pi_\beta - \tau^{k+1}\big(T^{0,\alpha(1-\tau)}_{\pi_0} h_{-1} - h_0\big) \quad (20)
\end{aligned}$$

The above derivations are based on the following facts:

(a) According to definition in Eq.(13), we have $h_0 = \tau h_{-1} + (1-\tau)(Q'_k + \alpha(1-\tau)\log \pi_\beta)$ where $h_{-1} = 0$.

(b) Define $E_{k+1} = -(1-\tau)\sum_{j=1}^{k+1} \tau^{k+1-j}\epsilon_j = \tau E_k - (1-\tau)\epsilon_{k+1}$ with $E_0 = 0$

□

Combining Lemma 1 and Lemma 3, and assuming that (i) there are no approximated errors, *i.e.,* $E_k = 0, \forall k \geq 0$; (ii) $k \to \infty \Rightarrow \tau^{k+1} \to 0$, then we can get the following Corollary 2 directly (*i.e.,* **Lemma 1** presented in the main text).

Corollary 2. *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:*

$$\begin{cases} \pi_{k+1} = \arg\max_{\pi} \langle \pi, h_k \rangle + \alpha(1-\tau)\mathcal{H}(\pi) \\ h_{k+1} = \hat{r} + \gamma P(\langle \pi_{k+1}, h_k \rangle + \alpha(1-\tau)\mathcal{H}(\pi_{k+1})) \end{cases}$$

Where the modified reward function is denoted by $\hat{r} = r + \alpha(1-\tau)\log \pi_\beta$ and the implicit value function is defined as $h_k = (1-\tau)\sum_{j=0}^{k} \tau^{k-j}\big(Q'_j + \alpha(1-\tau)\log \pi_\beta\big)$

A.3.2 Fromal Proof of Theorem 2

Many prior works [2, 4, 10] have proven the convergence and the unique optimal (fixed) point of the entropy-regularized MDP. Specifically, for a standard entropy-regularized VI as follows:

Definition 2. *For any entropy-regularized* MDP $\{\mathcal{S}, \mathcal{A}, r, P, \gamma, \rho, \lambda\}$ *where* λ *is the temperature coefficient of entropy regularization, the entropy-regularized VI can be defined as:*

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

And for any policy π_k*, we define the soft Q-value function, V-value functions and the advantage function*[1] *by:*

$$Q^{\pi_k}_{\lambda} = r + \gamma P V^{\pi_k}_{\lambda} \tag{21}$$

$$V^{\pi_k}_{\lambda} = \langle \pi_k, Q^{\pi_k}_{\lambda} - \lambda \log \pi_k \rangle \tag{22}$$

$$A^{\pi_k}_{\lambda} = Q^{\pi_k}_{\lambda} - V^{\pi_k}_{\lambda} - \lambda \log \pi_k \tag{23}$$

Corollary 3 ([4, Proposition 3])**.** *For any policy* $\pi \in \Pi$*, the temperature coefficient of entropy regularization* $\lambda > 0$*, if the reward function is bounded by* $r_{\max}$*, i.e.,* $r(s,a) \leq r_{\max}, \forall s, a$*, then we have:*

$$Q^{\pi}_{\lambda} \leq v^{\lambda}_{\max} \triangleq \frac{r_{\max} + \lambda \log |\mathcal{A}|}{1-\gamma} \tag{24}$$

The above entropy-regularized VI scheme will converge to the optimal solutions Q^*_{λ} and π^*, and satisfies the fact that $\pi^* = \arg\max_{\pi} \langle \pi, Q^*_{\lambda} \rangle + \lambda \mathcal{H}(\pi)$ and $Q^*_{\lambda} = T^{0,\lambda}_{\pi^*} Q^*_{\lambda} = Q^{\pi^*}_{\lambda}$. Also, we have $V^*_{\lambda} = V^{\pi^*}_{\lambda}$ and $A^*_{\lambda} = A^{\pi^*}_{\lambda}$. Given an initial state distribution ρ, for any $\pi \in \Pi$, we define the state visitation distribution as follows:

$$d^{\pi}_{\rho}(s) = (1-\gamma) \sum_{s_0 \in \mathcal{S}} \rho(s_0) \sum_{t=0}^{\infty} \gamma^t \mathbf{Pr}(s_t = s | s_0, a_t \sim \pi) \tag{25}$$

Similarly, the state-action visitation distribution is defined as:

$$d^{\pi}_{\rho}(s,a) = (1-\gamma) \sum_{s_0 \in \mathcal{S}} \rho(s_0) \sum_{t=0}^{\infty} \gamma^t \mathbf{Pr}(s_t = s, a_t = a | s_0, \pi) \tag{26}$$

Note that we may slightly abuse d^{π}_{ρ} as the vector of state visitation distribution or state-action visitation distribution that is dependent on the context.

And the performance of π within the entropy-regularized VI scheme is defined as:

$$V^{\pi}_{\lambda}(\rho) = \sum_{s,a} \rho(s) \pi(a|s) \Big(Q^{\pi}_{\lambda}(s,a) - \lambda \log \pi(a|s) \Big) = \rho \langle \pi, Q^{\pi}_{\lambda} - \lambda \log \pi \rangle = \rho V^{\pi}_{\lambda} \tag{27}$$

According to this definition, [2] provides the performance difference lemma between any two policies as follows:

Lemma 4 ([2, Lemma 5])**.** *Within the entropy-regularized VI,* $\forall \pi, \pi' \in \Pi$ *and* $s_0 \sim \rho$*, we have:*

$$V^{\pi'}_{\lambda}(\rho) - V^{\pi}_{\lambda}(\rho) = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d^{\pi'}_{\rho}, a \sim \pi'} \Big[A^{\pi}_{\lambda}(s,a) + \lambda \log \frac{\pi(a|s)}{\pi'(a|s)} \Big] \tag{28}$$

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* π^* *and* $d^{\mathcal{D}}$ *is the visitation distribution where the offline dataset is sampled.*

[1] $A^{\pi_k}_{\lambda}$ is the vector representation of $A^{\pi_k}_{\lambda}(s,a) = Q^{\pi_k}_{\lambda}(s,a) - V^{\pi_k}_{\lambda}(s) - \lambda \log \pi_k(a|s)$

Definition 3. *As mentioned above, our modified BRAC-VI finally equals an entropy-regularized VI with a modified reward function $\hat{r} = r + \alpha(1-\tau)\log\pi_\beta$ and the temperature $\lambda = \alpha(1-\tau)$, so we define $\hat{r}_{\max} \triangleq \|r + \alpha(1-\tau)\log\pi_\beta\|_\infty$ and the corresponding optimal value function $\hat{v}^\lambda_{\max} = \frac{\hat{r}_{\max}+\lambda\log|\mathcal{A}|}{1-\gamma}$.*

Assumption 2. *Let the initial Q-value function Q_0 meets $Q_0 = \alpha\tau\log\pi_0 - \alpha\log\pi_\beta$ which leads to a zero-initialized implicit h-value function, i.e., $h_0 = 0$, according to the relationship between Q and h is referred to Eq.(13)*

Theorem 2. *For the sequence of policies $\{\pi_0, \cdots, \pi_{k+1}\}$ learned by our implicit trust-region VI scheme with coefficients (α, τ) using the given dataset $\{s_i, a_i, r_i, s_{i+1}\}_{i=1}^N \sim d^\mathcal{D}$, we denote the temperature coefficient by $\lambda = \alpha(1-\tau)$, and define ϵ_{k+1} as the iteration error at $k+1$-th timestep and $E_{k+1} = -(1-\tau)\sum_{i=1}^{k+1}\tau^{k+1-i}\epsilon_i$ as a moving average of all past iteration errors. With these notations, the performance bound of π_{k+1} can be represented as:*

$$\begin{aligned}&V^*_\lambda(\rho) - V^{\pi_{k+1}}_\lambda(\rho)\\ \leq&\frac{2}{1-\gamma}\sum_{j=1}^{k}\gamma^{k-j}\sqrt{c(k-j)}\big\|E_j\big\|_{2,d^\mathcal{D}} + \frac{2\gamma^k}{1-\gamma}\left(\frac{1}{1-\gamma}+\sum_{j=1}^{k}\left(\frac{\tau}{\lambda}\right)^j\right)(\hat{r}_{\max}+\lambda\log|\mathcal{A}|)\end{aligned} \tag{29}$$

Proof. Starting from Lemma 4:

$$\begin{aligned}&(1-\gamma)\Big(V^*_\lambda(\rho) - V^{\pi_{k+1}}_\lambda(\rho)\Big)\\ =&\mathbb{E}_{s\sim d_\rho^{\pi_{k+1}}, a\sim\pi_{k+1}}\left[-A^{\pi^*}_\lambda(s,a) - \lambda\log\frac{\pi^*(a|s)}{\pi_{k+1}(a|s)}\right]\\ =&\mathop{\mathbb{E}}_{s\sim d_\rho^{\pi_{k+1}}}\left[\underbrace{\mathop{\mathbb{E}}_{a\sim\pi^*}\big[Q^*_\lambda(s,a) - \lambda\log\pi^*(a|s)\big]}_{V^{\pi^*}_\lambda(s)} - \mathop{\mathbb{E}}_{a\sim\pi_{k+1}}\left[Q^*_\lambda(s,a) - \lambda\log\pi^*(a|s) + \lambda\log\frac{\pi^*(a|s)}{\pi_{k+1}(a|s)}\right]\right]\\ =&\mathbb{E}_{s\sim d_\rho^{\pi_{k+1}}}\Big[\big\langle\pi^*, Q^*_\lambda\big\rangle + \lambda\mathcal{H}(\pi^*) - \big\langle\pi_{k+1}, Q^*_\lambda\big\rangle - \lambda\mathcal{H}(\pi_{k+1})\Big]\\ \leq&\mathbb{E}_{s\sim d_\rho^{\pi_{k+1}}}\Big[\big\langle\pi^*, Q^*_\lambda\big\rangle + \lambda\mathcal{H}(\pi^*) + \underbrace{\big\langle\pi_{k+1}, h_k\big\rangle + \lambda\mathcal{H}(\pi_{k+1}) - \big\langle\pi^*, h_k\big\rangle - \lambda\mathcal{H}(\pi^*)}_{\geq 0 \text{ as } \pi_{k+1}=\arg\max_\pi\langle\pi,h_k\rangle+\lambda\mathcal{H}(\pi) \text{ in Eq.(8a)}} - \big\langle\pi_{k+1}, Q^*_\lambda\big\rangle - \lambda\mathcal{H}(\pi_{k+1})\Big]\\ =&\mathbb{E}_{s\sim d_\rho^{\pi_{k+1}}}\Big[\big\langle\pi^*, Q^*_\lambda - h_k\big\rangle - \big\langle\pi_{k+1}, Q^*_\lambda - h_k\big\rangle\Big]\\ \leq&\|Q^*_\lambda - h_k\|_{2,d_\rho^{\pi_{k+1}}\circ\pi^*} + \|Q^*_\lambda - h_k\|_{2,d_\rho^{\pi_{k+1}}\circ\pi_{k+1}}\end{aligned} \tag{30}$$

Now we consider the bound of the RHS of the above inequality. Firstly, it's important to note that, in Eq.(30), Q^*_λ is the optimal solution of entropy-regularized VI with the modified reward $\hat{r} = r+\alpha(1-\tau)\log\pi_\beta$, thus we have $Q^*_\lambda = r+\alpha(1-\tau)\log\pi_\beta+\gamma P(\langle\pi^*, Q^*_\lambda\rangle+\lambda\mathcal{H}(\pi^*)) \triangleq \hat{T}^{0,\lambda}_{\pi^*}Q^*_\lambda$. While h_k is the intermediate variable which satisfies $T^{0,\lambda}_\pi h_k \triangleq r + \gamma P(\langle\pi, h_k\rangle + \lambda\mathcal{H}(\pi))$.

Diving into the key term $Q^*_\lambda - h_k$, we can utilize Lemma 3 and have:

$$\begin{aligned}&Q^*_\lambda - h_k\\ =&Q^*_\lambda - T^{0,\lambda}_{\pi_k}h_{k-1} + E_k - \alpha(1-\tau)(1-\tau^k)\log\pi_\beta + \tau^k\big(T^{0,\lambda}_{\pi_0}h_{-1} - h_0\big)\\ =&\hat{T}^{0,\lambda}_{\pi^*}Q^*_\lambda - T^{0,\lambda}_{\pi^*}h_{k-1} + T^{0,\lambda}_{\pi^*}h_{k-1} - T^{0,\lambda}_{\pi_k}h_{k-1} + E_k - \alpha(1-\tau)(1-\tau^k)\log\pi_\beta + \tau^k\big(T^{0,\lambda}_{\pi_0}h_{-1} - h_0\big)\\ =&\underbrace{\hat{T}^{0,\lambda}_{\pi^*}Q^*_\lambda - T^{0,\lambda}_{\pi^*}h_{k-1} - \alpha(1-\tau)\log\pi_\beta}_{\gamma P_{\pi^*}(Q^*_\lambda - h_{k-1})} + \underbrace{T^{0,\lambda}_{\pi^*}h_{k-1} - T^{0,\lambda}_{\pi_k}h_{k-1}}_{\leq 0 \text{ as Eq.(8a)}} + E_k + \tau^k\big(T^{0,\lambda}_{\pi_0}h_{-1} + \alpha(1-\tau)\log\pi_\beta - h_0\big)\\ \leq&\gamma P_{\pi^*}(Q^*_\lambda - h_{k-1}) + E_k + \tau^k\big(T^{0,\lambda}_{\pi_0}h_{-1} + \alpha(1-\tau)\log\pi_\beta - h_0\big)\\ \overset{(a)}{\leq}&(\gamma P_{\pi^*})^k(Q^*_\lambda - h_0) + \sum_{j=1}^{k}(\gamma P_{\pi^*})^{k-j}\left(E_j + \tau^j\Big(T^{0,\lambda}_{\pi_0}h_{-1} + \alpha(1-\tau)\log\pi_\beta - h_0\Big)\right)\end{aligned}$$

The relation (a) is derived from recursive iterations. For any state-action distribution $\nu(s,a)$, we have:

$$\|Q^*_\lambda - h_k\|_{2,\nu}$$

$$\leq \gamma^k \|Q^*_\lambda - h_0\|_{2,\nu P^k_{\pi^*}} + \sum_{j=1}^{k} \gamma^{k-j} \left\| E_j + \tau^j \left(T^{0,\lambda}_{\pi_0} h_{-1} + \alpha(1-\tau)\log\pi_\beta - h_0 \right) \right\|_{2,\nu P^{k-j}_{\pi^*}} \tag{31}$$

For the first term on the RHS of the above inequality, we can bound it as follow:

$$\begin{aligned}
&\gamma^k \|Q^*_\lambda - h_0\|_{2,\nu P^k_{\pi^*}} \\
\leq &\gamma^k \|Q^*_\lambda\|_{2,\nu P^k_{\pi^*}} && \text{(Assumption 2)} \\
\leq &\gamma^k \frac{\hat{r}_{\max} + \lambda \log|\mathcal{A}|}{1-\gamma} && \text{(Definition 3)}
\end{aligned} \tag{32}$$

As for the second term, we have:

$$\begin{aligned}
&\sum_{j=1}^{k} \gamma^{k-j} \left\| E_j + \tau^j \left(T^{0,\lambda}_{\pi_0} h_{-1} + \alpha(1-\tau)\log\pi_\beta - h_0 \right) \right\|_{2,\nu P^{k-j}_{\pi^*}} \\
\leq &\sum_{j=1}^{k} \gamma^{k-j} \sqrt{c(k-j)} \left\| E_j \right\|_{2,d^{\mathcal{D}}} + \gamma^k \sum_{j=1}^{k} \left(\frac{\tau}{\gamma}\right)^j \left\| T^{0,\lambda}_{\pi_0} h_{-1} + \alpha(1-\tau)\log\pi_\beta \right\|_{2,\nu P^{k-j}_{\pi^*}} && \text{(Assumption 1\&2)} \\
\leq &\sum_{j=1}^{k} \gamma^{k-j} \sqrt{c(k-j)} \left\| E_j \right\|_{2,d^{\mathcal{D}}} + \gamma^k \sum_{j=1}^{k} \left(\frac{\tau}{\gamma}\right)^j \left\| r + \alpha(1-\tau)\log\pi_\beta + \gamma P \lambda \mathcal{H}(\pi_0) \right\|_{2,\nu P^{k-j}_{\pi^*}} && (\mathrm{h}_{-1} = 0) \\
\leq &\sum_{j=1}^{k} \gamma^{k-j} \sqrt{c(k-j)} \left\| E_j \right\|_{2,d^{\mathcal{D}}} + \gamma^k \sum_{j=1}^{k} \left(\frac{\tau}{\gamma}\right)^j \left(\hat{r}_{\max} + \lambda \log|\mathcal{A}| \right) && \text{(Definition 3)}
\end{aligned} \tag{33}$$

Plugging Eq.(32-33) into Eq.(31) yields:

$$\begin{aligned}
&\|Q^*_\lambda - h_k\|_{2,\nu} \\
\leq &\sum_{j=1}^{k} \gamma^{k-j} \sqrt{c(k-j)} \left\| E_j \right\|_{2,d^{\mathcal{D}}} + \gamma^k \left(\frac{1}{1-\gamma} + \sum_{j=1}^{k} \left(\frac{\tau}{\lambda}\right)^j \right) \left(\hat{r}_{\max} + \lambda \log|\mathcal{A}| \right)
\end{aligned} \tag{34}$$

For all $\forall \nu$, including $d^{\pi_{k+1}}_\rho \circ \pi^*$ and $d^{\pi_{k+1}}_\rho \circ \pi_{k+1}$, the above inequality holds, so we get the final result in Theorem 2.

□

B Experimental Details

In this section, we provide more details about the main experiments on Mujoco Locomotion tasks, helping for reliable reproductivity. Note that all experiments in this work were run on the server with an Intel Xeon(R) CPU, 64 GB of memory and a GeForce RTX 2080Ti GPU. And we build our code based on the 'rlkit' project (`https://github.com/rail-berkeley/rlkit`).

B.1 D4RL Benckmarks

As mentioned in the main text, Mujoco Locomotion domain contains 15 tasks composed of three environments (halfcheetah, hopper, walker2d) and five types of datasets (random, medium, medium-replay, medium-expert, expert). Among them, **'random'** datasets are collected by a random policy; **'medium'** datasets contain experience sampled by an early-stopping SAC agent; **'medium-replay'** datasets contain all the interactive samples during the training of the 'medium' SAC agent; **'medium-expert'** datasets combine both the suboptimal samples and the expert samples; **'expert'** datasets are made up of expert trajectories. For clarity of performance comparison, we uniformly use the normalized score to measure the performance, which is defined as:

$$\mathrm{normalizedscore} = \frac{\mathrm{averagereturn} - \mathrm{returnoftheexpertpolicy}}{\mathrm{returnoftheexpertpolicy} - \mathrm{returnoftherandompolicy}} \times 100$$

Note that all datasets are chosen their '-v2' version, which fixes some bugs. All the datasets can be downloaded from `http://rail.eecs.berkeley.edu/datasets/offline_rl/gym_mujoco_v2_old/`.

B.2 Implementation

B.2.1 Our iTRPO

Behavior policy. In our iTRPO, we need to pretrain a behavior policy used for the computation of shaping reward and KL divergence term. Though some previous studies often adopt the conditional variational autoencoder (CVAE) [8] or the Mixture of Gaussians as the [5] density estimator to model the behavior policy π_β, in this paper, we found that a simple single Gaussian model is sufficient to achieve good performance. So we choose to train a Gaussian policy model using the maximum likelihood estimation (MLE) objective:

$$\pi_\beta = \arg\max_{\pi_\theta} \mathcal{L}_{\text{MLE}}(\pi_\theta) = \mathbb{E}_{s,a\sim\mathcal{D}}[\log \pi_\theta(s,a)] \tag{35}$$

To be consistent with the learned policy, we can further add a policy entropy term $-\tau \log \pi_\theta(a|s)$. Specific hyperparameter choices are shown in the following Table 1

Table 1: Hyperparameters used to train the single Gaussian behavior policy

	Hyperparameters	Value
Training Setting	Optimizer	Adam
	Num of iterations	1e6
	Batch size	256
	Use entropy term	True
	Learning rate(LR)	1e-3
	LR decay	MultiStepRL: milestones:[8e5,9e5], gamma=0.1
MLP policy	Hidden dim	256
	Num of layers	2

Loss functions. As the core of our algorithm, we instantiate the modified BRAC-VI scheme (Eq.8 in the main text) within the actor-critic framework. We parameterize the Q-value function and policy function with Q_θ and π_ϕ, respectively. Then we get the final loss function of the critic objective for the offline data $(s,a,r,s') \sim \mathcal{D}$:

$$\mathcal{L}(\theta) = \mathbb{E}_{s,a,r,s'\sim\mathcal{D}}\left[\left(Q_\theta(s,a) - y_{\text{target}}\right)^2\right] \tag{36}$$

Where the update target y_{target} is defined as:

$$y_{\text{target}} = r(s,a) + \tau\alpha \log\frac{\pi_\phi(a|s)}{\pi_\beta(a|s)} + \gamma\left(\min_{i=1,2} Q_{\bar{\theta}_i}(s',a') - \alpha \cdot \mathop{\mathbb{E}}_{\{a'_j\}^N\sim\pi_\beta}\left[\frac{1}{N}\sum_{j=1}^{N}\frac{\log\pi_\phi(a'_j|s')}{\log\pi_\beta(a'_j|s')}\right]\right) \tag{37}$$

Note that we adopt double Q-learning technique and then maintain two Q-functions $Q_{\theta_1}, Q_{\theta_2}$ and the corresponding target Q-functions $Q_{\bar{\theta}_1}, Q_{\bar{\theta}_2}$. As shown in Eq.37, all the next actions are sampled by the learned policy, *i.e.*, $a'_j \sim \pi_\phi$, and we estimate the KL-divergence at the next state s' by the Monte Carlo (MC) sampling, *i.e.*, $D_{\text{KL}}\left[\pi_\phi \| \pi_\beta\right](s') \simeq \mathop{\mathbb{E}}_{\{a'_j\}^N\sim\pi_\beta}\left[\frac{1}{N}\sum_{j=1}^{N}\frac{\log\pi_\phi(a'_j|s')}{\log\pi_\beta(a'_j|s')}\right]$. The extra reward shaping term (blue term in Eq.37) is the main difference from the original BRAC. After updating both Q-functions by minimizing the objective in Eq.36, we then learn the policy function by solving

Algorithm 1 **i**mplicit **T**rust **R**egion **P**olicy **O**ptimization

```
Input: Dataset $\mathcal{D} = \{s_i, a_i, r_i, s_{i+1}\}_{i=0}^{K}$.
initialize behavior policy $\pi_\beta$, learned policy network $\pi_\phi$, Q-function networks $Q_{\theta_1}, Q_{\theta_2}$ and
    target Q-function networks $Q_{\bar{\theta}_1}, Q_{\bar{\theta}_2}$.
// Behavior Policy Training
for $t = 0, 1, \cdots, M$ do
    Sample mini-batch samples $(s, a) \sim \mathcal{D}$.
    Update $\pi_\beta$ using Eq.(35).
end for
// implicit TRPO Training
for $k = 0, 1, \cdots, K$ do
    Sample mini-batch samples $(s, a, r, s') \sim \mathcal{D}$.
    Update Q-functions $Q_{\theta_1}, Q_{\theta_2}$ by Eq.(36).
    Update policy function $\pi_\phi$ by Eq.(38).
    Update the target Q-functions via $\bar{\theta}_i \leftarrow (1 - \kappa)\bar{\theta}_i + \kappa\theta$
end for
```

the following optimization problem:

$$\pi_\phi := \arg\max_{\phi} \mathop{\mathbb{E}}_{s\sim\mathcal{D}, a\sim\pi_\phi(\cdot|s)} \left[\min_{i=1,2} Q_{\theta_i}(s, a) - \alpha \cdot \mathop{\mathbb{E}}_{\{a_j\}^N \sim \pi_\beta} \left[\frac{1}{N} \sum_{j=1}^{N} \frac{\log \pi_\phi(a_j|s)}{\log \pi_\beta(a_j|s)} \right] \right] \tag{38}$$

Note we also use MC sampling to estimate the KL-divergence at the current state s. Finally, Algorithm 1 describes the practical algorithm for our iTRPO method.

B.2.2 Other Baselines

Besides our iTRPO method, we reimplement some related baselines due to their lack of the complete results on the latest '-v2' datesets.

- **IQL** [6]: We retrain IQL with our implementation, referring to its official implementation (`https://github.com/ikostrikov/implicit_q_learning`) and the specific parameter configurations (`https://github.com/tinkoff-ai/CORL`).
- **CQL** [7]: We retrain CQL using its official implementation (`https://github.com/aviralkumar2907/CQL`). Note that we finetune the critic hyperparameter $\alpha \in \{5.0, 10.0\}$ to obtain the stronger baseline.
- **TD3-BC** [3]: We retrain TD3-BC using its official implementation (`https://github.com/sfujim/TD3_BC`) and following its provided hyperparameter.

B.3 Hypermetaer Setup

Implementation details. Following Algorithm 1, there are still some critical considerations on the hyperparameter choices, especially for the parameter α that determines the strength of both behavior regularization and the one τ that balances the two different behavior regularization. According to Eq.2, if we set $\tau = 0$, our iTRPO returns to be the original BRAC method, but if $\tau \to 1$, our method turns to be a classical trust region-based policy optimization method while without any regularization that constrains the distance between π and π_β. Due to the importance of this behavior regularization for the offline RL setting, so we should avoid choosing a too large τ. Figure 1 shows that $[0.03, 0.3]$ is a reasonable range to choose a proper τ for the learning tasks. For convenience, we tune the specific τ across the candidate set $\{0.03, 0.07, 0.1\}$ for all of MuJoCo Locomotion tasks.

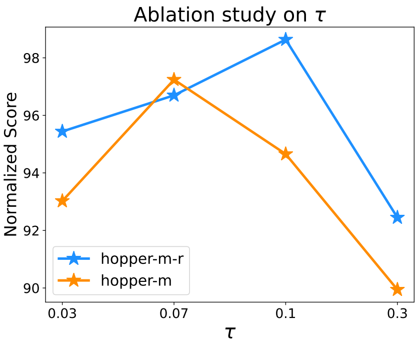

Figure 1: Ablation study on τ.

As for another α, we found that the quality of datasets has an important impact on its choice. More specifically, the high-quality datasets ('medium-expert' & 'expert') prefer a larger α, while the low-quality datasets ('random', 'medium', 'medium-replay') are more inclined to a smaller one. This observation satisfies the intuition that these high-quality datasets need more behavior regularization to mimic the (near)-expert's behavior and, in contrast, relaxing behavior regularization is beneficial to improve policy for the low-quality ones. In this paper, we choose this parameter from the candidate set $\{0.1, 0.3, 0.9\}$. We summarize the above two hyperparameters used for our experimental results in Table 2. To avoid the potential numerical issue on the shaping reward, *i.e.,* $\log \frac{\pi}{\pi_\beta}$, we use the clipping function to limit its lower bound like $\max(\log \frac{\pi}{\pi_\beta}, l_{clip})$. The detailed hyperparameters can be found in Table 3

Table 2: Specific choices of τ and α across all MuJoCo Locomotion tasks for our iTRPO method

Task Name	scaling coefficient τ	strength coefficient α
halfcheetah-random-v2	0.03	0.1
hopper-random-v2	0.1	0.1
walker2d-random-v2	0.03	0.9
halfcheetah-medium-v2	0.03	0.1
hopper-medium-v2	0.03	0.1
walker2d-medium-v2	0.1	0.3
halfcheetah-medium-replay-v2	0.03	0.1
hopper-medium-replay-v2	0.03	0.3
walker2d-medium-replay-v2	0.03	0.3
halfcheetah-medium-expert-v2	0.03	0.9
hopper-medium-expert-v2	0.07	0.9
walker2d-medium-expert-v2	0.07	0.3
halfcheetah-expert-v2	0.03	0.9
hopper-expert-v2	0.03	0.9
walker2d-expert-v2	0.07	0.9

Table 3: Hyperparameters used in our iTRPO method for MuJoCo Locomotion tasks

Parameters	Value	Description
Optimizer	Adam	Optimization algorithm used for Critic and Actor learning
K	1e6	Number of update iterations
lr_c	3e-4	Critic learning rate
lr_a	3e-4	Actor learning rate
γ	0.99	Discount factor
κ	0.005	Soft target rate for target Q-functions
N_{bs}	256	Batch size
N_{KL}	10	Number of samples for KL-divergence estimation
N_{fc}	3	AC layer size
H_{fc}	256	AC hidden dim
l_{clip}	-10	the lower bound used for clipping log-probability ratio
τ	$\{0.03, 0.07, 0.1\}$	the trade-off coefficient for two different behavior regularization
α	$\{0.1, 0.3, 0.9\}$	the strength coefficient for both behavior regularization

B.4 Learning Curves

Taking the uncertainty of environments into account, we run four seeds for all the experiments. The overall learning curves are shown in Fig.2

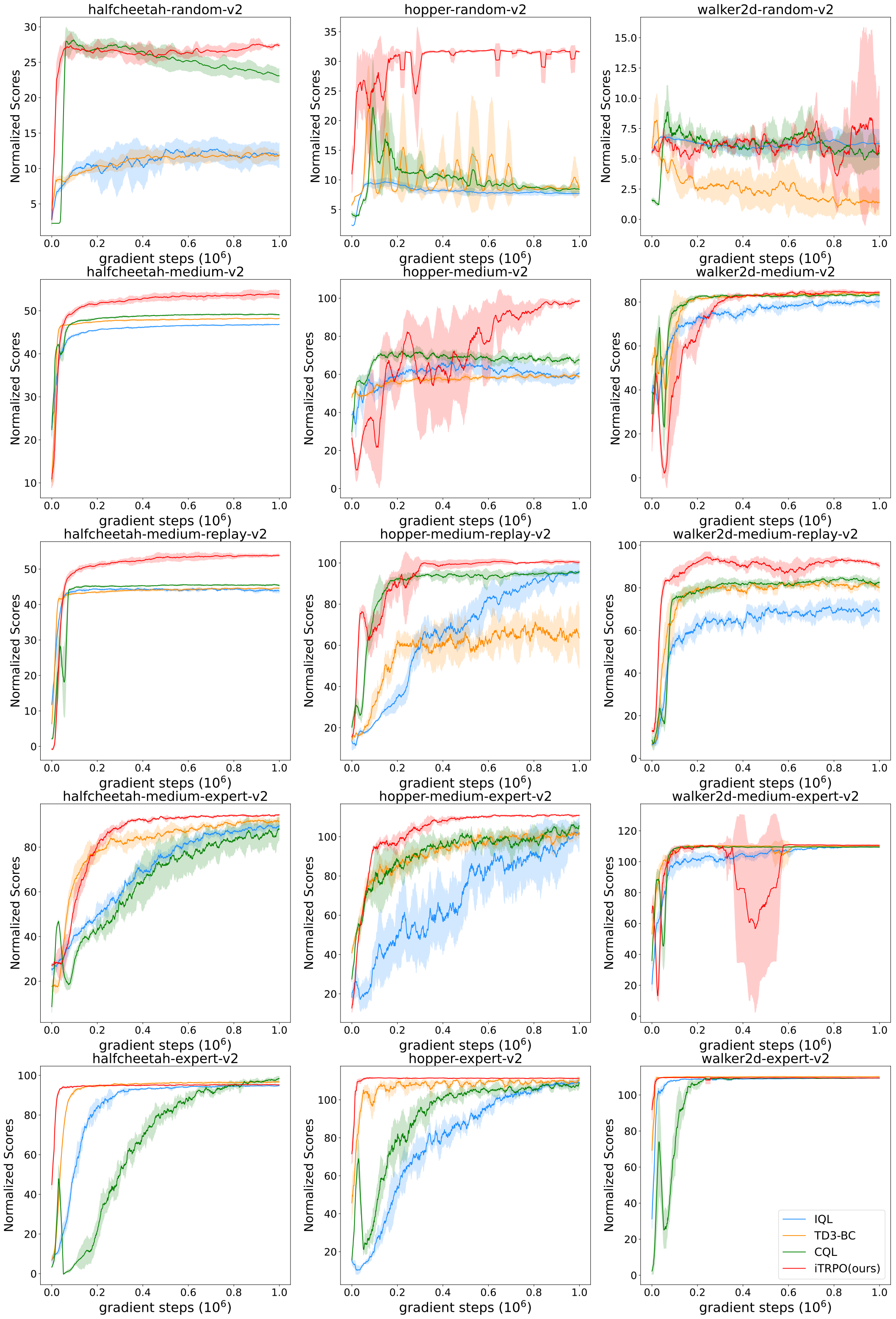

Figure 2: Learning curves of our iTRPO and some baselines across 15 MuJoCo Locomotion tasks.

C Experiments on Other Datasets

In this section, we further verify the effectiveness of the proposed iTRPO method on the more challenging Antmaze domain. These D4RL tasks require composing parts of suboptimal trajectories to form more optimal policies for reaching goals on a MuJoco Ant robot. Similarly, we conduct these experiments on the bug-fixed '-v2' datasets.

C.1 Implementation Details

Specific choices. We implement our iTRPO method following similar details used in MuJoCo Locomotion tasks. Analogously, we show only some choices with variations compared with the ones used for MuJoCo Locomotion tasks in Table 4 & 5.

Other baselines. For convenience, we compare our iTRPO method with some specific baselines whose results are shown in prior work [9]. Some different baselines are pointed out as follows:

- SPOT [9]: introduces a pluggable regularization term applied directly to the estimated behavior density (Code link: `https://github.com/thuml/SPOT`)
- PLAS [11]: Learning the policy in the latent action space (PLAS). (Code link: `https://github.com/Wenxuan-Zhou/PLAS`)

Table 4: Hyperparameters used in our iTRPO method for Antmaze tasks

Hyperparameters	Value	Description
lr_c	5e-5	Critic learning rate
γ	0.999	Discount factor
l_{clip}	-5	the lower bound used for clipping log-probability ratio
τ	0.03	the trade-off coefficient
α	$\{0.003, 0.004, 0.009\}$	the strength coefficient

Table 5: Specific choices of τ and α across some Antmaze tasks for our iTRPO method

Task Name	scaling coefficient τ	strength coefficient α
Antmaze-umaze-v2	0.03	0.009
Antmaze-umaze-diverse-v2	0.03	0.009
Antmaze-medium-play-v2	0.03	0.003
Antmaze-medium-diverse-v2	0.03	0.004
Antmaze-large-play-v2	0.03	0.006
Antmaze-large-diverse-v2	0.03	0.006

C.2 Results and Learning Curves

We have demonstrated the numerical performance in the main text, all the results are directly taken from the paper of SPOT [9]. In this part, we further demonstrate the learning curves of our iTRPO method some Antmaze tasks. Similarly, we train all these methods for $1e6$ gradient steps, and we evaluate the learned policy per 50000 iterations. The overall learning curves are shown in Fig.3

References

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. PMLR, 2017.

[2] Semih Cayci, Niao He, and Rayadurgam Srikant. Linear convergence of entropy-regularized natural policy gradient with linear function approximation. *arXiv preprint arXiv:2106.04096*, 2021.

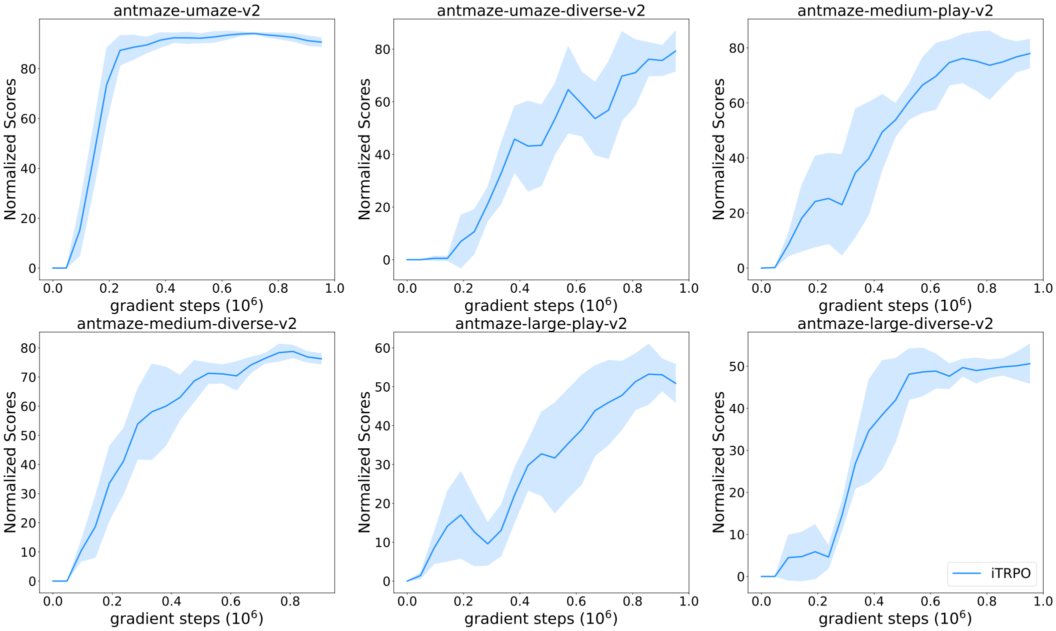

Figure 3: Learning curves of our iTRPO and some baselines across Antmaze tasks.

[3] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

[4] Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169. PMLR, 2019.

[5] Ilya Kostrikov, Rob Fergus, Jonathan Tompson, and Ofir Nachum. Offline reinforcement learning with fisher divergence critic regularization. In *International Conference on Machine Learning*, pages 5774–5783. PMLR, 2021.

[6] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 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, 2022.

[7] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191, 2020.

[8] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[9] Jialong Wu, Haixu Wu, Zihan Qiu, Jianmin Wang, and Mingsheng Long. Supported policy optimization for offline reinforcement learning. In *NeurIPS*, 2022.

[10] Wenhao Zhan, Shicong Cen, Baihe Huang, Yuxin Chen, Jason D Lee, and Yuejie Chi. Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence. *arXiv preprint arXiv:2105.11066*, 2021.

[11] Wenxuan Zhou, Sujay Bajracharya, and David Held. Plas: Latent action space for offline reinforcement learning. In *Conference on Robot Learning*, pages 1719–1735. PMLR, 2021.

[12] Zifeng Zhuang, Kun Lei, Jinxin Liu, Donglin Wang, and Yilang Guo. Behavior proximal policy optimization. *arXiv preprint arXiv:2302.11312*, 2023.