

A APPENDIX

A.1 ANALYSIS FOR CONVERGENCE PROPERTIES

We firstly state **Theorem B.1** in (Kumar et al., 2019) as Theorem A.1 for the sake of completeness.

Theorem A.1. Suppose we run approximate distribution-constrained value iteration with a set constrained backup \mathcal{T}^Π on a set of policies Π . Let $\delta(s, a)$ be the upper-bound for the Bellman approximation error for a given state-action pair (s, a) over k training steps: $\delta(s, a) = \sup_k |Q_k(s, a) - \mathcal{T}^\Pi Q_{k-1}(s, a)|$. Then,

$$\lim_{k \rightarrow \infty} \mathbb{E}_{\rho_0} [|V_k(s) - V^*(s)|] \leq \frac{\gamma}{(1-\gamma)^2} \left[C(\Pi) \mathbb{E}_\mu \left[\max_{\pi \in \Pi} \mathbb{E}_\pi [\delta(s, a)] \right] + \frac{1-\gamma}{\gamma} \alpha(\Pi) \right]$$

with the suboptimality constant $(\alpha(\Pi))$ and the concentrability coefficient defined as:

$$\alpha(\Pi) = \max_{s,a} |\mathcal{T}^\Pi Q^*(s, a) - \mathcal{T} Q^*(s, a)| ; C(\Pi) \stackrel{\text{def}}{=} (1-\gamma)^2 \sum_{k=1}^{\infty} k \gamma^{k-1} c(k)$$

The proof of the theorem is a direct modification of the contraction proof in Theorem 3 of (Farahmand et al., 2010) or Theorem 1 of (Munos, 2003).

The *suboptimality constant* $(\alpha(\Pi))$ captures how far π^* is from Π , namely the suboptimality of the actor. The *concentrability coefficient* quantifies how far the visitation distribution generated by policies from Π is from the training data distribution, namely the degree to which the training may encounter OOD actions and states. (Kumar et al., 2019) note a trade-off between $\alpha(\Pi)$ and $C(\Pi)$, and propose to bound both terms by constraining Π to the set of policies with support the same as the training set policy with MMD loss.

However, the most important Bellman approximation error term which is the root cause of the bootstrapping problem is still left unbounded. We proceed to show that for $\pi'(a|s) = \frac{\beta}{\sup_k \sqrt{\text{Var}[Q_k(s, a)]}} \pi(a|s) / Z$. Assuming that $Z \geq 1$, and that Q is bounded, we can bound the Bellman error term $\max_{\pi'} \mathbb{E}_{\pi'} [\delta(s, a)]$ by any constant C with arbitrarily high probability by optimizing on π' .

Note that Theorem A.2 considers down-weighting by inverse the square-root of the variance (standard deviation), which is different from the inverse variance in Equation 3, 4, 5 and Algorithm 1. Down-weighting by the variance has the same practical effect since we clip the ratio for numerical stability. We adopt variance for practical implementation for the ease of tracing after multiple max, min, summation operations in Algorithm 1.

Theorem A.2. Let $\pi'(a|s) = \frac{\beta}{\sup_k \sqrt{\text{Var}[Q_k(s, a)]}} \pi(a|s) / Z(s)$, with the normalization factor $Z(s) = \int_a \frac{\beta}{\text{Var}[Q(s, a)]} \pi(a|s)$ as in equation 3. Assume that 1) $\forall s : Z(s) \geq 1$ and 2) Q is bounded ($\forall s, a : |Q(s, a)| \leq Q_m$).

Then for any $C \in \mathbb{R}$, there exists β, K such that

$$P \left(\max_{\pi'} \mathbb{E}_{\pi'} [\delta(s, a)] \geq C \right) \leq \frac{1}{K^2}$$

Proof. We firstly apply triangle inequality to unwrap the original formulation into a sum of two differences, and bound the two terms respectively.

$$\begin{aligned} \max_{\pi'} \mathbb{E}_{\pi'} [\delta(s, a)] &= \max_{\pi'} \mathbb{E}_{\pi'} \left[\sup_k |Q_k(s, a) - \mathcal{T}^\Pi Q_{k-1}(s, a)| \right] \\ &= \max_{\pi'} \mathbb{E}_{\pi'} \left[\sup_k |Q_k(s, a) + E[Q_k(s, a)] - E[Q_k(s, a)] - \mathcal{T}^\Pi Q_{k-1}(s, a)| \right] \\ &\leq \max_{\pi'} \mathbb{E}_{\pi'} \left[\sup_k |Q_k(s, a) - E[Q_k(s, a)]| \right] + \max_{\pi'} \mathbb{E}_{\pi'} \left[\sup_k |E[Q_k(s, a)] - \mathcal{T}^\Pi Q_{k-1}(s, a)| \right] \end{aligned}$$

Starting with the **red** term, we firstly obtain a probabilistic bound on the distance term inside the expectation with the Chebyshev’s inequality

$$P(|X - E[X]| \geq \sigma K) \leq \frac{1}{K^2}$$

$$P\left(|Q_k(s, a) - E[Q_k(s, a)]| \geq K \sqrt{\text{Var}[Q_k(s, a)]}\right) \leq \frac{1}{K^2}$$

$$P\left(\frac{\beta}{\sup_k \sqrt{\text{Var}[Q_k(s, a)]}} |Q_k(s, a) - E[Q_k(s, a)]| \geq \beta K\right) \leq \frac{1}{K^2}$$

Secondly, note that by assumption $|Q|$ is bounded by Q_m . This provides us an upper-bound on the difference $|Q(s, a) - E[Q(s, a)]| \leq 2Q_m$. Making use of both the general upper-bound and the tight probabilistic bound, by setting $\pi'(a|s) = \frac{\beta}{\sup_k \sqrt{\text{Var}[Q_k(s, a)]}} \pi(a|s)/Z(s)$, we have

$$\begin{aligned} \max_{\pi'} \mathbb{E}_{\pi'} \left[\sup_k |Q_k(s, a) - E[Q_k(s, a)]| \right] &= \max_{\pi'} \mathbb{E}_{\pi} \left[\frac{\beta}{\sup_k \sqrt{\text{Var}[Q_k(s, a)]}} \sup_k |Q_k(s, a) - E[Q_k(s, a)]| / Z(s) \right] \\ &\leq \left(1 - \frac{1}{K^2}\right) \beta K + \frac{2}{K^2} Q_m \leq B \end{aligned}$$

By assumption $Z(s) \geq 1$ and can be safely ignored. By picking the appropriate K and β , we can bound the **red** term by any constant $B \in \mathbb{R}$. The same bound holds for the **blue** term since $E[\mathcal{T}^\Pi Q_{k-1}(s, a)] = E[Q_k(s, a)]$. We therefore arrive at a constant bound for the Bellman error term $\max_{\pi'} \mathbb{E}_{\pi'} [\delta(s, a)]$. \square

Note that in Theorem A.2 Assumption 1) does not change the optimization problem in equation 4, 5 and Assumption 2) can be easily satisfied by imposing Spectral Norm on the Q function.

Now according to the constant bound on $\delta(s, a)$ from Theorem A.2 the convergence of our proposed framework follows directly from Theorem A.1 (Kumar et al., 2019; Farahmand et al., 2010; Munos, 2003), with the set of policies $\Pi' = \left\{ \pi' \mid \pi'(a|s) = \frac{\beta}{\sup_k \sqrt{\text{Var}[Q_k(s, a)]}} \pi(a|s)/Z(s), \pi \in \Pi \right\}$.

A.2 IMPLEMENTATION DETAILS

LunarLander: We set our expert to be a simple 3-layer actor-critic agent trained to completion with (Peng et al., 2019). We take the final replay buffer (size 100,000) with average reward of 269.7. The vertically clipped dataset in Figure 6 contains 76,112 samples, and the horizontally clipped dataset in Figure 3 contains 21,038 samples.

We then train a simple 3-layer actor-critic off-policy agent on the clipped datasets according to Algorithm 1 (we do not take the MMD loss in line 11 to enlarge the effect of OOD samples).

Baseline (BEAR): We ran benchmarks on the official GitHub code² of BEAR and the updated version³ provided by the authors. We ran parameter search on all the recommended parameters $\text{kernel_type} \in \{\text{gaussian}, \text{laplacian}\}$, $\text{mmd_sigma} \in \{10, 20\}$, 100 actions sampled for evaluation, and 0.07 being the $\text{mmd_target_threshold}$. We are able to reproduce the results reported in (Fu et al., 2020) with both the official GitHub and the updated version.

Our method (UWAC): We apply our weighted loss to Algorithm 1 to the updated BEAR code provided by (Kumar et al., 2019). We keep the hyper-parameters and the network architecture exactly the same as in BEAR. For experiments on the Adroit hand environment, we further enforce Spectral Norm on the Q function for better stability similar to (Yu et al., 2020) and theoretical guarantee as

²github.com/aviralkumar2907/BEAR

³github.com/rail-berkeley/d4rl_evaluations

shown in Appendix A.1. We clip the inverse variance to within the range of (0.0, 1.5) for numerical stability. For the choice of β in Algorithm 1. We swept over 3 beta values from the set $\{0.8, 1.6, 2.5\}$, determined by matching the average uncertainty output during training time. We found that the model is quite robust against betas: 0.8, 1.6 gave similarly good performance across all tasks in our experiments. We also note that β can be absorbed into the learning rate since it acts both on the actor loss and critic loss. However, since the MMD loss from BEAR is not β -weighted, we make the design choice to tune β in stead of the MMD weight α .

Ablations:

Our first study isolates the effect of Spectral Norm on agent performance. Although BEAR + Spectral Norm enforces a bounded Q function and maintains good training stability, Spectral Norm does not handle OOD backups on the narrow Adroit datasets. We discover experimentally that BEAR+SN performs much worse than BEAR only, we plot the complete results of BEAR+SN v.s. BEAR in Figure 9.

Our second study isolates the effect of Dropout on agent performance as a regularizer, since dropout alone does not handle OOD backups on the narrow Adroit datasets. We observe experimentally that UWAC without uncertainty weighing (BEAR+Dropout+Spectral Norm) does not change the behavior of BEAR under Spectral Norm (Figure 10) and performs worse than UWAC (Figure 11) and the original BEAR (Figure 12).

B FIGURES

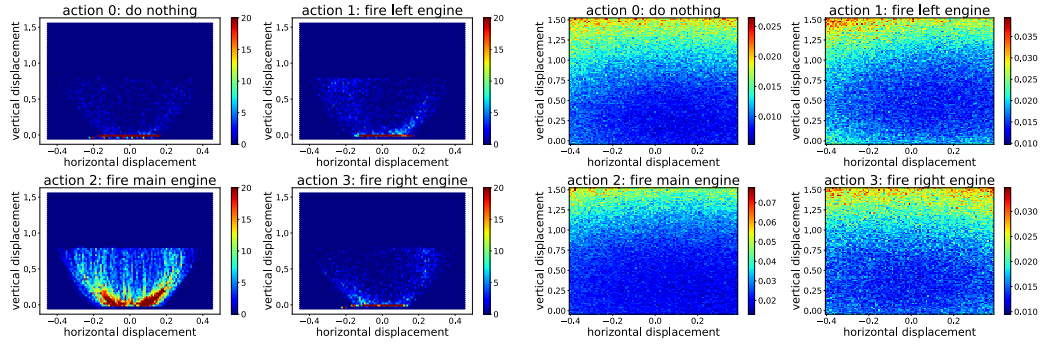


Figure 6: **Left.** The training dataset has observations with vertical displacements > 0.8 removed. This makes all states on the top OOD states. **Right.** Our model estimates higher uncertainty (brighter color) on the top and lower uncertainty (colder color) on the bottom.

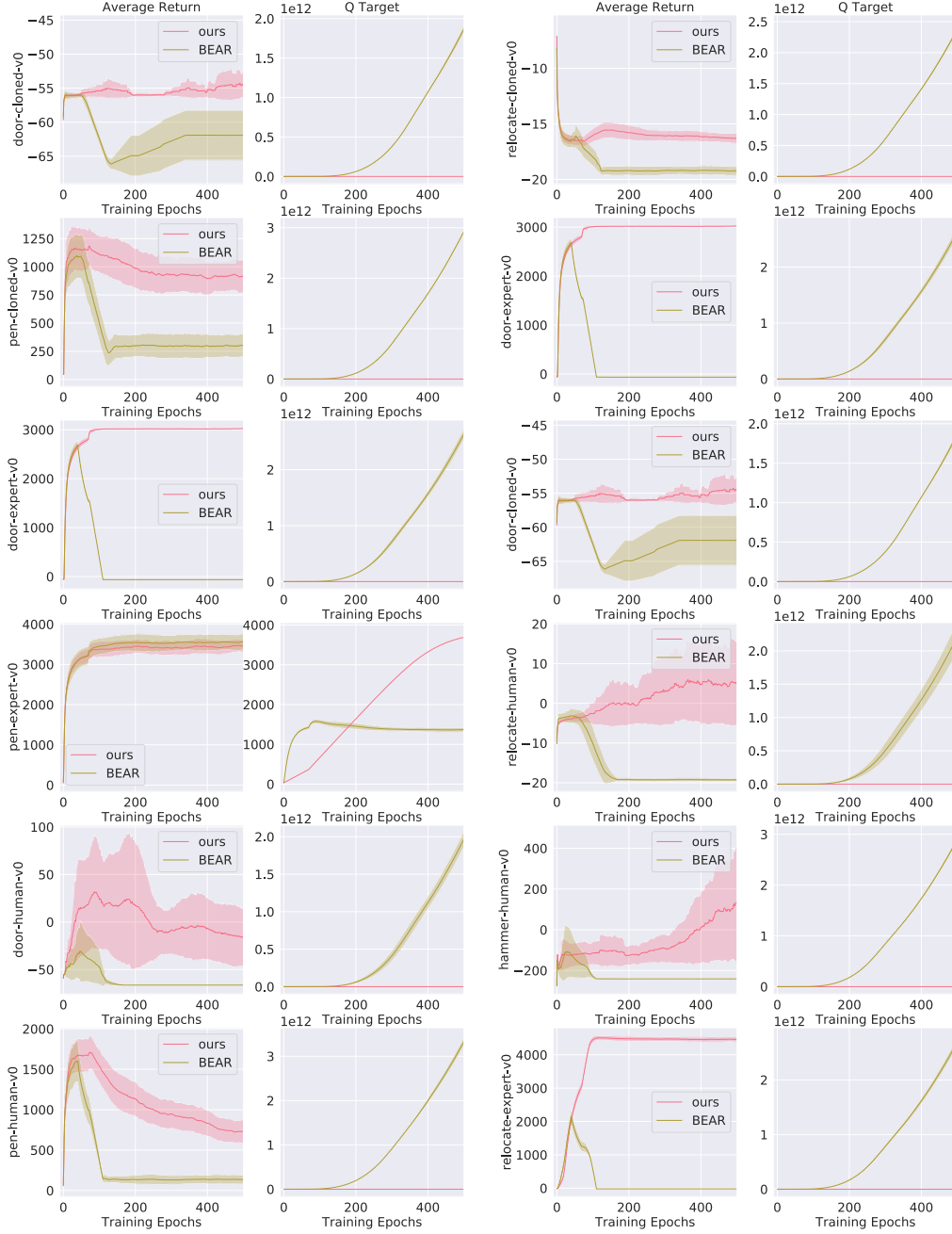


Figure 7: Plot of average return v.s. training epochs, together with the corresponding average Q Target over training epochs on the D4RL Adroit hand offline data set. Results are averaged across 5 random seeds. Note that the performance of baseline (BEAR) degrades over time (also noted in original paper [Kumar et al. \(2019\)](#)), and the Target Q value explodes. Our method, UWAC, achieves significantly better overall performance and training stability.

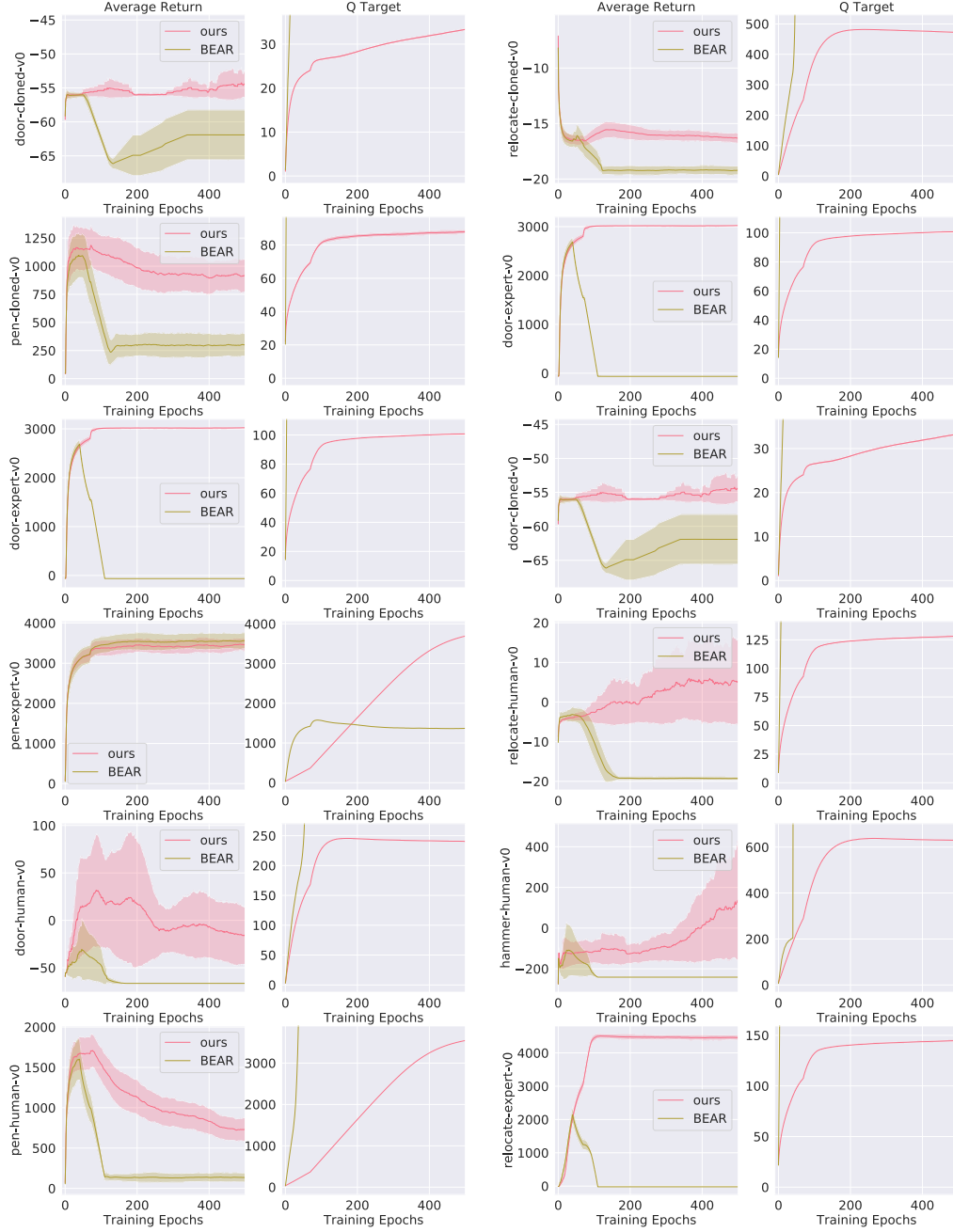


Figure 8: Plot of average return v.s. training epochs (zoomed-in). The figure is the same as [7](#) except that the second column is zoomed-in on the Q values of the UWAC critic.

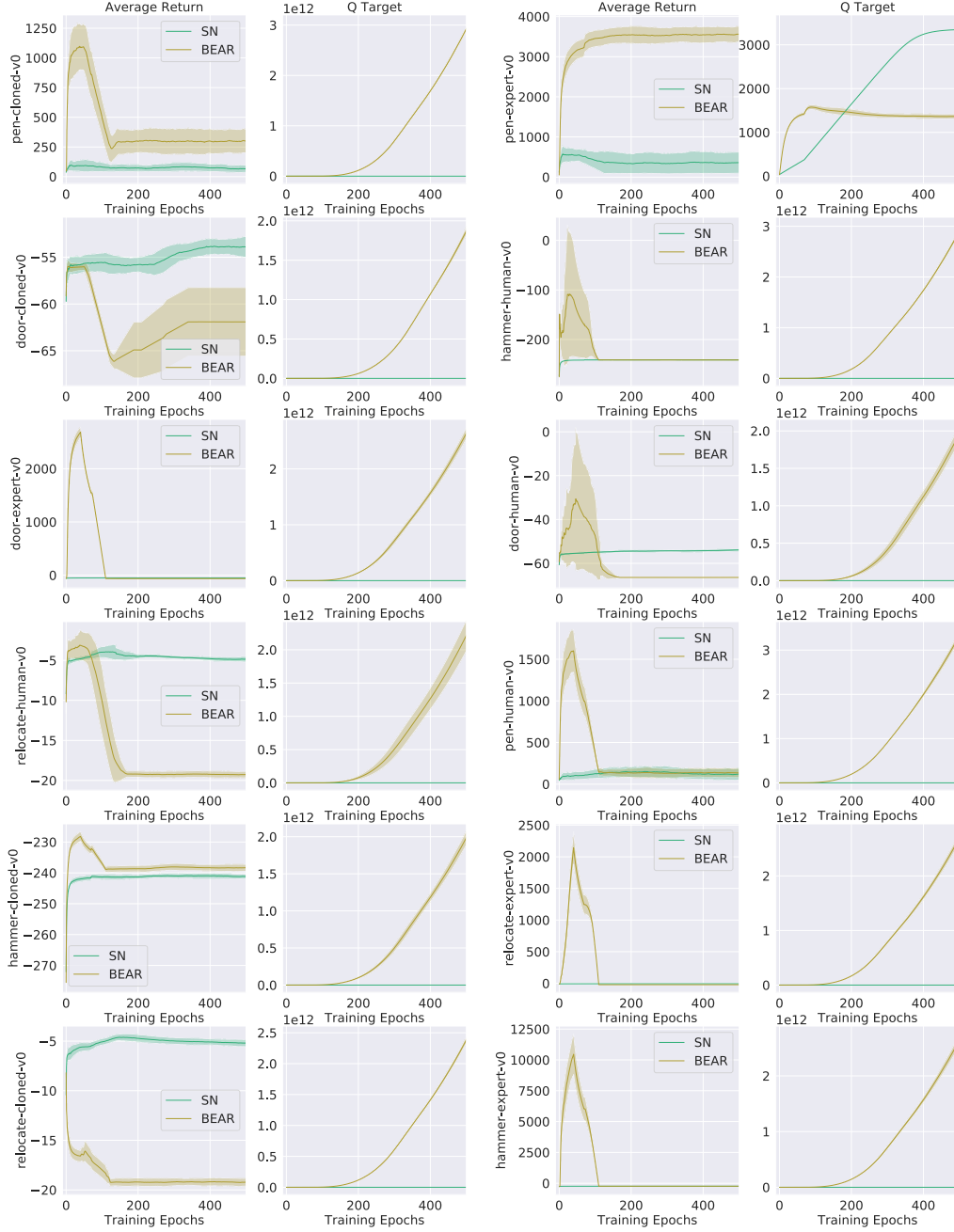


Figure 9: **Ablation:** Plot of average return v.s. training epochs for BEAR v.s. BEAR+Spectral Norm, together with the corresponding average Q Target over training epochs on the D4RL Adroit hand offline data set. Results are averaged across 5 random seeds. Although BEAR with Spectral Normalized Q function maintains stable Q estimate during training, BEAR with SN achieves significantly worse training performance in terms of average return.

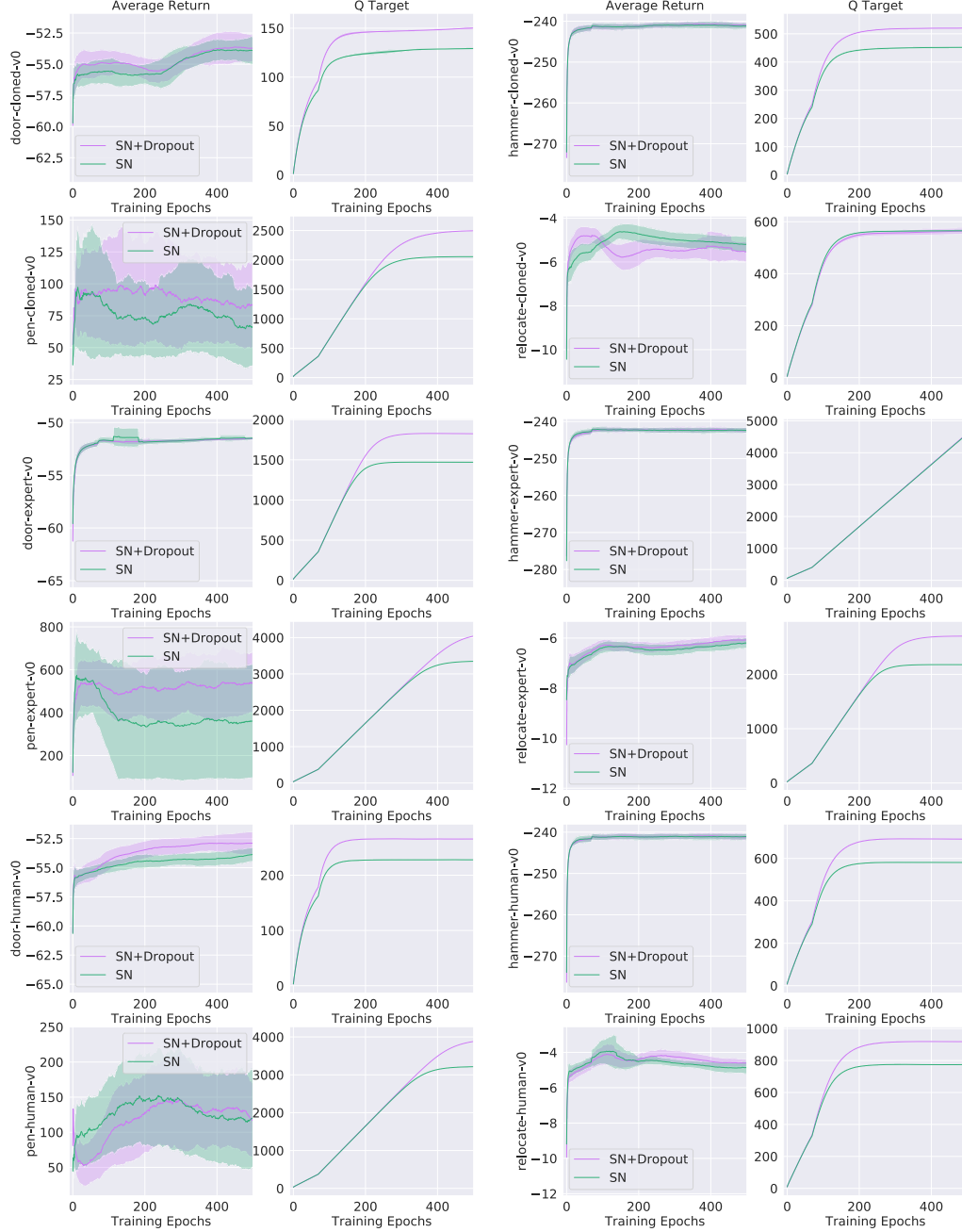


Figure 10: **Ablation:** Plot of average return v.s. training epochs for BEAR+Spectral Norm v.s. BEAR+Dropout+Spectral Norm, together with the corresponding average Q Target over training epochs on the D4RL Adroit hand offline data set. The results are averaged across 5 random seeds. Without the UWAC reweighing loss, performing dropout on the Q function does not lead to improved performance.

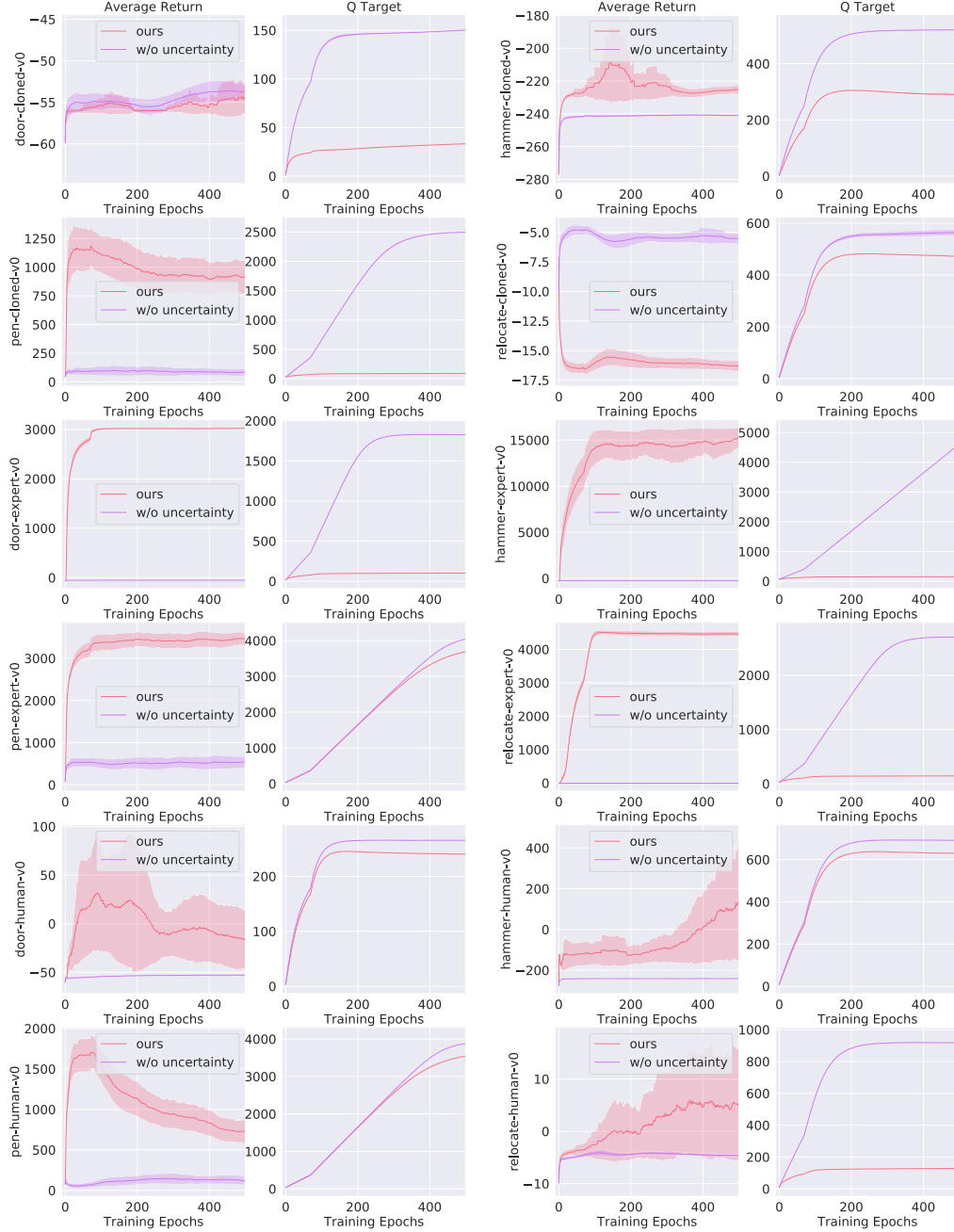


Figure 11: **Ablation:** Plot of average return v.s. training epochs for UWAC (ours) v.s. ours without uncertainty weighing but with dropout in the Q function, together with the corresponding average Q Target over training epochs on the D4RL Adroit hand offline data set. The results are averaged across 5 random seeds. Without the weighing loss, performance of the agent drops drastically. Note that low performance on hammer-cloned, door-cloned, and relocate-cloned may be attributed to the bad quality of the datasets caused by data collection (explained in section 5.3)

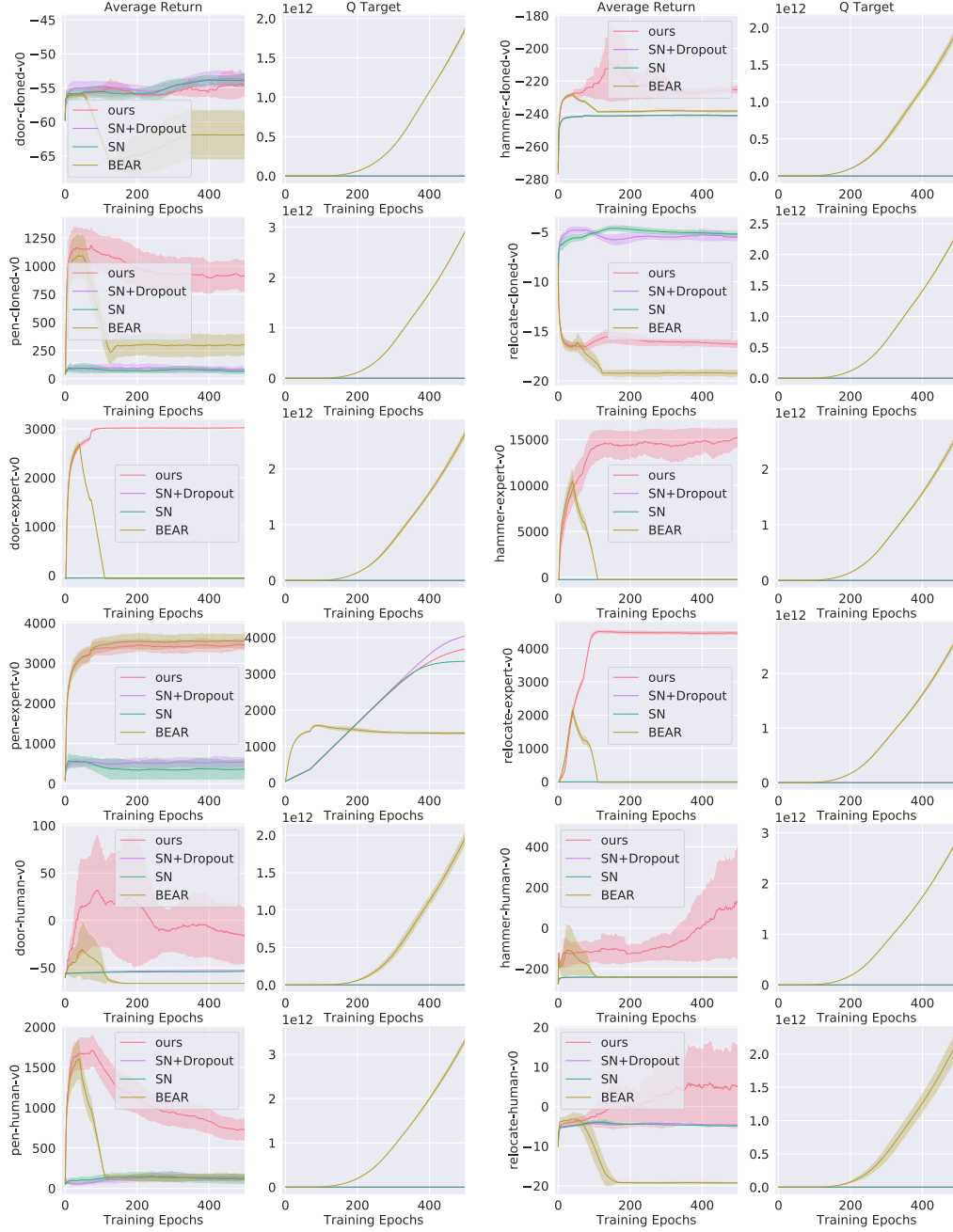


Figure 12: **Ablation:** Figure 9, 10, 11 plotted together. Note that SN+Dropout (purple) is also denoted as ours-w/o-uncertainty in Figure 11.

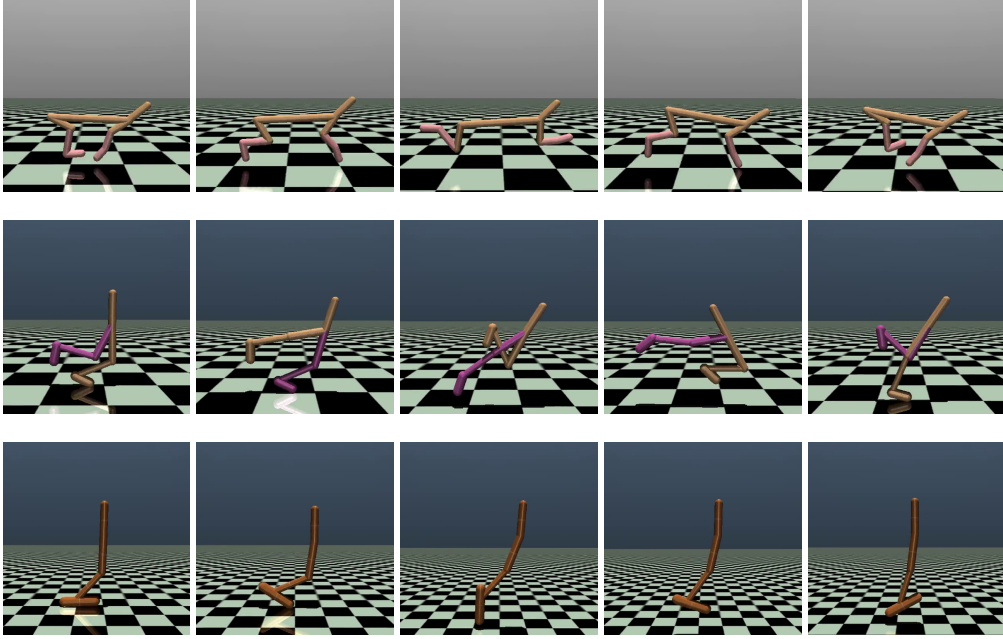


Figure 13: Sequences of our offline agent trained from expert demonstrations executing learned policies performing on the halfcheetah, walker2d, and hopper tasks in the MuJuCo Gym environment. See the videos attached in the supplementary.

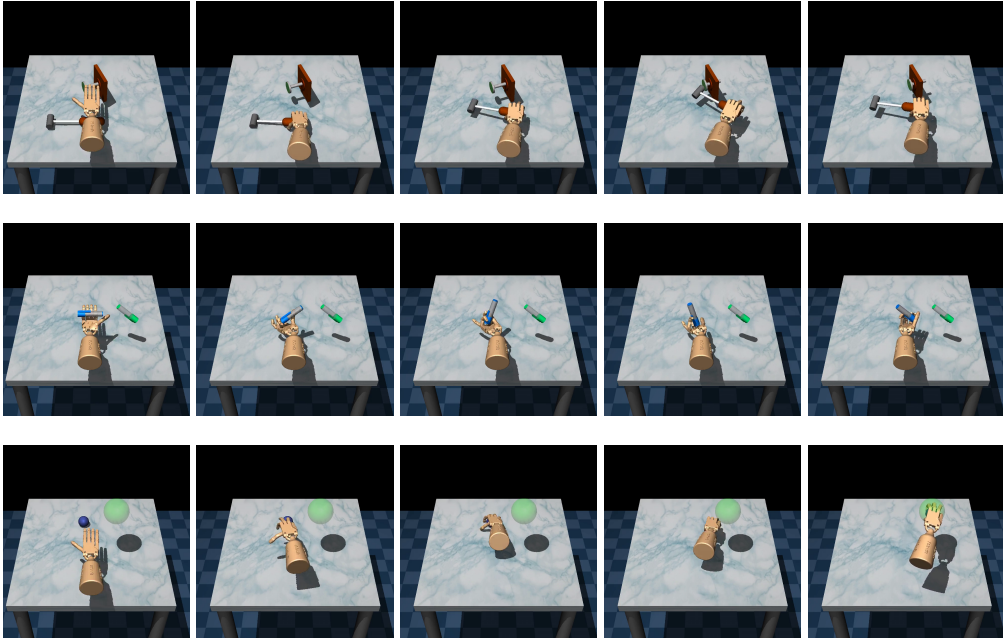


Figure 14: Sequences of the agent trained from human demonstrations executing learned policies performing the Adroit tasks of hammering a nail, twirling a pen and picking/moving a ball. The task of opening a door is shown in Figure 4. See the videos attached in the supplementary.