A Extended experimental results

Here we present extended versions of the D4RL experiments. We use the same setup as in Section 6, but run each of the algorithms on three different datasets in each environment. Explicitly we show results on ANTMAZE-UMAZE, ANTMAZE-MEDIUM-PLAY, and ANTMAZE-LARGE-PLAY in Figure 5. Then we show results on HALFCHEETAH-MEDIUM, HALFCHEETAH-MEDIUM-REPLAY, and HALFCHEETAH-MEDIUM-EXPERT in Figure 6. Finally we show results on PEN-HUMAN, PEN-CLONED, and PEN-EXPERT in Figure 7.

These experiments corroborate the story from the main text. Without return coverage (as in the larger antmaze tasks), RCSL can fail dramatically. But in the case with return coverage but poor state coverage (as in the pen human dataset that only has 25 trajectories), RCSL can beat DP. However we see that with larger datasets that yield more coverage, DP recovers it's performance (as in pen expert which has 5000 trajectories, or 200x the amount of data as in the human dataset).

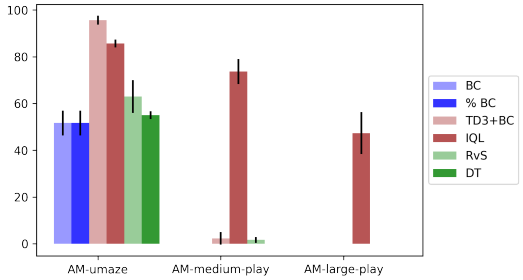


Figure 5: Experimental results on antmaze datasets.

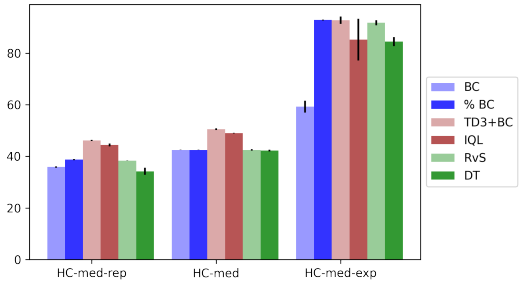


Figure 6: Experimental results on halfcheetah datasets.

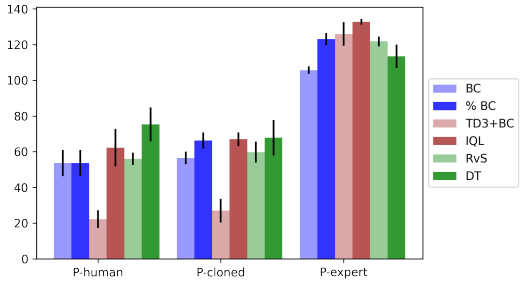


Figure 7: Experimental results on pen datasets.

B Trajectory stitching

B.1 Theory

A common goal from the offline RL literature is to be able to stitch together previously collected trajectories to outperform the behavior policy. This is in general not possible with RCSL. The

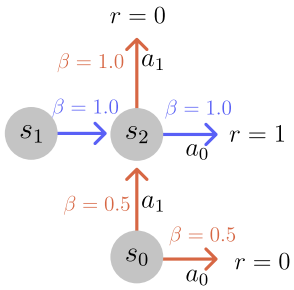


Figure 8: An example where RCSL fails to stitch trajectories.

main issue here is that RCSL is using *trajectory level* information during training, which precludes combining information across trajectories. In this example we show that even with infinite data, when attempting to combine two datastreams using standard approaches to conditioning RCSL can fail to recover the optimal policy.

Consider the MDP illustrated in Figure 8 with three states s_0, s_1, s_2 and horizon $H = 2$. All transitions and rewards are deterministic as shown. We consider the case where data has been collected by two different processes. One process (illustrated in red) consists of episodes that always start at s_0 and chooses the first action uniformly but chooses the bad action a_0 deterministically from s_2. The other process (illustrated in blue) consists of trajectories that always start at s_1 and deterministically go towards the good action, receiving reward of 1. We will consider what happens to RCSL at test time when initialized at s_0.

The data does not contain any trajectories that begin in s_0 and select a_1 to transition to s_2 followed by a_1, which is the optimal decision. But, the data does have enough information to stitch together the optimal trajectory from s_0, and it is clear to see that DP-based approaches would easily find the optimal policy.

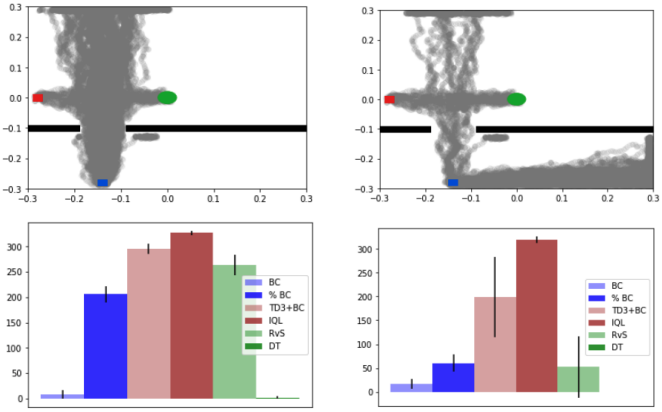
For RCSL, if we condition on optimal return $g = 1$, we get that $\pi(\cdot|s_1, g = 1)$ is undefined since we only observe trajectories with $g = 0$ that originate at s_0. To get a well-defined policy, we must set $f(s_0) = 0$, but then $\pi(a_1|s_1, g = 0) = 0.5$. Thus, π will never choose the optimal path with probability larger than 0.5, for any conditioning function f. Moreover, the conditioning function that does lead to success is non-standard: $f(s_0) = 0, f(s_2) = 1$. For the standard approach to conditioning of setting the initial value and decreasing over time with observed rewards, RCSL will never achieve non-zero reward from s_0.

Note that DT-style learning where we condition on the entire history of states rather than just the current state can perform even worse since $P_{data}(a_1|s_0, a_0, s_2, g = 1) = 0$, i.e. even using the non-standard conditioning function described above will not fix things. Also, it is worth mentioning that it is possible that conditioning on the out-of-distribution return $g = 1$ from s_0 could work due to extrapolation of a neural network policy. However, as we will see in the experiments section, this does not happen empirically in controlled settings.

B.2 Experiments

The above example does not take into account the possibility of generalization out of distribution (i.e. when conditioning on returns that were not observed in the data). To test whether generalization could lead to stitching we construct two datasets: stitch-easy and stitch-hard. Both datasets use the same simple point-mass environment with sparse rewards as before, but now we introduce a wall into the environment to limit the paths to the goal. The stitch-easy dataset contains two types of trajectories: some beginning from the initial state region and moving upwards (with added Gaussian noise in the actions) and some beginning from the left side of the environment and moving towards the goal (with added Gaussian noise in the actions). This is "easy" since just following the behavior policy for the first half of the trajectory leads to states where the dataset indicates how to reach the goal. We also create the stitch-hard dataset which includes a third type of trajectory that begins from the initial state and goes to the right (mirroring the tabular example). This is "hard" since the dominant action from the behavior in the initial state is now to go right rather than to move towards

the goal-reaching trajectories. This acts as a distraction for methods that are biased towards the behavior. Datasets and results are illustrated in Figure 9.



(a) Stitch-easy (b) Stitch-hard

Figure 9: Results on two datasets that require stitching.

We see that on the stitch-easy dataset RvS is able to perform almost as well as the DP algorithms and better than %BC. This indicates that it is able to follow the behavior when conditioning on an out-of-distribution return until it reaches a state where the return directs it to the goal. In contrast, DT totally fails on this task since it conditions on the entire history of the trajectory. Since the dataset only contains trajectories from the initial state that continue to the top of the board, DT always reproduces such trajectories from the initial state and does not stitch together trajectories.

In the stitch-hard dataset, we see that RvS fails as well and does not outperform %BC. This indicates that indeed, RvS can be distracted by the distractor trajectories from the initial state. The conditioning itself was not what cause the stitching in the stitch-easy dataset, but rather the learned policy simply defaults to the behavior. This can be beneficial in some problems, but prevents trajectory stitching that might allow the learned policy to dramatically outperform the behavior. TD3+BC also struggles here, likely due to some combination of instability and the BC regularization causing issues due to the distractor actions.

C Proofs

C.1 Proof of Theorem 1

Proof. Let $g(s_1, a_{1:H})$ be the value of the return by rolling out the open loop sequence of actions $a_{1:H}$ under the deterministic dynamics induced by T and r. Then we can write

$$\mathbb{E}_{s_1}[f(s_1)] - J(\pi_f) = \mathbb{E}_{s_1}\left[\mathbb{E}_{\pi_f|s_1}[f(s_1) - g_1]\right] \tag{12}$$

$$= \mathbb{E}_{s_1}\left[\mathbb{E}_{a_{1:H}\sim\pi_f|s_1}[f(s_1) - g(s_1, a_{1:H})]\right] \tag{13}$$

$$+ \mathbb{E}_{s_1}\left[\mathbb{E}_{a_{1:H}\sim\pi_f|s_1}[g(s_1, a_{1:H}) - g_1]\right] \tag{14}$$

$$\leq \mathbb{E}_{s_1}\left[\mathbb{E}_{a_{1:H}\sim\pi_f|s_1}[f(s_1) - g(s_1, a_{1:H})]\right] + \epsilon H^2. \tag{15}$$

where the last step follows by bounding the magnitude of the difference between g_1 and $g(s_1, a_{1:H})$ by H and applying a union bound over the H steps in the trajectory (using the near determinism assumption), namely:

$$H \cdot \sup_{s_1} \bigcup_t P_{a_t\sim\pi_f|s_1}(r_t \neq r(s_t, a_t) \text{ or } s_{t+1} \neq T(s_t, a_t)) \leq \epsilon H^2. \tag{16}$$

Now we consider the first term from eq. (15). Again bounding the magnitude of the difference by H we get that

$$\mathbb{E}_{s_1}\left[\mathbb{E}_{a_{1:H}\sim\pi_f|s_1}[f(s_1) - g(s_1, a_{1:H})]\right] \leq \mathbb{E}_{s_1} \int_{a_{1:H}} P_{\pi_f}(a_{1:H}|s_1)\mathbb{1}[g(s_1, a_{1:H}) \neq f(s_1)]H \tag{17}$$

To bound this term, we will more carefully consider what happens under the distribution P_{π_f}.

To simplify notation, let $\bar{s}_t = T(s_1, a_{1:t-1})$ be the result of following the deterministic dynamics defined by T up until step t. Expanding it out, applying the near determinism, the consistency of f, the coverage assumption, canceling some terms, and then inducting we see that:

$$P_{\pi_f}(a_{1:H}|s_1) = \pi_f(a_1|s_1) \int_{s_2} P(s_2|s_1, a_1) P_{\pi_f}(a_{2:H}|s_1, s_2) \tag{18}$$

$$\leq \pi_f(a_1|s_1) P_{\pi_f}(a_{2:H}|s_1, \bar{s}_2) + \epsilon \tag{19}$$

$$= \beta(a_1|s_1) \frac{P_\beta(g_1 = f(s_1)|s_1, a_1)}{P_\beta(g_1 = f(s_1)|s_1)} P_{\pi_f}(a_{2:H}|s_1, \bar{s}_2) + \epsilon \tag{20}$$

$$\leq \beta(a_1|s_1) \frac{\epsilon + P_\beta(g_1 - r(s_1, a_1) = f(s_1) - r(s_1, a_1)|s_1, a_1, \bar{s}_2)}{P_\beta(g_1 = f(s_1)|s_1)} P_{\pi_f}(a_{2:H}|s_1, \bar{s}_2) + \epsilon \tag{21}$$

$$= \beta(a_1|s_1) \frac{\epsilon + P_\beta(g_2 = f(\bar{s}_2)|\bar{s}_2)}{P_\beta(g_1 = f(s_1)|s_1)} P_{\pi_f}(a_{2:H}|s_1, \bar{s}_2) + \epsilon \tag{22}$$

$$\leq \beta(a_1|s_1) \frac{P_\beta(g_2 = f(\bar{s}_2)|\bar{s}_2)}{P_\beta(g_1 = f(s_1)|s_1)} P_{\pi_f}(a_{2:H}|s_1, \bar{s}_2) + \epsilon \left(\frac{1}{\alpha_f} + 1\right) \tag{23}$$

$$\leq \beta(a_1|s_1)\beta(a_2|\bar{s}_2) \frac{\cancel{P_\beta(g_2 = f(\bar{s}_2)|\bar{s}_2)}}{P_\beta(g_1 = f(s_1)|s_1)} \cdot \frac{P_\beta(g_2 = f(\bar{s}_2)|\bar{s}_2, a_2)}{\cancel{P_\beta(g_2 = f(\bar{s}_2)|\bar{s}_2)}} P_{\pi_f}(a_{3:H}|s_1, \bar{s}_3)) \tag{24}$$

$$+ 2\epsilon \left(\frac{1}{\alpha_f} + 1\right) \tag{25}$$

$$\leq \prod_{t=1}^{H} \beta(a_t|\bar{s}_t) \frac{P_\beta(g_H = f(\bar{s}_H)|\bar{s}_H, a_h)}{P_\beta(g_1 = f(s_1)|s_1)} + H\epsilon \left(\frac{1}{\alpha_f} + 1\right) \tag{26}$$

$$= \prod_{t=1}^{H} \beta(a_t|\bar{s}_t) \frac{\mathbb{1}[g(s_1, a_{1:H}) = f(s_1)]}{P_\beta(g_1 = f(s_1)|s_1)} + H\epsilon \left(\frac{1}{\alpha_f} + 1\right) \tag{27}$$

where the last step follows from the determinism of the trajectory that determines $\bar{s}_H$ and the consistency of f. Plugging this back into eq (17) and noticing that the two indicator functions can never both be 1, we get that:

$$\mathbb{E}_{s_1}\left[\mathbb{E}_{a_{1:H} \sim \pi_f|s_1}[f(s_1) - g(s_1, a_{1:H})]\right] \leq H^2\epsilon \left(\frac{1}{\alpha_f} + 1\right) \tag{28}$$

Plugging this back into eq (15) yields the result. □

C.2 Proof of Corollary 1

Proof. We need to define a function f so that $\mathbb{E}[f(s_1)]$ is approximately $J(\pi^*)$. To do this, note that there exists a deterministic optimal policy π^*, and since the environment dynamics are nearly deterministic we can set $f(s_1)$ to be the return of π^* under the deterministic dynamics. To do this, let $T^{\pi^*}(s_1, t)$ represent the state reached by running π^* from s_1 for t steps under the deterministic dynamics defined by T. Then:

$$f(s_1) = \sum_{t=1}^{H} r(T^{\pi^*}(s_1, t), \pi^*(T^{\pi^*}(s_1, t))) \tag{29}$$

Now we have as in the proof of Theorem 1 that the probability that $g \neq f(s)$ is bounded by ϵH, so that

$$\mathbb{E}_{s_1}[f(s_1)] - J(\pi^*) = \mathbb{E}_{s_1}[\mathbb{E}_{g \sim \pi^*|s_1}[f(s_1) - g]] \leq \mathbb{E}_{s_1}[P_{\pi^*}(g \neq f(s_1)|s_1) \cdot H] \leq \epsilon H^2 \tag{30}$$

Combining this with Theorem 1 yields the result. □

C.3 Proof of Theorem 2

First we prove the following Lemma. This can be seen as a finite-horizon analog to results from Achiam et al. [1].

Lemma 1. *Let d_π refer to the marginal distribution of P_π over states only. For any two policies π, π' we have:*

$$\|d_\pi - d_{\pi'}\|_1 \leq 2H \cdot \mathbb{E}_{s\sim d_\pi}[TV(\pi(\cdot|s)\|\hat{\pi}'(\cdot|s))] \tag{31}$$

Proof. First we will define a few useful objects. Let $d^h_\pi(s) = P_\pi(s_h = s)$. Let $\Delta_h = \|d^h_\pi(s) - d^h_{\pi'}(s)\|_1$. Let $\delta_h = 2\mathbb{E}_{s\sim d^h_\pi}[TV(\pi(\cdot|s)\|\hat{\pi}'(\cdot|s))]$.

Now we claim that $\Delta_h \leq \delta_{h-1} + \Delta_{h-1}$ for $h > 1$ and $\Delta_1 = 0$.

To see this, consider some fixed h. Note that $d^h_\pi(s) = \int_{s'} d^{h-1}_\pi(s') \int_{a'} \pi(a'|s')P(s|s', a')$. Then expanding the definitions and adding and subtracting we see that

$$\begin{aligned}
\Delta_h &= \int_s |d^h_\pi(s) - d^h_{\pi'}(s)| && (32)\\
&\leq \int_s \left| \int_{s'} d^{h-1}_\pi(s') \int_{a'} (\pi(a'|s') - \pi'(a'|s'))P(s|s', a') \right| && (33)\\
&\quad + \int_s \left| \int_{s'} (d^{h-1}_\pi(s') - d^{h-1}_{\pi'}(s')) \int_{a'} \pi'(a'|s')P(s|s', a') \right| && (34)\\
&\leq 2\mathbb{E}_{s\sim d^{h-1}_\pi}[TV(\pi(\cdot|s)\|\hat{\pi}'(\cdot|s))] + \|d^{h-1}_\pi - d^{h-1}_{\pi'}\|_1 = \delta_{h-1} + \Delta_{h-1}. && (35)
\end{aligned}$$

Now applying the claim and the definition of d_π we get that

$$\|d_\pi - d_{\pi'}\|_1 \leq \frac{1}{H}\sum_{h=1}^{H} \Delta_h \leq \frac{1}{H}\sum_{h=1}^{H}\sum_{j=1}^{h-1} \delta_j \leq H\frac{1}{H}\sum_{h=1}^{H} \delta_h = 2H \cdot \mathbb{E}_{s\sim d_\pi}[TV(\pi(\cdot|s)\|\hat{\pi}'(\cdot|s))]. \tag{36}$$

□

Now we can prove the Theorem.

Proof. Applying the definition of J and Lemma 1, we get

$$\begin{aligned}
J(\pi_f) - J(\hat{\pi}_f) &= H(\mathbb{E}_{P_{\pi_f}}[r(s, a)] - \mathbb{E}_{P_{\hat{\pi}_f}}[r(s, a)]) && (37)\\
&\leq H\|d_{\pi_f} - d_{\hat{\pi}_f}\|_1 && (38)\\
&\leq 2 \cdot \mathbb{E}_{s\sim d_{\pi_f}}[TV(\pi_f(\cdot|s)\|\hat{\pi}_f(\cdot|s))]H^2 && (39)
\end{aligned}$$

Expanding definitions, using the multiply and divide trick, and applying the assumptions:

$$\begin{aligned}
2 \cdot \mathbb{E}_{s\sim d_{\pi_f}}[TV(\pi_f(\cdot|s)\|\hat{\pi}_f(\cdot|s))] &= \mathbb{E}_{s\sim d_{\pi_f}}\left[\int_a |P_\beta(a|s, f(s)) - \hat{\pi}(a|s, f(s))|\right] && (40)\\
&= \mathbb{E}_{s\sim d_{\pi_f}}\left[\frac{P_\beta(f(s)|s)}{P_\beta(f(s)|s)}\int_a |P_\beta(a|s, f(s)) - \hat{\pi}(a|s, f(s))|\right] && (41)\\
&\leq \frac{C_f}{\alpha_f}\mathbb{E}_{s\sim d_\beta}\left[P_\beta(f(s)|s)\int_a |P_\beta(a|s, f(s)) - \hat{\pi}(a|s, f(s))|\right] && (42)\\
&\leq \frac{C_f}{\alpha_f}\mathbb{E}_{s\sim d_\beta}\left[\int_g P_\beta(g|s)\int_a |P_\beta(a|s, g) - \hat{\pi}(a|s, g)|\right] && (43)\\
&= 2\frac{C_f}{\alpha_f}\mathbb{E}_{s\sim d_{\pi_f}, g\sim P_\beta|s}[TV(P_\beta(\cdot|s, g)\|\hat{\pi}(\cdot|s, g))] && (44)\\
&\leq \frac{C_f}{\alpha_f}\sqrt{2L(\hat{\pi})} && (45)
\end{aligned}$$

where the last step comes from Pinsker's inequality. Combining with the above bound on the difference in expected values yields the result. □

C.4 Proof of Corollary 3

Proof. We may write $L(\pi) = \bar{L}(\pi) - H_\beta$, where $H_\beta = -\mathbb{E}_{(s,a,g)\sim P_\beta}[\log P_\beta(a|s,g)]$ and

$$\bar{L}(\pi) := -\mathbb{E}_{(s,a,g)\sim P_\beta}[\log \pi(a|s,g)]$$

is the cross-entropy loss. Denoting $\pi^\dagger \in \arg\min_{\pi\in\Pi} L(\pi)$, we have

$$L(\hat{\pi}) = L(\hat{\pi}) - L(\pi^\dagger) + L(\pi^\dagger) \leq \bar{L}(\hat{\pi}) - \bar{L}(\pi^\dagger) + \epsilon_{approx}.$$

Denoting $\hat{L}$ the empirical cross-entropy loss that is minimized by $\hat{\pi}$, we may further decompose

$$\begin{aligned}\bar{L}(\hat{\pi}) - \bar{L}(\pi^\dagger) &= \bar{L}(\hat{\pi}) - \hat{L}(\hat{\pi}) + \hat{L}(\hat{\pi}) - \hat{L}(\pi^\dagger) + \hat{L}(\pi^\dagger) - \bar{L}(\pi^\dagger) \\ &\leq 2\sup_{\pi\in\Pi} |\bar{L}(\pi) - \hat{L}(\pi)|\end{aligned}$$

Under the assumptions on bounded loss differences, we may bound this, e.g., using McDiarmid's inequality and a union bound on Π to obtain the final result. □

C.5 Top-% BC

Theorem 3 (Alignment with respect to quantile)**.** *Let g_ρ be the $1-\rho$ quantile of the return distribution induced by β over all initial states. Let $\pi_\rho = P_\beta(a|s, g \geq g_\rho)$. Assume the following:*

1. *Coverage: $P_\beta(s_1|g \geq g_\rho) \geq \alpha_\rho$ for all initial states s_1.*
2. *Near determinism: $P(r \neq r(s,a) \text{ or } s' \neq T(s,a)|s,a) \leq \epsilon$ at all s, a for some functions T and r. Note that this does not constrain the stochasticity of the initial state at all.*

Then

$$g_\rho - J(\pi_\rho) \leq \epsilon\left(\frac{1}{\alpha_\rho} + 2\right) H^2. \tag{46}$$

Proof. The proof essentially follows the same argument as Theorem 1 with $f(s_1)$ replaced by g_ρ. The main difference comes from the fact that

$$\pi_\rho(a|s) = P_\beta(a|s, g \geq g_\rho) = \beta(a|s)\frac{P_\beta(g \geq g_\rho|s,a)}{P_\beta(g \geq g_\rho|s)} \tag{47}$$

Explicitly, we have similar to before that:

$$\begin{aligned}g_\rho - J(\pi_\rho) &= \mathbb{E}_{s_1}[\mathbb{E}_{\pi_\rho|s_1}[g_\rho - g_1]] &(48)\\ &\leq \mathbb{E}_{s_1}\mathbb{E}_{a_{1:H}\sim\pi_f|s_1}[g_\rho - g(s_1, a_{1:H})] + \epsilon H^2. &(49)\\ &\leq \mathbb{E}_{s_1}\mathbb{E}_{a_{1:H}\sim\pi_f|s_1}[\mathbb{1}[g(s_1, a_{1:H}) < g_\rho]] \cdot H + \epsilon H^2. &(50)\end{aligned}$$

We now define $\bar{s}_t = T(s_1, a_{1:t-1})$ to be the state at step t under the determinisitic dynamics and similarly $\bar{r}_t = r(\bar{s}_t, a_t)$ the reward under deterministic dynamics. Then again mirroring the proof

above, we have that

$$P_{\pi_\rho}(a_{1:H}|s_1) \leq \pi_\rho(a_1|s_1)P_{\pi_\rho}(a_{2:H}|s_1,\bar{s}_2,\bar{r}_1) + \epsilon \tag{51}$$

$$= \beta(a_1|s_1)\frac{P_\beta(g_1 \geq g_\rho|s_1,a_1)}{P_\beta(g_1 \geq g_\rho|s_1)}P_{\pi_\rho}(a_{2:H}|s_1,\bar{s}_2,\bar{r}_1) + \epsilon \tag{52}$$

$$\leq \beta(a_1|s_1)\frac{\epsilon + P_\beta(g_1 \geq g_\rho|\bar{s}_2,\bar{r}_1,a_1)}{P_\beta(g_1 \geq g_\rho|s_1)}P_{\pi_\rho}(a_{2:H}|s_1,\bar{s}_2,\bar{r}_1) + \epsilon \tag{53}$$

$$\leq \beta(a_1|s_1)\frac{P_\beta(g_1 \geq g_\rho|s_1,a_1)}{P_\beta(g_1 \geq g_\rho|s_1)}P_{\pi_\rho}(a_{2:H}|s_1,\bar{s}_2,\bar{r}_1) + \epsilon \tag{54}$$

$$\leq \beta(a_1|s_1)\frac{P_\beta(g_1 \geq g_\rho|\bar{s}_2,\bar{r}_1,a_1)}{P_\beta(g_1 \geq g_\rho|s_1)}P_{\pi_\rho}(a_{2:H}|s_1,\bar{s}_2,\bar{r}_1) + \epsilon\left(\frac{1}{\alpha_\rho}+1\right) \tag{55}$$

$$\leq \beta(a_1|s_1)\beta(a_2|\bar{s}_2)\frac{\cancel{P_\beta(g_1 \geq g_\rho|\bar{s}_2,\bar{r}_1)}}{P_\beta(g_1 \geq g_\rho|s_1)}\frac{P_\beta(g_1 \geq g_\rho|\bar{s}_2,\bar{r}_1,a_2)}{\cancel{P_\beta(g_1 \geq g_\rho|\bar{s}_2,\bar{r}_1)}}P_{\pi_\rho}(a_{3:H}|s_1,\bar{s}_3,\bar{r}_{1:2}) \tag{56}$$

$$+ 2\epsilon\left(\frac{1}{\alpha_\rho}+1\right) \tag{57}$$

$$\leq \prod_{t=1}^{H}\beta(a_t|\bar{s}_t)\frac{P_\beta(g_1 \geq g_\rho|\bar{s}_H,\bar{r}_{1:H})}{P_\beta(g_1 \geq g_\rho|s_1)} + H\epsilon\left(\frac{1}{\alpha_\rho}+1\right) \tag{58}$$

$$= \prod_{t=1}^{H}\beta(a_t|\bar{s}_t)\frac{\mathbb{1}[g(s_1,a_{1:H}) \geq g_\rho]}{P_\beta(g_1 \geq g_\rho|s_1)} + H\epsilon\left(\frac{1}{\alpha_\rho}+1\right) \tag{59}$$

Plugging this into Equation 50 we get the result. □

Theorem 4 (Reduction of %BC to SL)**.** *Let g_ρ be the $1-\rho$ percentile of the return distribution induced by β. Let $\pi_\rho = P_\beta(a|s, g \geq g_\rho)$. Assume*

1. *Bounded mismatch: $\frac{P_{\pi_\rho}(s)}{P_\beta(s|g\geq g_\rho)} \leq C_\rho$ for all s.*

Define the expected loss as $L_\rho(\hat{\pi}) = \mathbb{E}_{s\sim P_\beta|g\geq g_\rho}[KL(\pi_\rho(\cdot|s)\|\hat{\pi}(\cdot|s))]$. Then we have that

$$J(\pi_\rho) - J(\hat{\pi}) \leq C_\rho H^2\sqrt{2L_\rho(\hat{\pi})}. \tag{60}$$

Proof. Recall that d_π refers to the marginal distribution of P_π over states only. Applying the definition of J and Lemma 1, we get

$$J(\pi_\rho) - J(\hat{\pi}) = H(\mathbb{E}_{P_{\pi_\rho}}[r(s,a)] - \mathbb{E}_{P_{\hat{\pi}}}[r(s,a)]) \tag{61}$$

$$\leq H\|d_{\pi_\rho} - d_{\hat{\pi}}\|_1 \tag{62}$$

$$\leq 2\cdot\mathbb{E}_{s\sim d_{\pi_\rho}}[TV(\pi_\rho(\cdot|s)\|\hat{\pi}(\cdot|s))]H^2 \tag{63}$$

Expanding definitions, using the multiply and divide trick, and applying the assumptions:

$$2\cdot\mathbb{E}_{s\sim d_{\pi_\rho}}[TV(\pi_\rho(\cdot|s)\|\hat{\pi}(\cdot|s))] \leq C_\rho\cdot 2\mathbb{E}_{s\sim P_\beta(\cdot|g\geq g_\rho)}[TV(\pi_\rho(\cdot|s)\|\hat{\pi}(\cdot|s))] \tag{64}$$

$$\leq C_\rho\sqrt{2L(\hat{\pi})} \tag{65}$$

where the last step comes from Pinsker's inequality. Combining with the above bound on the difference in expected values yields the result. □

Corollary 5 (Sample complexity for %BC)**.** *To get finite data guarantees, add to the above assumptions the assumptions that (1) the policy class Π is finite, (2) $|\log\pi(a|s) - \log\pi(a'|s')| \leq c$ for any (a,s,a',s') and all $\pi\in\Pi$, and (3) the approximation error of Π is bounded by ϵ_{approx}, i.e. $\min_{\pi\in\Pi}L_\rho(\pi) \leq \epsilon_{approx}$. Then with probability at least $1-\delta$,*

$$J(\pi_\rho) - J(\hat{\pi}) \leq O\left(C_\rho H^2\left(\sqrt{c}\left(\frac{\log|\Pi|/\delta}{(1-\rho)N}\right)^{1/4} + \sqrt{\epsilon_{approx}}\right) + \frac{\epsilon}{\alpha_\rho}H^2\right). \tag{66}$$

D Experimental details

Data. Data for point-mass tasks was sampled from the scripted policies described in the text. We sampled 100 trajectories of length 400 for each dataset, unless otherwise indicated. Data for the benchmark experiments was taken directly from the D4RL benchmark [10].

Hyperparameters. Below we list all of the hyperparameters used across the various agorithms. We train each algorithm on 90% of the trajectories in the dataset, using the remaining 10% as validation. All algorithms are trained with the Adam optimizer [18]. We evaluate each algorithm for 100 episodes in the environment per seed and hyperparameter configuration and report the best performance for each algorithm for it's relevant hyperparameter (All algorithms were tuned across 3 values of the hyperparameter except for DT on pointmass where we tried more values, but still got poor results). Error bars are reported across seeds, as explained in the text.

Table 1: Shared hyperparameters for all non-DT algorithms

Hyperparameter	Value
Training steps	$5e5$
Batch size	256
MLP width	256
MLP depth	2

Table 2: Algorithm-specific hyperparameters for all non-DT algorithms

Algorithm	Hyperparameter	Value(s)
%BC	fraction ρ	[0.02, 0.10, 0.5]
	learning rate	1e-3
RvS	fraction of max return for conditioning	[0.8, 1.0, 1.2]
	learning rate	1e-3
TD3+BC	α	[1.0, 2.5, 5.0]
	learning rate (actor and critic)	3e-4
	discount	0.99
	τ for target EWMA	0.005
	target update period	2
IQL	expectile	[0.5, 0.7, 0.9]
	learning rate (actor, value, and critic)	3e-4
	discount	0.99
	τ for target EWMA	0.005
	temperature	10.0

Table 3: Hypereparameters for DT (exactly as in [8])

Hyperparameter	Value
Training steps	$1e5$
Batch size	64
Learning rate	1e-4
Weight decay	1e-4
K	20
Embed dimension	128
Layers	3
Heads	1
Dropout	0.1

Compute. All experiments were run on CPU on an internal cluster. Each of the non-DT algorithms takes less than 1 hour per run (i.e. set of hyperparameters and seed) and the DT algorithm takes 5-10 hours per run.

Table 4: Environment-specific reward targets for DT

Environment	Values
Point-mass	[300, 200, 100, 50, 10, 0]
Antmaze	[1.0, 0.75, 0.5]
Half-cheetah	[12000, 9000, 6000]
Pen	[3000, 2000, 1000]

Asset licenses. For completeness, we also report the licenses of the assets that we used in the paper: JAX [6]: Apache-2.0, Flax [14]: Apache-2.0, jaxrl [19]: MIT, Decision Transformer [8]: MIT, Deepmind control suite [28]: Apache-2.0, mujoco [29]: Apache-2.0, D4RL [10]: Apache-2.0.

Code. The code for our implementations can be found at `https://github.com/davidbrandfonbrener/rcsl-paper`.

E Potential negative societal impact

This paper follows a line work aiming at a better understanding of Offline RL algorithms. Even though it does not directly contribute to any specific application, it promotes the development and dissemination of the Offline RL technology, which, as any technology, can be used for harmful purposes. Moreover, we acknowledge that Offline RL has been proved in the past to lack robustness, and RL and even machine learning in general to potentially reproduce and amplify bias.

We note that this specific work attempts at better understanding the conditions for RCSL algorithms to work, and where it should not be used. In that spirit, it has the potential benefit of dissuading practitioners from using such algorithms in settings where. they may fail in socially undesirable ways.