A PSEUDOCODE FOR ALGORITHMS

Algorithm 1 Conservative Offline RL Algorithm

Require: Offline dataset $\mathcal{D}$, discount factor γ, and confidence level δ

1: Compute $n(s,a)$ from $\mathcal{D}$, and estimate $\widehat{r}(s,a)$, $\widehat{P}(s'|s,a)$, $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$

2: Initialize $\widehat{Q}(s,a) \leftarrow 0, \widehat{V}(s) \leftarrow 0, \ \forall (s,a)$

3: **for** $i = 1, 2, \ldots, m$ **do**

Calculate $b(s,a)$ as:

$$b(s,a) \leftarrow \sqrt{\frac{\mathbb{V}(\widehat{P}(s,a), \widehat{V}) \log(|\mathcal{S}||\mathcal{A}|m/\delta))}{(n(s,a) \wedge 1)}} + \sqrt{\frac{\widehat{r}(s,a) \log(|\mathcal{S}||\mathcal{A}|m/\delta)}{(n(s,a) \wedge 1)}} + \frac{\log(|\mathcal{S}||\mathcal{A}|m/\delta)}{(n(s,a) \wedge 1)}$$

Calculate $\widehat{\pi}^*(s)$ as:

$$\widehat{Q}(s,a) \leftarrow \widehat{r}(s,a) - b(s,a) + \gamma \widehat{P}(s,a) \cdot \widehat{V}$$
$$\widehat{V}(s) \leftarrow \max_a \widehat{Q}(s,a)$$
$$\widehat{\pi}^*(s) \leftarrow \arg\max_a \widehat{Q}(s,a)$$

4: Return $\widehat{\pi}^*$

Algorithm 2 Policy-Constraint Offline RL Algorithm

Require: Offline dataset $\mathcal{D}$, discount factor γ, and threshold b

1: Compute $n(s,a)$ from $\mathcal{D}$, and estimate $\widehat{r}(s,a)$, $\widehat{P}(s'|s,a)$, $\widehat{\mu}(s,a)$, $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$

2: Compute $\zeta(s,a) \leftarrow 1\{\widehat{\mu}(s,a) \geq b\}$, $\forall (s,a)$

3: Initialize $\widehat{\pi}^*(a|s) \leftarrow \frac{1}{|\mathcal{A}|}$, $\widehat{Q}_\zeta^{\widehat{\pi}^*}(s,a) \leftarrow 0$, $\widehat{V}_\zeta^{\widehat{\pi}^*}(s) \leftarrow 0$, $\forall (s,a)$

4: **for** $\ell = 1, 2, \ldots, k$ **do**

5: **for** $i = 1, 2, \ldots, m$ **do**

Update $\widehat{Q}_\zeta^{\widehat{\pi}^*}(s,a), \widehat{V}_\zeta^{\widehat{\pi}^*}(s)$ as:

$$\widehat{Q}_\zeta^{\widehat{\pi}^*}(s,a) \leftarrow \widehat{r}(s,a) + \gamma \widehat{P}(s,a) \cdot \widehat{V}_\zeta^{\widehat{\pi}^*}$$
$$\widehat{V}_\zeta^{\widehat{\pi}^*}(s) \leftarrow \sum_a \widehat{\pi}^*(a|s) \zeta(s,a) \cdot \widehat{Q}(s,a)$$

Compute $\widehat{\pi}^*$ as:

$$\widehat{\pi}^* \leftarrow \arg\max_\pi \mathbb{E}_{\mathbf{s} \sim \mathcal{D}} \left[\mathbb{E}_{\mathbf{a} \sim \pi'} \left[\zeta(\mathbf{s}, \mathbf{a}) \cdot \widehat{Q}_\zeta^\pi(\mathbf{s}, \mathbf{a}) \right] \right]$$

6: Return $\widehat{\pi}^*$.

B PROOFS

B.1 PROOF OF THEOREM 4.1

Let π_β be the behavior policy that we fit our learned policy $\widehat{\pi}_\beta$ to. Recall that the BC algorithm we analyze fits $\widehat{\pi}_\beta$ to choose actions according to the empirical dataset distribution for states that appear in dataset $\mathcal{D}$, and uniformly at random otherwise. We have

$$\mathbb{E}_\mathcal{D}\left[J(\pi^*) - J(\widehat{\pi}_\beta)\right] \leq J(\pi^*) - J(\pi_\beta) + \mathbb{E}_\mathcal{D}\left[J(\pi_\beta) - J(\widehat{\pi}_\beta)\right]$$

The following lemma from Rajaraman et al. (2020) bounds the suboptimality from performing BC on a (potentially stochastic) expert, which we adapt below factoring in bounded returns of trajectories from Condition 3.2.

Lemma B.1 (Theorem 4.4, Rajaraman et al. (2020))**.** *The policy returned by BC on behavior policy π_β has expected error bounded as*

$$\mathbb{E}_\mathcal{D}\left[J(\pi_\beta) - J(\widehat{\pi}_\beta)\right] \leq \frac{SH \log N}{N},$$

where π_β could be stochastic.

Using Lemma B.1, we have $\mathbb{E}_{\mathcal{D}}\left[J(\pi_\beta) - J(\widehat{\pi}_\beta)\right] \leq \frac{SH\iota}{N}$. What remains is bounding the suboptimality of the behavior policy, which we can upper-bound as

$$\begin{aligned}
J(\pi^*) - J(\pi_\beta) &\leq \sum_{t=0}^{\infty}\sum_{s}\gamma^t \mathbb{P}\left(s_t = s\right)\mathbb{E}_{\pi_\beta(\cdot|s)}\left[1\{a \neq \pi_t^*(s)\}\right] \\
&= \frac{1}{2}\sum_{t=0}^{\infty}\sum_{s}\gamma^t d_t^*(s)\sum_{a}\left|\pi_\beta(a \mid s) - 1\{a = \pi_t^*(s)\}\right| \\
&= \frac{1}{2}\sum_{t=0}^{\infty}\sum_{(s,a)}\gamma^t \left|d_t^*(s)\pi_\beta(a \mid s) - d_t^*(s,a)\right| \\
&\leq \frac{C^* - 1}{2} H \sum_{(s,a)}\mu(s,a) \\
&= \frac{(C^* - 1)H}{2},
\end{aligned}$$

where we use the definition of C^* in Condition 3.1. Taking the sum of both terms yields the desired result.

B.2 Proof of Theorem 4.2

In this section, we proof the performance guarantee for the conservative offline RL algorithm detailed in Algorithm 1. Recall that the algorithm we consider builds upon empirical value iteration but subtracts a penalty during each Q-update. Specifically, we initialize $Q_0(s,a) = 0, V_0(s) = 0$ for all (s,a). Let $n(s,a)$ be the number of times (s,a) appeared in $\mathcal{D}$, and let $\widehat{r}(s,a)$, $\widehat{P}(s,a)$ be the empirical estimates of their reward and transition probabilities. Then, for each iteration $i \in [m]$:

$$\begin{aligned}
\widehat{Q}_i(s,a) &\leftarrow \widehat{r}(s,a) - b_i(s,a) + \gamma \widehat{P}(s,a)\cdot \widehat{V}_{i-1}, \quad \text{for all } s,a, \\
\widehat{V}_i(s) &\leftarrow \max\{\widehat{V}_{t-1}(s), \max_a \widehat{Q}_i(s,a)\}, \quad \text{for all } s,
\end{aligned}$$

In our algorithm we define the penalty function as

$$b_i(s,a) \leftarrow \sqrt{\frac{\mathbb{V}(\widehat{P}(s,a), \widehat{V}_{i-1})\iota}{(n(s,a)\wedge 1)}} + \sqrt{\frac{\widehat{r}(s,a)\iota}{(n(s,a)\wedge 1)}} + \frac{\iota}{(n(s,a)\wedge 1)},$$

where we let ι to capture all poly-logarithmic terms. As notation, we drop the subscript i to denote the final $\widehat{Q}$ and $\widehat{V}$ at iteration m, where $m = H\log N$. Finally, the learned policy $\widehat{\pi}^*$ satisfies $\widehat{\pi}^*(s) \in \arg\max_a \widehat{Q}(s,a)$ for all s, if multiple such actions exist, then the policy samples an action uniformly at random.

B.2.1 Technical Lemmas

Lemma B.2 (Bernstein's inequality)**.** *Let $X, \{X_i\}_{i=1}^n$ be i.i.d random variables with values in $[0,1]$, and let $\delta > 0$. Then we have*

$$\mathbb{P}\left(\left|\mathbb{E}\left[X\right] - \frac{1}{n}\sum_{i=1}^{n} X_i\right| > \sqrt{\frac{2\mathrm{Var}\left[X\right]\log(2/\delta)}{n}} + \frac{\log(2/\delta)}{n}\right) \leq \delta.$$

Lemma B.3 (Theorem 4, Maurer & Pontil (2009))**.** *Let $X, \{X_i\}_{i=1}^n$ with $n \geq 2$ be i.i.d random variables with values in $[0,1]$. Define $\bar{X} = \frac{1}{n}\sum_{i=1}^n X_i$ and $\widehat{\mathrm{Var}}(X) = \frac{1}{n}\sum_{i=1}^n (X_i - \bar{X})^2$. Let $\delta > 0$. Then we have*

$$\mathbb{P}\left(\left|\mathbb{E}\left[X\right] - \frac{1}{n}\sum_{i=1}^{n} X_i\right| > \sqrt{\frac{2\widehat{\mathrm{Var}}(\bar{X})\log(2/\delta)}{n-1}} + \frac{7\log(2/\delta)}{3(n-1)}\right) \leq \delta.$$

Lemma B.4 (Lemma 4, Ren et al. (2021)). *Let $\lambda_1, \lambda_2 > 0$ be constants. Let $f : \mathbb{Z}_{\geq 0} \to \mathbb{R}$ be a function such that $f(i) \leq H$, $\forall i$ and $f(i)$ satisfies the recursion*

$$f(i) \leq \sqrt{\lambda_1 f(i+1)} + \lambda_1 + 2^{i+1}\lambda_2 \,.$$

Then, we have that $f(0) \leq 6(\lambda_1 + \lambda_2)$.

B.2.2 Pessimism Guarantee

The first thing we want to show is that with high probability, the algorithm provides pessimistic value estimates, namely that $\widehat{V}_i(s) \leq V^*(s)$ for all $t \in [T]$ and $s \in \mathcal{S}$. To do so, we introduce a notion of a "good" event, which occurs when our empirical estimates of the MDP are not far from the true MDP. We define $\mathcal{E}_1$ to be the event where

$$\left|(\widehat{P}(s,a) - P(s,a)) \cdot \widehat{V}_i\right| \leq \sqrt{\frac{\mathbb{V}(\widehat{P}(s,a), \widehat{V}_i)\iota}{(n(s,a) \wedge 1)}} + \frac{\iota}{(n(s,a) \wedge 1)} \tag{1}$$

holds for all $i \in [m]$ and $(s,a) \in \mathcal{S} \times \mathcal{A}$. We also define $\mathcal{E}_2$ to be the event where

$$|\widehat{r}(s,a) - r(s,a)| \leq \sqrt{\frac{\widehat{r}(s,a)\iota}{(n(s,a) \wedge 1)}} + \frac{\iota}{(n(s,a) \wedge 1)} \tag{2}$$

holds for all (s,a).

We want to show that the good event $\mathcal{E} = \mathcal{E}_1 \cap \mathcal{E}_2$ occurs with high probability. The proof mostly follows from Bernstein's inequality in Lemma B.2 . Note that because $\widehat{P}(s,a), \widehat{V}_i$ are not independent, we cannot straightforwardly apply Bernstein's inequality. We instead use the approach of Agarwal et al. (2020a) who, for each state s, partition the range of $\widehat{V}_i(s)$ within a modified s-absorbing MDP to create independence from $\widehat{P}$. The following lemma from Agarwal et al. (2020a) is a result of such analysis, and is slightly modified below to account for bounded returns of trajectories, *i.e.*, $\widehat{V}_i(s) \leq 1$:

Lemma B.5 (Lemma 9, Agarwal et al. (2020a)). *For any iteration t, state-action $(s,a) \in \mathcal{S} \times \mathcal{A}$ such that $n(s,a) \geq 1$, and $\delta > 0$, we have*

$$\mathbb{P}\left(\left|(\widehat{P}(s,a) - P(s,a)) \cdot \widehat{V}_i\right| > \sqrt{\frac{\mathbb{V}(\widehat{P}(s,a), \widehat{V}_i)\iota}{n(s,a)}} + \frac{\iota}{n(s,a)}\right) \leq \delta \,.$$

Using this, we can show that $\mathcal{E}$ occurs with high probability:

Lemma B.6. $\mathbb{P}(\mathcal{E}) \geq 1 - 2|\mathcal{S}||\mathcal{A}|m\delta$.

Proof. For each i and (s,a), if $n(s,a) \leq 1$, then equation 1 and equation 2 hold trivially. For $n(s,a) \geq 2$, we have from Lemma B.5 that

$$\mathbb{P}\left(\left|(\widehat{P}(s,a) - P(s,a)) \cdot \widehat{V}_i\right| > \sqrt{\frac{\mathbb{V}(\widehat{P}(s,a), \widehat{V}_i)\iota}{n(s,a)}} + \frac{\iota}{n(s,a)}\right) \leq \delta \,.$$

Similarly, we can use Lemma B.3 to derive

$$\begin{aligned}
&\mathbb{P}\left(|\widehat{r}(s,a) - r(s,a)| > \sqrt{\frac{\widehat{r}(s,a)\iota}{n(s,a)}} + \frac{\iota}{n(s,a)}\right) \\
&\leq \mathbb{P}\left(|\widehat{r}(s,a) - r(s,a)| > \sqrt{\frac{\widehat{\mathrm{Var}}(\widehat{r}(s,a))\iota}{2(n(s,a)-1)}} + \frac{\iota}{2(n(s,a)-1)}\right) \leq \delta \,,
\end{aligned}$$

where we use that $\widehat{\mathrm{Var}}(\widehat{r}(s,a)) \leq \widehat{r}(s,a)$ for $[0,1]$ rewards, and with slight abuse of notation, let ι capture all constant factors. Taking the union bound over all i and (s,a) yields the desired result. □

Now, we can prove that our value estimates are indeed pessimistic.

Lemma B.7 (Pessimism Guarantee)**.** *On event $\mathcal{E}$, we have that $\widehat{V}_i(s) \le V^{\widehat{\pi}^*}(s) \le V^*(s)$ for any iteration $i \in [m]$ and state $s \in \mathcal{S}$.*

Proof. We aim to prove the following for any i and s: $\widehat{V}_{i-1}(s) \le \widehat{V}_i(s) \le V^{\widehat{\pi}^*}(s) \le V^*(s)$. We prove the claims one by one.

$\widehat{V}_{i-1}(s) \le \widehat{V}_i(s)$: This is directly implied by the monotonic update of our algorithm.

$\widehat{V}_i(s) \le V^{\widehat{\pi}^*}(s)$: We will prove this via induction. We have that this holds for $\widehat{V}_0$ trivially. Assume it holds for $t-1$, then we have

$$\begin{aligned} V^{\widehat{\pi}^*}(s) &\ge \mathbb{E}_{a\sim\widehat{\pi}^*(\cdot|s)}\left[r(s,a) + \gamma P(s,a)\cdot \widehat{V}_{i-1}\right] \\ &\ge \mathbb{E}_a\left[\widehat{r}(s,a) - b_i(s,a) + \gamma \widehat{P}(s,a)\cdot \widehat{V}_{t-1}\right] + \\ &\quad \mathbb{E}_a\left[b_i(s,a) - (\widehat{r}(s,a) - r(s,a)) - \gamma(\widehat{P}(s,a) - P(s,a))\cdot \widehat{V}_{i-1}\right] \\ &\ge \widehat{V}_t(s)\,, \end{aligned}$$

where we use that

$$\begin{aligned} b_i(s,a) &= \sqrt{\frac{\mathbb{V}(\widehat{P}(s,a), \widehat{V}_{i-1})\iota}{(n(s,a)\wedge 1)}} + \sqrt{\frac{\widehat{r}(s,a)\iota}{(n(s,a)\wedge 1)}} + \frac{\iota}{(n(s,a)\wedge 1)} \\ &\ge (\widehat{r}(s,a) - r(s,a)) + \gamma(\widehat{P}(s,a) - P(s,a))\cdot \widehat{V}_{i-1} \end{aligned}$$

under event $\mathcal{E}$.

Finally, the claim of $V^{\widehat{\pi}^*}(s) \le V^*(s)$ is trivial, which completes the proof of our pessimism guarantee. □

B.2.3 Value Difference Lemma

Now, we are ready to derive the performance guarantee from Theorem 4.2. The following lemma is a bound on the estimation error of our pessimistic Q-values.

Lemma B.8. *On event $\mathcal{E}$, the following holds for any $i \in [m]$ and $(s,a) \in \mathcal{S}\times\mathcal{A}$:*

$$Q^*(s,a) - \widehat{Q}_i(s,a) \le \gamma P(s,a)\cdot(Q^*(\cdot;\pi^*) - \widehat{Q}_{i-1}(\cdot;\pi^*)) + 2b_i(s,a)\,, \tag{3}$$

where $f(\cdot;\pi)$ satisfies $f(s;\pi) = \sum_a \pi(a\mid s) f(s,a)$.

Proof. We have,

$$\begin{aligned} &Q^*(s,a) - \widehat{Q}_i(s,a) \\ &\quad = r(s,a) + \gamma P(s,a)\cdot V^* - (\widehat{r}(s,a) - b_i(s,a) + \gamma\widehat{P}(s,a)\cdot\widehat{V}_{t-1}) \\ &\quad = b_i(s,a) + r(s,a) - \widehat{r}(s,a) + \gamma P(s,a)\cdot(V^* - \widehat{V}_{t-1}) + \gamma(P(s,a) - \widehat{P}(s,a))\cdot\widehat{V}_{t-1} \\ &\quad \le \gamma P(s,a)\cdot(V^* - \widehat{V}_{t-1}) + 2b_i(s,a) \\ &\quad \le \gamma P(s,a)\cdot(Q^*(\cdot;\pi^*) - \widehat{Q}_{t-1}(\cdot;\pi^*)) + 2b_i(s,a)\,. \end{aligned}$$

The first inequality is due by definition of $\mathcal{E}$ and the second is because $\widehat{V}_{t-1} \ge \max_a \widehat{Q}_{t-1}(\cdot,a) \ge \widehat{Q}_i(\cdot,\pi^*)$. □

By recursively applying Lemma B.8, we can derive the following value difference lemma:

Lemma B.9 (Value Difference Lemma)**.** *On event $\mathcal{E}$, at any iteration $i \in [m]$, we have*

$$J(\pi^*) - J(\widehat{\pi}^*) \le \gamma^i + 2\sum_{t=1}^{i}\sum_{(s,a)} \gamma^{i-t} d^*_{i-t}(s,a) b_t(s,a)\,, \tag{4}$$

where $d^*_t(s,a) = \mathbb{P}(s_t = s, a_t = a; \pi^*)$.

Proof. We have,

$$J(\pi^*) - J(\widehat{\pi}^*) = \mathbb{E}_\rho\left[V^*(s) - V^{\widehat{\pi}^*}(s)\right] \le \mathbb{E}_\rho\left[V^*(s) - \widehat{V}_i(s)\right] \le \rho(Q^*(\cdot;\pi^*) - \widehat{Q}_i(\cdot;\pi^*))$$

where we use Lemma B.7 in the first inequality. As shorthand, let $P^\pi \in \mathbb{R}^{(\mathcal{S}\times\mathcal{A})\times(\mathcal{S}\times\mathcal{A})}$ where $P^\pi(s,a,s',a') = P(s'|s,a)\pi(a'|s')$ be the transition matrix for policy π. Now, we can apply Lemma B.8 recursively to derive

$$\begin{aligned}
\rho^{\pi^*}(Q^* - \widehat{Q}_i) &\le \rho^{\pi^*}\left(\gamma P^{\pi^*}(Q - \widehat{Q}_{i-1}) + 2b_i\right) \\
&\le \rho^{\pi^*}\left(\gamma P^{\pi^*}\left(\gamma P^{\pi^*}(Q^* - \widehat{Q}_{i-2}) + 2b_{i-1}\right) + 2b_i\right) \\
&\le \dots \\
&\le \rho^{\pi^*}(\gamma P^{\pi^*})^i(Q^* - \widehat{Q}_0) + 2\sum_{t=1}^{i}\rho^{\pi^*}(\gamma P^{\pi^*})^{i-t}b_t \\
&\le \gamma^i \mathbf{1} + 2\sum_{t=1}^{i}\gamma^{i-t}d^*_{i-t}b_t
\end{aligned}$$

where we use that $d^*_t = \rho^{\pi^*}(P^{\pi^*})^t$. This yields the desired result. □

Now, we are ready to bound the desired quantity $\mathsf{SubOpt}(\widehat{\pi}^{**}) = \mathbb{E}_\mathcal{D}\left[J(\pi^*) - J(\widehat{\pi}^*)\right]$. We have

$$\begin{aligned}
\mathbb{E}_\mathcal{D}\left[J(\pi^*) - J(\widehat{\pi}^*)\right] &= \mathbb{E}_\mathcal{D}\left[\sum_s \rho(s)(V^*(s) - V^{\widehat{\pi}^*}(s))\right] \\
&= \underbrace{\mathbb{E}_\mathcal{D}\left[1\{\bar{\mathcal{E}}\}\sum_s \rho(s)(V^*(s) - V^{\widehat{\pi}^*}(s))\right]}_{:=\Delta_1} \\
&\quad + \underbrace{\mathbb{E}_\mathcal{D}\left[1\{\exists s \in \mathcal{S},\ n(s,\pi^*(s)) = 0\}\sum_s \rho(s)(V^*(s) - V^{\widehat{\pi}^*}(s))\right]}_{:=\Delta_2} \\
&\quad + \underbrace{\mathbb{E}_\mathcal{D}\left[1\{\forall s \in \mathcal{S},\ n(s,\pi^*(s)) > 0\}\,1\{\mathcal{E}\}\sum_s \rho(s)(V^*(s) - V^{\widehat{\pi}^*}(s))\right]}_{:=\Delta_3}.
\end{aligned} \tag{5}$$

We bound each term individually. The first is bounded as $\Delta_1 \le \mathbb{P}\left(\bar{\mathcal{E}}\right) \le 2|\mathcal{S}||\mathcal{A}|m\delta \le \frac{\iota}{N}$ for choice of $\delta = \frac{1}{2|\mathcal{S}||\mathcal{A}|HN}$.

B.2.4 Bound on Δ_2

For the second term, we have

$$\begin{aligned}
\Delta_2 &\le \sum_s \rho(s)\mathbb{E}_\mathcal{D}\left[1\{n(s,\pi^*(s)) = 0\}\right] \\
&\le H\sum_s d^*(s,\pi^*(s))\mathbb{E}_\mathcal{D}\left[1\{n(s,\pi^*(s)) = 0\}\right] \\
&\le C^* H\sum_s \mu(s,\pi^*(s))(1-\mu(s,\pi^*(s)))^N \\
&\le \frac{4C^*|\mathcal{S}|H}{9N},
\end{aligned}$$

where we use that $\rho(s) \le Hd^*(s,\pi^*(s))$, and that $\max_{p\in[0,1]} p(1-p)^N \le \frac{4}{9N}$.

B.2.5 BOUND ON Δ_3

What remains is bounding the last term, which we know from Lemma B.9 is bounded by

$$\Delta_3 \leq \frac{1}{N} + 2\mathbb{E}_{\mathcal{D}}\left[1\{\forall s \in \mathcal{S},\ n(s,\pi^*(s)) > 0\} \sum_{t=0}^{m} \sum_{(s,a)} \gamma^{m-t} d^*_{m-t}(s,a) b_t(s,a)\right],$$

where we use that $\gamma^m \leq \frac{1}{N}$ for $m = H \log N$. Recall that $b_t(s,a)$ is given by

$$b_t(s,a) = \sqrt{\frac{\mathbb{V}(\widehat{P}(s,a), \widehat{V}_{t-1})\iota}{n(s,a)}} + \sqrt{\frac{\widehat{r}(s,a)\iota}{n(s,a)}} + \frac{\iota}{n(s,a)}$$

We can bound the summation of each term separately. For the third term we have,

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[\sum_{t=1}^{m} \sum_{(s,a)} \gamma^{m-t} d^*_{m-t}(s,a)\ \frac{\iota}{n(s,a)}\right] &\leq \sum_{t=0}^{m-1} \sum_{(s,a)} \gamma^t d^*_t(s,a) \mathbb{E}_{\mathcal{D}}\left[\frac{\iota}{n(s,a)}\right] \\
&\leq \sum_{s} \sum_{t=0}^{\infty} \gamma^t d^*_t(s,\pi^*(s))\ \frac{\iota}{N\mu(s,\pi^*(s))} \\
&\leq \frac{H\iota}{N} \sum_{s} \left((1-\gamma)\sum_{t=0}^{\infty} \gamma^t d^*_t(s,\pi^*(s))\right)\ \frac{1}{\mu(s,\pi^*_h(s))} \\
&\leq \frac{C^*|\mathcal{S}|H\iota}{N}.
\end{aligned}$$

Here we use Jensen's inequality and that $(1-\gamma)\sum_{t=1}^{\infty} \gamma^t d^*_t(s,a) \leq C^* \mu(s,a)$ for any (s,a). For the second term, we similarly have

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\left[\sum_{t=1}^{m} \sum_{(s,a)} \gamma^{m-t} d^*_{m-t}(s,a)\ \sqrt{\frac{\widehat{r}(s,a)\iota}{n(s,a)}}\right] \\
&\leq \mathbb{E}_{\mathcal{D}}\left[\sqrt{\sum_{t=1}^{m} \sum_{(s,a)} \gamma^{m-t} d^*_{m-t}(s,a) \frac{\iota}{n(s,a)}}\right]\ \sqrt{\sum_{t=1}^{m} \sum_{(s,a)} \gamma^{m-t} d^*_{m-t}(s,a)\widehat{r}(s,a)} \\
&\leq \sqrt{\frac{C^*|\mathcal{S}|H\iota}{N}},
\end{aligned}$$

where we use Cauchy-Schwarz, then Condition 3.2 to bound the total estimated reward. Finally, we consider the first term of $b_t(s,a)$

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\left[\sum_{t=1}^{m} \sum_{(s,a)} \gamma^{m-t} d^*_{m-t}(s,a)\ \sqrt{\frac{\mathbb{V}(\widehat{P}(s,a), \widehat{V}_{t-1})\iota}{n(s,a)}}\right] \\
&\leq \mathbb{E}_{\mathcal{D}}\left[\sqrt{\sum_{t=1}^{m} \sum_{(s,a)} \gamma^{m-t} d^*_{m-t}(s,a) \frac{\iota}{n(s,a)}}\right]\ \sqrt{\sum_{t=1}^{m} \sum_{(s,a)} \gamma^{m-t} d^*_{m-t}(s,a)\mathbb{V}(\widehat{P}(s,a), \widehat{V}_{t-1})} \\
&\leq \sqrt{\frac{C^*|\mathcal{S}|H\iota}{N}}\ \sqrt{\sum_{t=1}^{m} \sum_{(s,a)} \gamma^{m-t} d^*_{m-t}(s,a)\mathbb{V}(\widehat{P}(s,a), \widehat{V}_{t-1})}.
\end{aligned}$$

Similar to what was done in Zhang et al. (2020); Ren et al. (2021) for finite-horizon MDPs, we can bound this term using variance recursion for infinite-horizon ones. Define

$$f(i) := \sum_{t=1}^{\infty} \sum_{(s,a)} \gamma^{m-t} d^*_{m-t}(s,a)\mathbb{V}(\widehat{P}(s,a), (\widehat{V}_{t-1})^{2^i}). \tag{6}$$

Using Lemma 3 of Ren et al. (2021) for the infinite-horizon case, we have the following recursion:

$$f(i) \leq \sqrt{\frac{C^*|\mathcal{S}|H\iota}{N} f(i+1)} + \frac{C^*|\mathcal{S}|H\iota}{N} + 2^{i+1}(\Phi + 1)\,,$$

where

$$\Phi := \sqrt{\frac{C^*|\mathcal{S}|H\iota}{N}} \sqrt{\sum_{t=1}^{m} \sum_{(s,a)} \gamma^{m-t} d^*_{m-t}(s,a) \mathbb{V}(\widehat{P}(s,a), \widehat{V}_{t-1})} + \frac{C^*|\mathcal{S}|H\iota}{N} \tag{7}$$

Using Lemma B.4, we can bound $f(0) = \mathcal{O}\left(\frac{C^*|\mathcal{S}|H\iota}{N} + \Phi + 1\right)$. Using that for constant c,

$$\begin{aligned}
\Phi &= \sqrt{\frac{C^*|\mathcal{S}|H\iota}{N} f(0)} + \frac{C^*|\mathcal{S}|H\iota}{N} \\
&\leq \sqrt{\frac{C^*|\mathcal{S}|H\iota}{N} \left(\frac{cC^*|\mathcal{S}|H\iota}{N} + c\Phi + c\right)} + \frac{C^*|\mathcal{S}|H\iota}{N} \\
&\leq \frac{c\Phi}{2} + \frac{2cC^*|\mathcal{S}|H\iota}{N} + \frac{c}{2}
\end{aligned}$$

we have that

$$\Phi \leq c + \frac{4cC^*|\mathcal{S}|H\iota}{N}\,.$$

Substituting this back into the inequality for Φ yields,

$$\Phi = \mathcal{O}\left(\sqrt{\frac{C^*|\mathcal{S}|H\iota}{N}} + \frac{C^*|\mathcal{S}|H\iota}{N}\right)$$

Finally, we can bound

$$\Delta_3 \leq \sqrt{\frac{C^*|\mathcal{S}|H\iota}{N}} + \frac{C^*|\mathcal{S}|H\iota}{N}\,.$$

Combining the bounds for the three terms yields the desired result.

B.3 Proof of Corollary 4.1

The proof of Corollary 4.1 mostly relies on the existing machinery in Appendix B.2. Recall that $\mathcal{C}$ is the set of critical states, and from Definition 4.1, that all $s \in \mathcal{S} \setminus \mathcal{C}$ satisfy having negligible advantage, *i.e.*, $Q^*(s, \pi^*(s)) - Q^*(s,a) \leq \varepsilon/H$ for any suboptimal action a.

The main difference between this proof and the one for Theorem 4.2 is in the derivation of the value difference lemma. Namely, by using the critical states structure, we have the following decomposition for suboptimality

$$\begin{aligned}
J(\pi^*) - J(\widehat{\pi}^*) &\leq \mathbb{E}\left[V^*(s) - V^{\widehat{\pi}^*}(s)\right] \\
&\leq \sum_{t=0}^{\infty} \gamma^t \sum_{s \in \mathcal{S}} d_t^{\widehat{\pi}^*}(s) \mathbb{E}\left[Q^*(s;\pi^*) - Q^*(s;\widehat{\pi}^*)\right] \\
&\leq \sum_{t=0}^{\infty} \gamma^t \sum_{s \in \mathcal{C}} d_t^{\widehat{\pi}^*}(s) \mathbb{E}\left[1\{s_\ell \in \mathcal{C},\, \forall \ell < t\}\left(Q^*(s;\pi^*) - Q^*(s;\widehat{\pi}^*)\right)\right] + \\
&\quad \sum_{t=0}^{\infty} \gamma^t d_t^{\widehat{\pi}^*}(s)\, \frac{\varepsilon}{H} \\
&\leq J_{\mathcal{C}}(\pi^*) - J_{\mathcal{C}}(\widehat{\pi}^*) + \varepsilon\,.
\end{aligned}$$

Here, we use the performance difference lemma, and define $J_{\mathcal{C}}(\pi)$ using an expectation over trajectories that only contain critical states.

This has an alternative interpretation. The suboptimality of $\widehat{\pi}^*$ over the true MDP is bounded as the suboptimality of the policy over a modified MDP consisting of only states in $\mathcal{C}$, and an additional constant ε. The modified MDP can be interpreted as aggregating all non-critical states into a single absorbing state.

Since Theorem 4.2 holds for any MDP, we can use it to bound

$$J_{\mathcal{C}}(\pi^*) - J_{\mathcal{C}}(\widehat{\pi}^*) \leq \sqrt{\frac{C^*|\mathcal{C}|H\iota}{N}} + \frac{C^*|\mathcal{C}|H\iota}{N}$$

Combining the above result with Condition 4.1 which bounds $|\mathcal{C}| \leq p_c|\mathcal{S}|$, completes the proof of the Corollary.

B.4 Proof of Corollary 4.2

The proof of Corollary 4.2 is a slight modification of the one for Theorem 4.2. For brevity, we will point out the parts of the proof that change, and defer the rest of the proof of Appendix B.2 Recall the decomposition for suboptimality in equation 5, which we restate below:

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[J(\pi^*) - J(\widehat{\pi}^*)\right] = &\underbrace{\mathbb{E}_{\mathcal{D}}\left[1\{\bar{\mathcal{E}}\}\sum_s \rho(s)(V^*(s) - V^{\widehat{\pi}^*}(s))\right]}_{\Delta_1} \\ &+ \underbrace{\mathbb{E}_{\mathcal{D}}\left[1\{\exists s \in \mathcal{S},\ n(s,\pi^*(s)) = 0\}\sum_s \rho(s)(V^*(s) - V^{\widehat{\pi}^*}(s))\right]}_{\Delta_2} \\ &+ \underbrace{\mathbb{E}_{\mathcal{D}}\left[1\{\forall s \in \mathcal{S},\ n(s,\pi^*(s)) > 0\}\,1\{\mathcal{E}\}\sum_s \rho(s)(V^*(s) - V^{\widehat{\pi}^*}(s))\right]}_{\Delta_3}.\end{aligned}$$

Δ_1 is bounded by $\frac{\iota}{N}$ as before.

B.4.1 Bound on Δ_2

The bound for Δ_2 changes slightly from Appendix B.2.4 due to accounting for the lower-bound on $\mu(\mathbf{s},\mathbf{a}) \geq b \geq \frac{\log H}{N}$. We have

$$\begin{aligned}\Delta_2 &\leq \sum_{\mathbf{s}} \rho(\mathbf{s})\mathbb{E}_{\mathcal{D}}\left[1\{n(\mathbf{s},\pi^*(\mathbf{s})) = 0\}\right] \\ &\leq H\sum_{\mathbf{s}} d^*(\mathbf{s},\pi^*(\mathbf{s}))\mathbb{E}_{\mathcal{D}}\left[1\{n(\mathbf{s},\pi^*(\mathbf{s})) = 0\}\right] \\ &\leq H\sum_{\mathbf{s}} d^*(\mathbf{s},\pi^*(\mathbf{s}))1\left\{d^*(\mathbf{s},\pi^*(\mathbf{s})) \leq \frac{b}{H}\right\} + H\sum_{\mathbf{s}} d^*(\mathbf{s},\pi^*(\mathbf{s}))\mathbb{E}_{\mathcal{D}}\left[1\{n(\mathbf{s},\pi^*(\mathbf{s})) = 0\}\right] \\ &\leq |\mathcal{S}|c + C^*H\sum_{\mathbf{s}} \mu(\mathbf{s},\pi^*(\mathbf{s}))(1-\mu(\mathbf{s},\pi^*(\mathbf{s})))^N \\ &\leq |\mathcal{S}|b + \frac{C^*|\mathcal{S}|\iota}{N},\end{aligned}$$

where we use that $\rho(\mathbf{s}) \leq Hd^*(\mathbf{s},\pi^*(\mathbf{s}))$, and that

$$\max_{p\in[\frac{\log H}{N},1]} p(1-p)^N \leq \frac{\log H}{N}\left(1 - \frac{\log H}{N}\right)^N \leq \frac{\log H}{HN}.$$

B.4.2 BOUND ON Δ_3

Due to the lower bound on $\mu(\mathbf{s}, \mathbf{a}) \geq b$, we can instead bound,

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[\sum_{t=1}^{m}\sum_{(s,a)}\gamma^{m-t}d^*_{m-t}(s,a)\,\frac{\iota}{n(s,a)}\right] &\leq \sum_{t=0}^{m-1}\sum_{(s,a)}\gamma^t d^*_t(s,a)\mathbb{E}_{\mathcal{D}}\left[\frac{\iota}{n(s,a)}\right] \\
&\leq \sum_{s}\sum_{t=0}^{\infty}\gamma^t d^*_t(s,\pi^*(s))\,\frac{\iota}{N\mu(s,\pi^*(s))} \\
&\leq 1\left\{d^*(\mathbf{s},\mathbf{a}) \leq \frac{b}{H}\right\} H \sum_{s}\left((1-\gamma)\sum_{t=0}^{\infty}\gamma^t d^*_t(s,\pi^*(s))\right) + \\
&\quad \frac{H\iota}{Nc}\sum_{s}\left((1-\gamma)\sum_{t=0}^{\infty}\gamma^t d^*_t(s,\pi^*(s))\right) \\
&\leq b + \frac{H\iota}{bN}.
\end{aligned}$$

The analysis for bounding Δ_3 proceeds exactly as in Appendix B.2.5 but using the new bound. Namely, we end up with the recursion

$$f(i) \leq \sqrt{\frac{H\iota}{bN} + b}\,\sqrt{f(i+1)} + \frac{H\iota}{bN} + b + 2^{i+1}(\Phi + 1)\,,$$

where

$$\Phi := \sqrt{\frac{H\iota}{bN} + b}\,\sqrt{\sum_{t=1}^{m}\sum_{(s,a)}\gamma^{m-t}d^*_{m-t}(s,a)\mathbb{V}(\widehat{P}(s,a),\widehat{V}_{t-1})} + \frac{H\iota}{Nb} + b\,.$$

Using Lemma B.4 and proceeding as in Appendix B.2.5 yields the bound

$$\Delta_3 \leq \sqrt{\frac{H\iota}{bN}} + \frac{H\iota}{bN} + \sqrt{b\iota}\,.$$

Combining the new bounds for Δ_2, Δ_3 results in the bound in the Corollary 4.2.

B.5 PROOF OF THEOREM 4.4

The proof of Theorem 4.4 builds on analysis by Agarwal et al. (2021a) that we apply to policies with a softmax parameterization, which we define below.

Definition B.1 (Softmax parameterization)**.** *For a given* $\theta \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$, $\pi_\theta(\mathbf{a}|\mathbf{s}) = \frac{\exp(\theta_{\mathbf{s},\mathbf{a}})}{\sum_{\mathbf{a}'}\exp(\theta_{\mathbf{s},\mathbf{a}'})}$.

We consider generalized BC algorithms that perform advantage-weighted policy improvement for k improvement steps. A BC algorithm with k-step policy improvement is defined as follows:

Definition B.2 (BC with k-step policy improvement)**.** *Let* $\widehat{A}^k(\mathbf{s},\mathbf{a})$ *denote the advantage of action* $\mathbf{a}$ *at state* $\mathbf{s}$ *under a given policy* $\widehat{\pi}_k$*, where the policy* $\widehat{\pi}^k(\mathbf{a}|\mathbf{s})$ *is defined via the recursion:*

$$\widehat{\pi}^{k+1}(\mathbf{a}|\mathbf{s}) := \widehat{\pi}^k(\mathbf{a}|\mathbf{s})\frac{\exp(\eta H\widehat{A}^k(\mathbf{s},\mathbf{a}))}{\mathbb{Z}_k(\mathbf{s})},$$

starting from $\widehat{\pi}^0(\mathbf{a}|\mathbf{s}) = \widehat{\pi}_\beta$*. Then, BC with* k*-step policy improvement returns* $\widehat{\pi}^k$*.*

This advantage weighted update is utilized in practical works such as Brandfonbrener et al. (2021), which first estimates the Q-function of the behavior policy using the offline dataset, i.e, $\widehat{Q}^0(\mathbf{s},\mathbf{a})$, and then computes $\widehat{\pi}^1$ as the final policy returned by the algorithm. To understand the performance difference between multiple values of k, we first utilize essentially Lemma 5 from Agarwal et al. (2021a), which we present below for completeness:

Lemma B.10 (Lower bound on policy improvement in the empirical MDP, $\widehat{M}$)**.** *The iterates $\widehat{\pi}^k$ generated by k-steps of policy improvement, for any initial state distributions $\rho_0(\mathbf{s})$ satisfy the following lower-bound on improvement:*

$$\widehat{J}(\widehat{\pi}^{k+1}) - \widehat{J}(\widehat{\pi}^k) := \mathbb{E}_{\mathbf{s}_0 \sim \rho_0}\left[\widehat{V}^{\widehat{\pi}_{k+1}}(\mathbf{s}_0)\right] - \mathbb{E}_{\mathbf{s}_0 \sim \rho_0}\left[\widehat{V}^{\widehat{\pi}_k}(\mathbf{s}_0)\right] \geq \frac{1}{\eta H}\mathbb{E}_{\mathbf{s}_0 \sim \rho_0} \log \mathbb{Z}_t(\mathbf{s}_0). \quad (8)$$

Proof. We utilize the performance difference lemma in the empirical MDP to show this:

$$\begin{aligned}
\widehat{J}(\widehat{\pi}^{k+1}) - \widehat{J}(\widehat{\pi}^k) &= H\mathbb{E}_{\mathbf{s}\sim d^{\widehat{\pi}_{k+1}}}\left[\sum_{\mathbf{a}} \widehat{\pi}_{k+1}(\mathbf{a}|\mathbf{s})\widehat{A}^k(\mathbf{s},\mathbf{a})\right] \\
&= \frac{1}{\eta}\mathbb{E}_{\mathbf{s}\sim d^{\widehat{\pi}_{k+1}}}\left[\sum_{\mathbf{a}} \widehat{\pi}^{k+1}(\mathbf{a}|\mathbf{s}) \log \frac{\widehat{\pi}^{k+1}(\mathbf{a}|\mathbf{s})\mathbb{Z}_k(\mathbf{s})}{\widehat{\pi}^k(\mathbf{a}|\mathbf{s})}\right] \\
&= \frac{1}{\eta}\mathbb{E}_{\mathbf{s}\sim d^{\widehat{\pi}_{k+1}}}\left[\mathrm{D}_{\mathrm{KL}}(\widehat{\pi}^{k+1}(\cdot|\mathbf{s})||\widehat{\pi}^k(\cdot|\mathbf{s}))\right] + \frac{1}{\eta}\mathbb{E}_{\mathbf{s}\sim d^{\widehat{\pi}_{k+1}}}\left[\log \mathbb{Z}_k(\mathbf{s})\right] \\
&\geq \frac{1}{\eta}\mathbb{E}_{\mathbf{s}\sim d^{\widehat{\pi}_{k+1}}}\left[\log \mathbb{Z}_k(\mathbf{s})\right].
\end{aligned}$$

Finally, note that the final term $\log \mathbb{Z}_t(\mathbf{s})$ is always positive because of Jensen's inequality, and the fact that the expected advantage under a given policy is 0 for any MDP. □

Utilizing Lemma B.10, we can then lower bound the total improvement of the learned policy in the actual MDP as:

$$\begin{aligned}
J(\widehat{\pi}^k) - J(\widehat{\pi}^l) &\geq \underbrace{J(\widehat{\pi}^k) - \widehat{J}(\widehat{\pi}^k)}_{\text{(a)}} + \underbrace{\widehat{J}(\widehat{\pi}^k) - \widehat{J}(\widehat{\pi}^l)}_{\text{(b)}} - \underbrace{J(\widehat{\pi}^l) - \widehat{J}(\widehat{\pi}^l)}_{\text{(c)}} \\
&\geq \frac{1}{\eta}\sum_{j=l}^{k} \mathbb{E}_{\mathbf{s}\sim d^{\widehat{\pi}_{j+1}}}\left[\log \mathbb{Z}_j(\mathbf{s})\right] - \sqrt{\frac{C^* H \iota}{N}}
\end{aligned}$$

where the $\sqrt{\frac{C^* H \iota}{N}}$ guarantee for terms (a) and (c) arises under the conditions studied in Section 4.3.

Interpretation of Theorem 4.4. Theorem 4.4 says that if atleast k many updates can be made to the underlying empirical MDP, $\widehat{M}$, such that each update is non-trivially lower-bounded, i.e., $\mathbb{E}_{\mathbf{s}\sim d^{\widehat{\pi}}_{k+1}}\left[\log \mathbb{Z}_k(\mathbf{s})\right] \geq c_0 > 0$, then the performance improvement obtained by k-steps of policy improvement is bounded below by $kc_0/\eta - \mathcal{O}(\sqrt{H/N})$. This result indicates that if $k = \mathcal{O}(H)$ many high advantage policy updates are possible in a given empirical MDP, then the methods with that perform $\mathcal{O}(H)$ steps of policy improvement will attain higher performance than the counterparts that perform only one update.

This is typically the case in maze navigation-style environments, where $\mathcal{O}(H)$ many possible high-advantage updates are possible on the empirical MDP, especially by "stitching" parts of suboptimal trajectories to obtain a much better trajectory. Therefore, we expect that in offline RL problems where stitching is possible, offline RL algorithms will attain an improved performance compared to one or a few-steps of policy improvement.

C Guarantees for Policy-Constraint Offline RL

In this section, we analyze a policy-constraint offline algorithm (Levine et al., 2020) that constrains the policy to choose a safe set of actions by explicitly preventing action selection from previously unseen, low-density actions. The algorithm we consider builds upon the `MBS-PI` algorithm from Liu et al. (2020b), which truncates Bellman backups and policy improvement steps from low-density, out-of-support state-action pairs. The algorithm is described in detail in Algorithm 2, but we provide a summary below. Let $\widehat{\mu}(\mathbf{s},\mathbf{a})$ denote the empirical state-action distribution and choose a constant b.

Then, let $\zeta(\mathbf{s}, \mathbf{a}) = 1\{\widehat{\mu}(\mathbf{s}, \mathbf{a}) \geq b\}$ be the indicator of high-density state-action tuples. The algorithm we analyze performs the following update until convergence:

$$\widehat{Q}^{\pi}_{\zeta}(\mathbf{s}, \mathbf{a}) \leftarrow \widehat{r}(\mathbf{s}, \mathbf{a}) + \gamma \sum_{(\mathbf{s}', \mathbf{a}')} \widehat{P}(\mathbf{s}'|\mathbf{s}, \mathbf{a})\pi(\mathbf{a}'|\mathbf{s}')\zeta(\mathbf{s}', \mathbf{a}') \cdot \widehat{Q}^{\pi}_{\zeta}(\mathbf{s}', \mathbf{a}'), \quad \text{for all } (\mathbf{s}, \mathbf{a}),$$

$$\widehat{\pi} \leftarrow \arg\max_{\pi} \mathbb{E}_{\mathbf{s} \sim \mathcal{D}} \left[\mathbb{E}_{\mathbf{a} \sim \pi'} \left[\zeta(\mathbf{s}, \mathbf{a}) \cdot \widehat{Q}^{\pi}_{\zeta}(\mathbf{s}, \mathbf{a}) \right] \right],$$

In order to derive performance guarantees for this generic policy-constraint algorithm, we define the notion of a $\zeta-$covered policy following Liu et al. (2020b) in Definition C.1. The total occupancy of all out-of-support state-action pairs (i.e., $(\mathbf{s}, \mathbf{a})$ such that $\zeta(\mathbf{s}, \mathbf{a}) = 0$) under a ζ-covered policy is bounded by a small constant U, which depends on the threshold b. Let π^*_{ζ} denote the best performing ζ-covered policy.

Definition C.1 (ζ-covered)**.** *π is called ζ-covered if* $\sum_{(\mathbf{s}, \mathbf{a})}(1 - \zeta(\mathbf{s}, \mathbf{a}))d^{\pi}(\mathbf{s}, \mathbf{a}) \leq (1 - \gamma)U(b)$.

Equipped with this definition C.1, Lemma C.1 shows that the total value estimation error of any given $\zeta-$covered policy, π, $|J(\pi) - \widehat{J}_{\zeta}(\pi)|$ is upper bounded in expectation over the dataset

Lemma C.1 (Value estimation error of a ζ-covered policy)**.** *For any given ζ-covered policy π, under Condition 3.2, the estimation error $|J(\pi) - \widehat{J}_{\zeta}(\pi)|$ is bounded as:*

$$\mathbb{E}_{\mathcal{D}} \left[\left| J(\pi) - \widehat{J}_{\zeta}(\pi) \right| \right] \lesssim \sqrt{\frac{C^* |\mathcal{S}| H \iota}{N}} + \frac{C^* |\mathcal{S}| H \iota}{N} + U(b) \tag{9}$$

Proof. To prove this lemma, we consider the following decomposition of the policy performance estimate:

$$\begin{aligned}
&\left| J(\pi) - \widehat{J}_{\zeta}(\pi) \right| \\
&= \sum_{t=0}^{\infty} \sum_{(\mathbf{s}, \mathbf{a})} \gamma^t d^{\pi}_t(\mathbf{s}, \mathbf{a}) \left[\sum_{(\mathbf{s}', \mathbf{a}')} \left(\widehat{P}(\mathbf{s}'|\mathbf{s}, \mathbf{a})\zeta(\mathbf{s}', \mathbf{a}') - P(\mathbf{s}'|\mathbf{s}, \mathbf{a}) \right) \cdot \widehat{Q}^{\pi}(\mathbf{s}', \mathbf{a}') \right] \\
&= \underbrace{\sum_{t=0}^{\infty} \sum_{(\mathbf{s}, \mathbf{a})} \gamma^t d^{\pi}_t(\mathbf{s}, \mathbf{a}) \sum_{(\mathbf{s}', \mathbf{a}')} (\widehat{P}(\mathbf{s}'|\mathbf{s}, \mathbf{a}) - P(\mathbf{s}'|\mathbf{s}, \mathbf{a})) \cdot \zeta(\mathbf{s}', \mathbf{a}') \cdot \pi(\mathbf{a}'|\mathbf{s}') \cdot \widehat{Q}(\mathbf{s}', \mathbf{a}')}_{\Delta_1 \text{:bound using concentrability and variance recursion}} \\
&\qquad + \underbrace{\sum_{t=0}^{\infty} \gamma^t d^{\pi}_t(\mathbf{s}, \mathbf{a}) \sum_{(\mathbf{s}', \mathbf{a}')} P(\mathbf{s}'|\mathbf{s}, \mathbf{a}) \cdot (1 - \zeta(\mathbf{s}', \mathbf{a}')) \cdot \pi(\mathbf{a}'|\mathbf{s}') \cdot \widehat{Q}^{\pi}(\mathbf{s}', \mathbf{a}')}_{\Delta_2 \text{:bias due to leaving support; upper bounded due to } \zeta\text{-cover}}
\end{aligned}$$

To bound the inner summation over $(\mathbf{s}', \mathbf{a}')$ in term (a), we can apply Lemma B.5 since $\widehat{P}(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ and $\zeta(\mathbf{s}', \mathbf{a}')$ are not independent, to obtain a horizon-free bound. Finally, we use Condition 3.1 to bound the density ratios, in expectation over the randomness in dataset $\mathcal{D}$, identical to the proof for the conservative lower-confidence bound method from before. Formally, using Lemma B.5, we get, with high probability $\geq 1 - \delta$:

$$\forall(\mathbf{s}, \mathbf{a}) \text{ s.t. } n(\mathbf{s}, \mathbf{a}) \geq 1, \left| \left(\widehat{P}(\mathbf{s}, \mathbf{a}) - P(\mathbf{s}, \mathbf{a}) \right) \cdot \widehat{V}^{\pi}_{\zeta} \right| \leq \sqrt{\frac{\mathbb{V}(\widehat{P}(s, a), \widehat{V}^{\pi}_{\zeta})\iota}{n(s, a)}} + \frac{\iota}{n(s, a)},$$

where we utilized the fact that $\widehat{V}^{\pi}_{\zeta} \leq \widehat{V}^{\pi} \leq 1$ due to Condition 3.2. For bounding Δ_2, we note that this term is bounded by the definition of ζ-covered policy:

$$\Delta_2 \leq \sum_{t=0}^{\infty} \gamma^t (1 - \gamma) U(b) \leq U(b). \tag{10}$$

Thus, the overall policy evaluation error is given by:

$$\left| J(\pi) - \widehat{J}_{\zeta}(\pi) \right| \lesssim \sum_{t=0}^{\infty} \gamma^t d^{\pi}_t(\mathbf{s}, \mathbf{a}) \left[\sqrt{\frac{\mathbb{V}(\widehat{P}(s, a), \widehat{V}^{\pi}_{\zeta})\iota}{n(s, a)}} + \frac{\iota}{n(s, a)} \right] + U(b). \tag{11}$$

Equation 11 mimics the Φ term in equation 7 that is bounded in Section B.2.5, with an additional offset $U(b)$. Hence, we can reuse the same machinery to show the bound in expectation over the randomness in the dataset, which completes the proof. $\square$

Using Lemma C.1, we can now that the policy constraint algorithm attains a favorable guarantee when compared to the best policy that is ζ-covered:

Theorem C.1 (Performance of our policy-constraint algorithm)**.** *Under Condition 3.2, the policy $\widehat{\pi}^*$ incurs bounded suboptimality against the best ζ-covered policy, with high probability $\geq 1-\delta$:*

$$\mathbb{E}_{\mathcal{D}}\left[J(\pi^*_\zeta) - J(\widehat{\pi}^*)\right] \lesssim \sqrt{\frac{C^*|\mathcal{S}|H\iota}{N}} + \frac{C^*|\mathcal{S}|H\iota}{N} + 2U(b)\,.$$

To prove this theorem, we use the result of Lemma C.1 for the fixed policy, that is agnostic of the dataset, and then again use the recursion as before to bound the value of the data-dependent policy. The latter uses Lemma B.5 and ends up attaining a bound previously found in Appendix B.2.5, which completes the proof of this Theorem. When the term $U(b)$ is small, such that $U(b) \leq \mathcal{O}(H^{0.5-\varepsilon})$ for $\varepsilon > 0$, then we find that the guarantee in Theorem C.1 matches that in Theorem 4.2, modulo a term that grows slower in the horizon than the other terms in the bound. If $U(b)$ is indeed small, then all properties that applied to conservative offline RL shall also follow for policy-constraint algorithms.

Note on the bound. We conjecture that it is possible to get rid of the $U(b)$ term, under certain assumptions on the support indicator $\zeta(\mathbf{s}, \mathbf{a})$, and by relating the values of $\zeta(\mathbf{s}, \mathbf{a})$ and $\zeta(\mathbf{s}', \mathbf{a}')$, at consecutive state-action tuples. For example, if $\zeta(\mathbf{s}', \mathbf{a}') = 1 \implies \zeta(\mathbf{s}, \mathbf{a}) = 1$, then we can derive a stronger guarantee.

D Experimental Details

In this section we provide a detailed description of the various tasks used in this paper, and describe the data collection procedures for various tasks considered. We discuss the details of our tasks and empirical validation at the following website: https://sites.google.com/view/shouldirunrlorbc/home.

D.1 Tabular Gridworld Domains

The gridworld domains we consider are described by 10×10 grids, with a start and goal state, and walls and lava placed in between. We consider a sparse reward where the agent earns a reward of 1 upon reaching the goal state; however, if the agent reaches a lava state, then its reward is 0 for the rest of the trajectory. The agent is able to move in either of the four direction (or choose to stay still); to introduce stochasticity in the transition dynamics, there is a 10% chance that the agent travels in a different direction than commanded.

The exact three gridworlds we evaluate on vary in the number of critical points encountered per trajectory. We model critical states as holes in walls through which the agent must pass; if the agent chooses a wrong action at those states, it veers off into a lava state. The exact three gridworlds we evaluate on are: (a) "Single Critical" with one critical state per trajectory, (b) "Multiple Critical" with three critical states per trajectory, and (c) "Cliffwalk", where every state is critical (Schaul et al., 2015). The renderings of each gridworld are in Figure 3.

D.2 Multi-Stage Robotic Manipulation Domains

Overview of domains. These tasks are taken from Singh et al. (2020). The robotic manipulation simulated domains comprise of a 6-DoF WidowX robot that interacts with objects in the environment. There are three tasks of interest, all of which involve a drawer and a tray. The objective of each task is to remove obstructions of the drawer, open the drawer, pick an object and place it in a tray. The obstructions of the drawer were varied giving rise to three different domains — **open-grasp** (no obstruction of the drawer), **close-open-grasp** (an open top drawer obstructs the bottom drawer), **pick-place-open-grasp** (an object obstructs the bottom drawer).

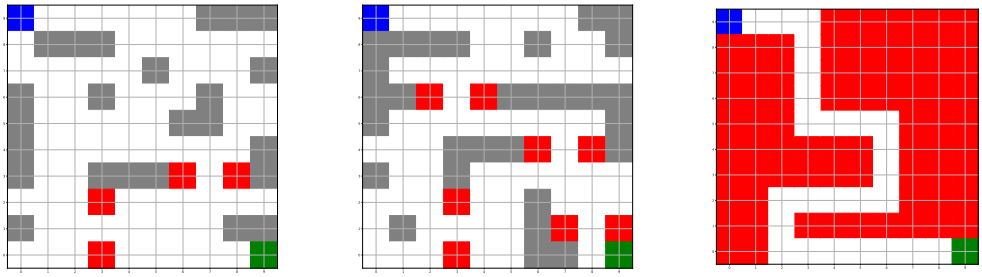

Figure 3: Renderings of three gridworld domains we evaluate on, where states are colored as: Start:blue, Goal:green, Lava:red, Wall:grey, and Open:white. The domains have varying number of critical points. *Left*: Single Critical. *Middle*: Multiple Critical. *Right*: Cliffwalk

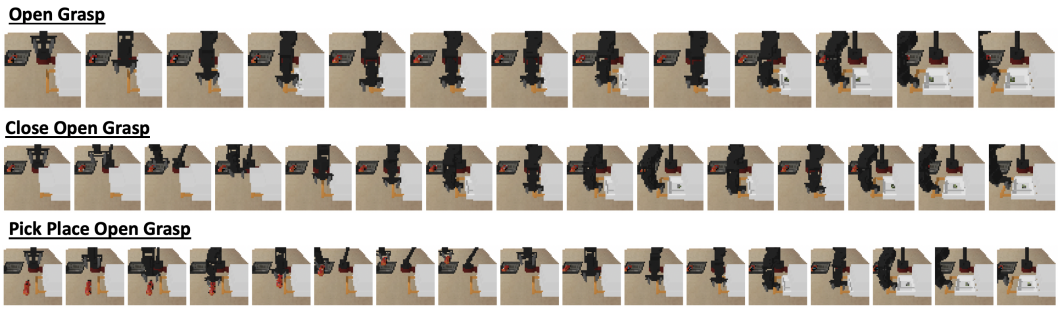

Figure 4: Filmstrip of the three tasks that we stufy for robotic manipulation – open-grasp, close-open-grasp and pick-place-open-grasp.

Reward function. For all the three tasks considered, a reward of $+1$ is provided when the robot is successfully able to open the drawer of interest (bottom drawer in close-open-grasp and pick-place-open-grasp; the only drawer in open-grasp) and is able to grasp the object inside it. If the robot fails at doing so, it gets no reward.

Dataset composition. For each task, we collected a dataset comprising of 5000 trajectories. For our experiments where we utilize expert data, we used the (nearly)-expert scripted policy for collecting trajectories and discarded the ones that failed to succeed. Thus the expert data attains a 100% success rate on this task. For our experiments with suboptimal data, which is used to train offline RL, we ran a noisy version of this near-expert scripted policy and collected 5000 trajectories. The average success rate in the suboptimal data is around 40-50% in both opening and closing the drawers with, 70% success rate in grasping objects, and a 70% success rate in place those objects at random locations in the workspace.

D.3 AntMaze Domains

Overview of the domain. This task is based on the antmaze-medium and antmaze-large environments from Fu et al. (2020). The goal in this environment is to train an 8-DoF quadraped ant robot to successfully navigate to a given, pre-specifcied target location in a maze. We consider two different maze layouts provided by Fu et al. (2020). We believe that this domain is well-suited to test BC and RL methods in the presence of multiple critical points, and is representative of real-world navigation scenarios.

Scripted policies and datasets. We utilize the scripted policies provided by Fu et al. (2020) to generate two kinds of expert datasets: first, we generate trajectories that actually traverse the path from a given default start location to the target goal location that we consider for evaluation, and second, we generate trajectories that go from multiple random start positions in the maze to the target

Domain / Behavior Policy	Task/Data Quality	BC	Naïve CQL	Tuned CQL
7 Atari games (RL policy)	Pong, Expert	109.78 ± 2.93	102.03 ± 4.43	105.84 ± 2.22
	Breakout, Expert	75.59 ± 21.59	71.22 ± 27.55	94.77 ± 27.02
	Asterix, Expert	41.10 ± 9.5	44.81 ± 12.0	80.19 ± 20.7
	SpaceInvaders, Expert	40.88 ± 4.17	45.27 ± 7.32	54.15 ± 2.96
	Q*bert, Expert	121.48 ± 9.06	105.83 ± 23.17	98.52 ± 18.62
	Enduro, Expert	78.67 ± 3.98	141.53 ± 18.79	127.02 ± 10.53
	Seaquest, Expert	63.15 ± 9.47	64.03 ± 27.67	85.28 ± 21.28

Table 3: Per-game results for the Atari domains with expert data. Note that while naïve CQL does not perform much better than BC (it performs similarly as BC), tuned CQL with the addition of the DR3 regularizer performs much better.

goal location in the maze. The latter has a wider coverage and a different initial state distribution compared to what we will test these algorithms on. We collected a dataset of 500k transitions, which was used by both BC and offline RL.

Reward functions. In this task, we consider a sparse binary reward $r(\mathbf{sa}) = +1$, if $|\mathbf{s}' - \mathbf{g}| \leq \varepsilon = 0.5$ and 0 otherwise. This reward is only provided at the end of a trajectory. This reward function is identical to the one reported by D4RL (Fu et al., 2020), but the dataset composition in our case comes from an expert policy.

D.4 Adroit Domains

Overview of the domain. The Adroit domains (Rajeswaran et al., 2018; Fu et al., 2020) involve controlling a 24-DoF simulated Shadow Hand robot tasked with hammering a nail (hammer), opening a door (door), twirling a pen (pen) or picking up and moving a ball (relocate). This domain presents itself with narrow data distributions, and we utilize the demonstrations provided by Rajeswaran et al. (2018) as our expert dataset for this task. The environments were instantiated via D4RL, and we utilized the environments marked as: `hammer-human-longhorizon`, `door-human-longhorizon`, `pen-human-longhorizon` and `relocate-human-longhorizon` for evaluation.

Reward functions. We directly utilize the data from D4RL (Fu et al., 2020) for this task. However, we modify the reward function to be used for RL. While the D4RL adroit domains provide a dense reward function, with intermediate bonuses provided for various steps, we train offline RL using a binary reward function. To compute this binary reward function, we first extract the D4RL dataset for these tasks, and then modify the reward function as follows:

$$r(\mathbf{s}, \mathbf{a}) = +1 \quad \text{if } r_{\text{D4RL}}(\mathbf{s}, \mathbf{a}) \geq 70.0 \quad \text{(hammer-human)} \tag{12}$$

$$r(\mathbf{s}, \mathbf{a}) = +1 \quad \text{if } r_{\text{D4RL}}(\mathbf{s}, \mathbf{a}) \geq 9.0 \quad \text{(door-human)} \tag{13}$$

$$r(\mathbf{s}, \mathbf{a}) = +1 \quad \text{if } r_{\text{D4RL}}(\mathbf{s}, \mathbf{a}) \geq 47.0 \quad \text{(pen-human)} \tag{14}$$

$$r(\mathbf{s}, \mathbf{a}) = +1 \quad \text{if } r_{\text{D4RL}}(\mathbf{s}, \mathbf{a}) \geq 18.0 \quad \text{(relocate-human)} \tag{15}$$

The constant thresholds for various tasks are chosen in a way that only any transition that actually activates the flag `goal_achieved=True` flag in the D4RL Adroit environments attains a reward +1, while other transitions attain a reward 0. We also evaluarte the performance of various algorithms on this new sparse reward that we consider for our setting.

D.5 Atari Domains

We utilized 7 Atari games which are commonly studied in prior work (Kumar et al., 2020; 2021b): Asterix, Breakout, Seaquest, Pong, SpaceInvaders, Q*Bert, Enduro for our experiments. We do not modify the Atari domains, directly utilize the sparse reward for RL training and operate in the stochastic Atari setting with sticky actions for our evaluations. For our experiments, we extracted datasets of different qualities from the DQN-Replay dataset provided by Agarwal et al. (2020b). The DQN-Replay dataset is stored as 50 buffers consisting of sequentially stored data observed during training of an online DQN agent over the course of training.

Task	BC-PI	CQL
Pong	100.03 ± 5.01	94.48 ± 8.39
Breakout	25.99 ± 1.98	86.92 ± 13.74
Asterix	29.77 ± 5.33	157.54 ± 37.94
SpaceInvaders	31.45 ± 1.96	63.7 ± 16.18
Q*bert	106.06 ± 8.63	88.72 ± 20.41
Enduro	68.56 ± 0.23	148.97 ± 12.3
Seaquest	22.51 ± 2.23	124.95 ± 43.86

Table 4: Comparing the performance of BC-PI and offline RL on noisy-expert data. Observe that in general, offline RL significantly outperforms BC-PI.

Expert data. To obtain expert data for training BC and RL algorithms, we utilized all the data from buffer with id 49 (i.e., the last buffer stored). Since each buffer in DQN-Replay consists of 1M transition samples, all algorithms training on expert data learn from 1M samples.

Noisy-expert data. For obtaining noisy-expert data, analogous to the gridworld domains we study, we mix data from the optimal policy (buffer 49) with an equal amount of random exploration data drawn from the initial replay buffers in DQN replay (buffers 0-5). i.e. we utilize 0.5M samples form buffer 49 in addition to 0.5M samples sampled uniformly at random from the first 5 replay buffers.

E Tuning and Hyperparameters

In this section, we discuss our tuning strategy for BC and CQL used in our experiments. We tuned CQL offline, using recommendations from prior work (Kumar et al., 2021c). We used default hyperparameters for the CQL algorithm (Q-function learning rate = 3e-4, policy learning rate = 1e-4), based on prior works that utilize these domains. Note that prior works do not use the kind of data distributions we use, and our expert datasets can be very different in composition compared to some of the other medium or diverse data used by prior work in these domains. In particular, with regards to the hyperaprameter α in CQL that trades off conservatism and the TD error objective, we used $\alpha = 0.1$ for all Atari games (following Kumar et al. (2021b)), and $\alpha = 1.0$ for the robotic manipulation domains following (Singh et al., 2020). For the Antmaze and Adroit domains, we ran CQL training with multiple values of $\alpha \in \{0.01, 0.1, 0.5, 1.0, 5.0, 10.0, 20.0\}$, and then picked the smallest α that did not lead to eventually divergent Q-values (either positively or negatively) with more (1M) gradient steps. Next, we discuss how we regularized the Q-function training and performed policy selection on the various domains.

- **Detecting overfitting and underfitting**: Following Kumar et al. (2021c), as a first step, we detect whether the run is overfitting or underfitting, by checking the trend in Q-values. In our experiments, we found that Q-values learned on Adroit domains exhibited a decreasing trend throughout training, from which we concluded it was overfitting. On the Antmaze and Atari experiments, Q-values continued to increase and eventually stabilized, indicating that the run might be underfitting (but not overfitting).
- **Correcting overfitting and policy selection**: As recommended, we applied a capacity decreasing regularizer to correct for overfitting, by utilizing dropout on every layer of the Q-function. We ran with three values of dropout parobability, $p \in \{0.1, 0.2, 0.4\}$, and found that 0.4 was the most effective in alleviating the monotonically decreasing trend in Q-values, so used that for our results. Then, we performed policy checkpoint selection by picking the earliest checkpoint that appears after the peak in the Q-values for our evaluation.
- **Correcting underfitting:** In the Atari and Antmaze domains, we observed that the Q-values exhibited a stable, convergent trend and did not decrease with more training. Following Kumar et al. (2021c), we concluded that this resembled underfitting and utilized a capacity-increasing regularizer (DR3 regularizer (Kumar et al., 2021a) for addressing this issue. We used identical hyperparameter for the multiplier (β) on this regularizer term for both Atari and Antmaze, $\beta = 0.03$ and never tuned it.

For BC, in all domains, we tested BC with different network architectures. On the antmaze domain, we evaluated two feed-forward policy architectures of sizes $(256, 256, 256)$ and $(256, 256, 256, 256, 256, 256)$ and picked the one that performed best online. ON Adroit domains,

we were not able to get a tanh-Gaussian policy, typically used in continuous control to work well, since it overfitted very quickly giving rise to worse-than-random performance and therefore, we switched to utilizing a Gaussian policy network with hidden layer sizes $(256, 256, 256, 256)$, and a learned, state-dependent standard deviation. To prevent overfitting in BC, we applied a strong dropout regularization of $p = 0.2$ after each layer for Adroit domains. On Atari and the manipulation domains, we utilized a Resnet architecture borrowed from IMPALA (Espeholt et al., 2018), but without any layer norm.

More details are at: https://sites.google.com/view/shouldirunrlorbc/home.