A. Appendix

A.1. Analysis for Convergence Properties

We first extend the convergence theorem (Theorem B.1) from (Kumar et al., 2019) to obtain a bound on the approximation error with respect to bellman approximation error $\delta(s,a)$ – Theorem A.1.

Note that Theorem A.1 alone does not imply convergence, since $\delta(s,a)$ can be arbitrarily large due to OOD estimates in offline RL. We then show (in Theorem A.2) that $\delta(s,a)$ can be bounded by any constant under mild assumptions that the Q function is Lipschitz continuous. Theorem A.2 shows our framework converges w.r.t. OOD samples.

Theorem A.1. *Suppose we run approximate distribution-constrained value iteration with a set constrained backup $\mathcal{T}^{\Pi}$ 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: $\delta(s,a) = \sup_k \left|Q_k(s,a) - \mathcal{T}^{\Pi}Q_{k-1}(s,a)\right|$. Then,*

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

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

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

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 $\pi'(a|s) = \frac{\beta}{\sup_k\sqrt{Var[Q_k(s,a)]}}\pi(a|s)/Z$. 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-weighing 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 $\pi'(a|s) = \frac{\beta}{\sup_k\sqrt{Var[Q_k(s,a)]}}\pi(a|s)/Z(s)$, with the normalization factor $Z(s) = \int_a \frac{\beta}{Var[Q(s,a)]}\pi(a|s)$ 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

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

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

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

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

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

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

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 $\pi'(a|s) = \frac{\beta}{\sup_k \sqrt{Var[Q_k(s,a)]}} \pi(a|s)/Z(s)$, we have

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

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 $\max_{\pi'} \mathbb{E}_{\pi'} [\delta(s,a)]$. □

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 $\Pi' = \left\{ \pi' \mid \pi'(a|s) = \frac{\beta}{\sup_k \sqrt{Var[Q_k(s,a)]}} \pi(a|s)/Z(s), \pi \in \Pi \right\}$.

A.2. Training Time of MC Dropout

Since the most time is spent during training is at communication between the GPU and CPU when performing roll-outs for evaluation. UWAC with dropout takes less than 1.5 times the training time of BEAR, with 100 times the original batchsize calculating sample uncertainty (in parallel on a single GPU).

A.3. Observations on the Q Value Divergence of BEAR

We tested BEAR learning rate from $\{10^{-3}, 10^{-4}, 10^{-5}\}$, the divergence behavior did not change. The action support constraint of BEAR is helpful, but it relies on the MMD approximation which is not perfect. Intuitively, UWAC provides a complementary enforcement to the action support constraint, where OOD actions that survive the MMD loss are further penalized.

B. Figures

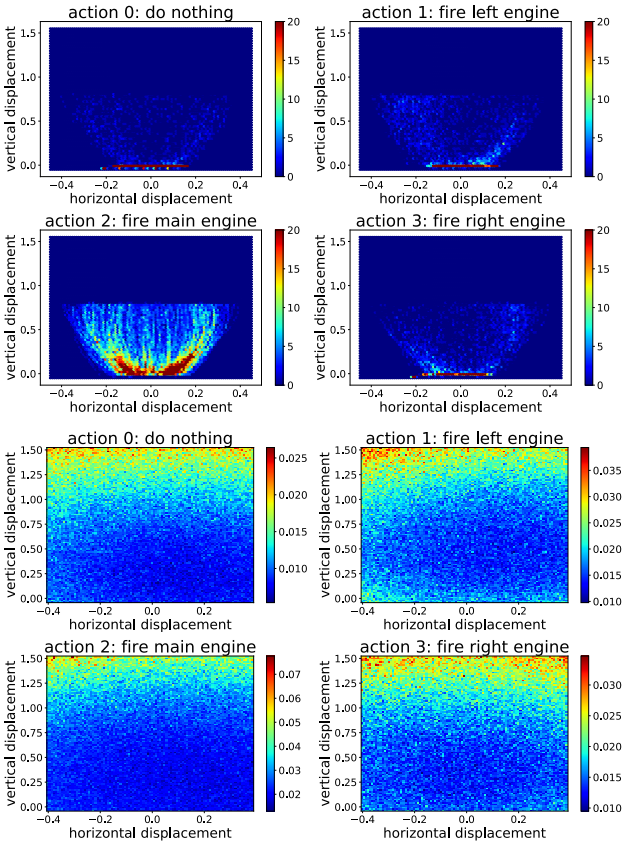

Figure 7: **Top.** The training dataset has observations with vertical displacements > 0.8 removed. This makes all states on the top OOD states. **Bottom.** Our model estimates higher uncertainty (brighter color) on the top and lower uncertainty (colder color) on the bottom.

-

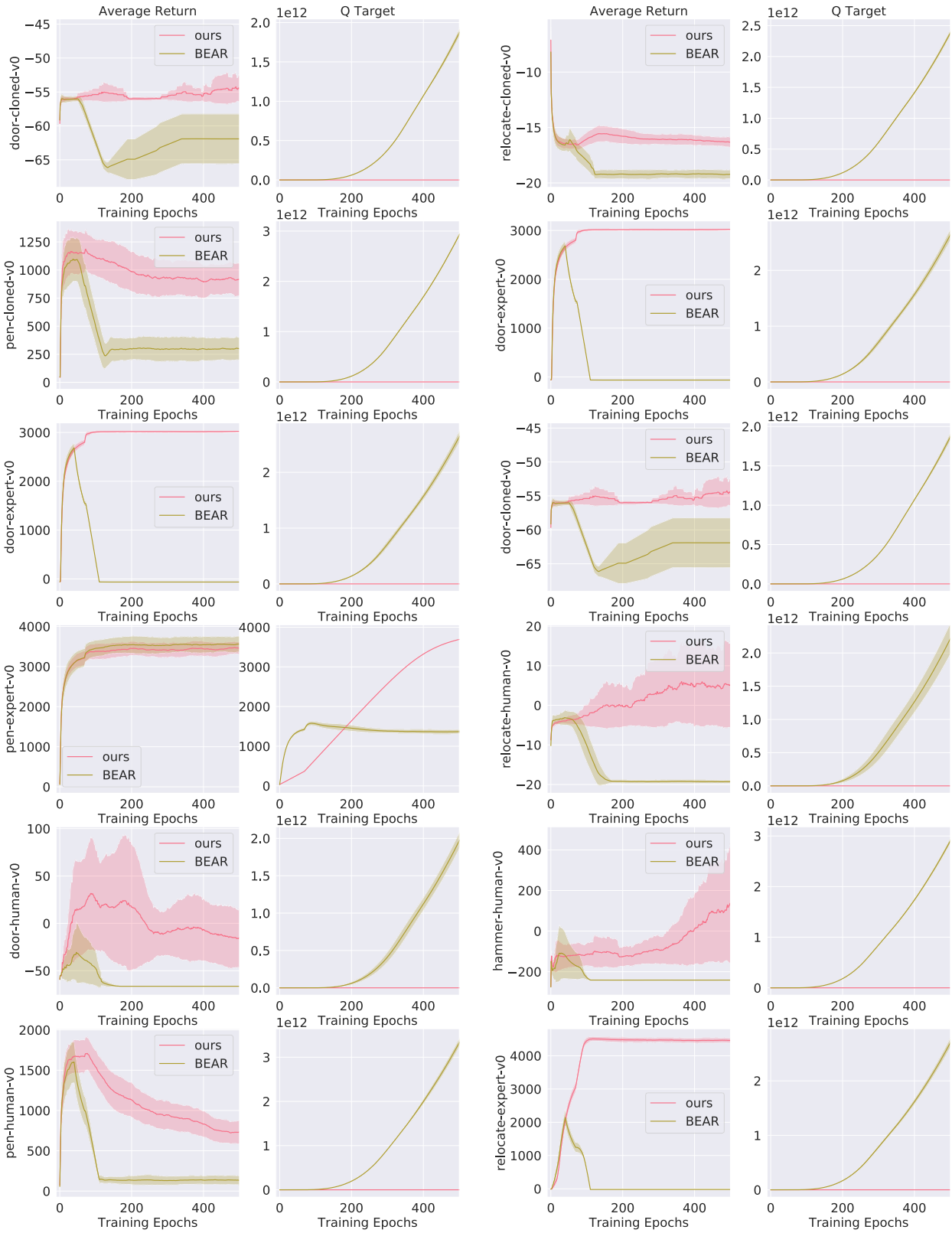

Figure 8: Plot of average return v.s. training epochs, together with the corresponding average Q Target over training epochs on the D4RL Adroit hand offline data set. Results are averaged across 5 random seeds. Note that the performance of baseline (BEAR) degrades over time (also noted in original paper (Kumar et al., 2019)), and the Target Q value explodes. Our method, UWAC, achieves significantly better overall performance and training stability.

-

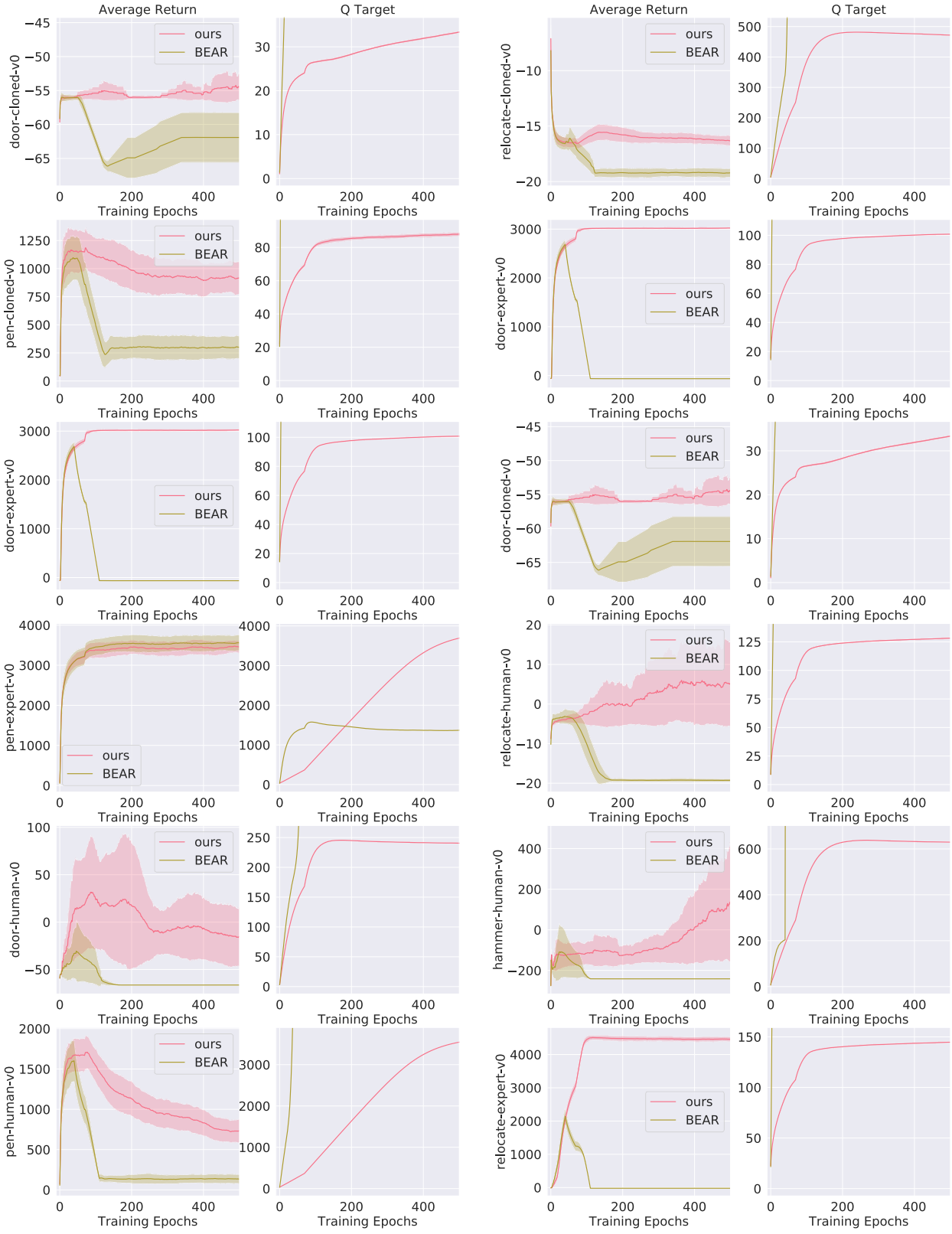

Figure 9: Plot of average return v.s. training epochs (zoomed-in). The figure is the same as 8, except that the second column is zoomed-in on the Q values of the UWAC critic.

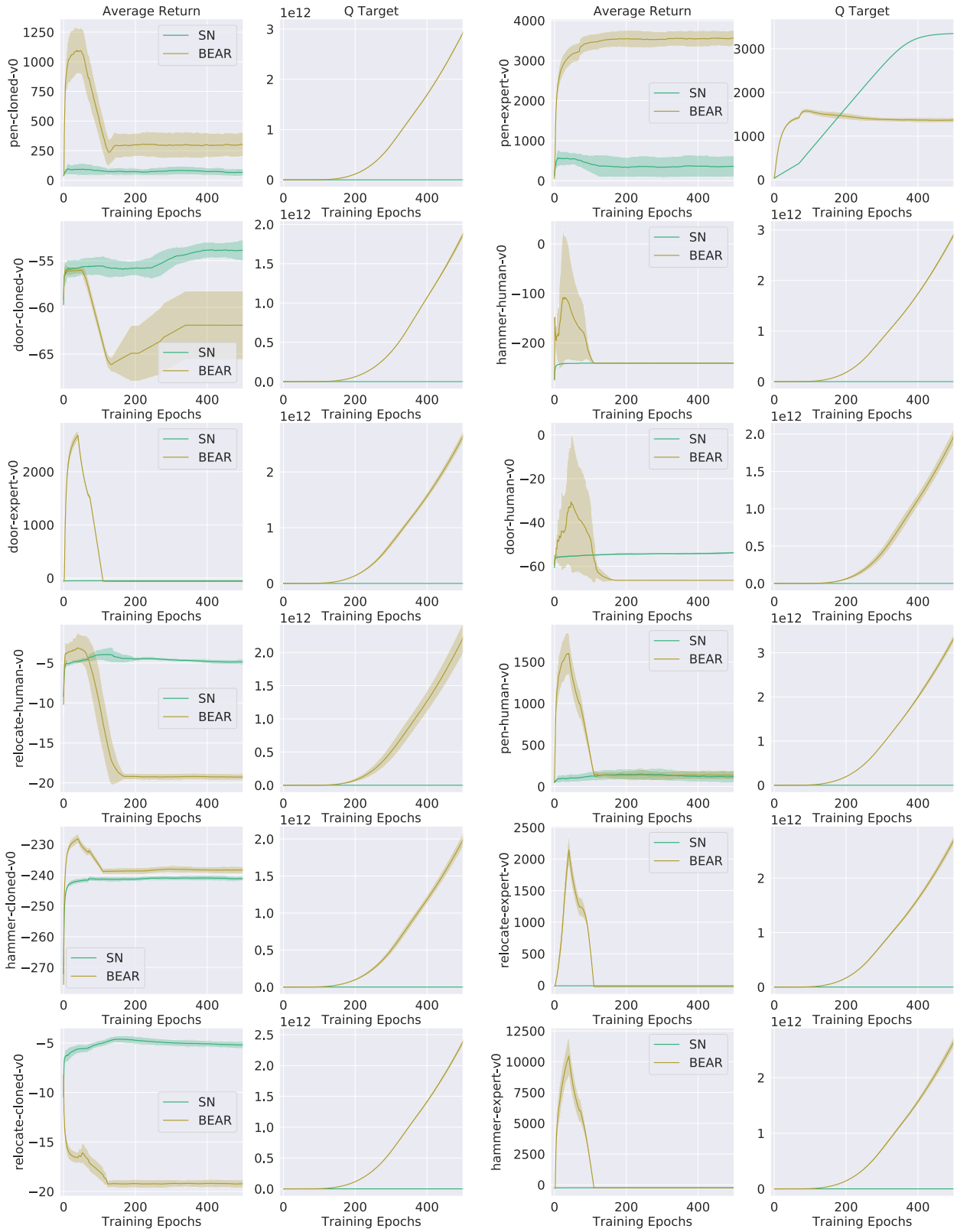

Figure 10: **Ablation:** Plot of average return v.s. training epochs for BEAR v.s. BEAR+Spectral Norm, together with the corresponding average Q Target over training epochs on the D4RL Adroit hand offline data set. Results are averaged across 5 random seeds. Although BEAR with Spectral Normalized Q function maintains stable Q estimate during training, BEAR with SN often achieves significantly worse training performance in terms of average return.

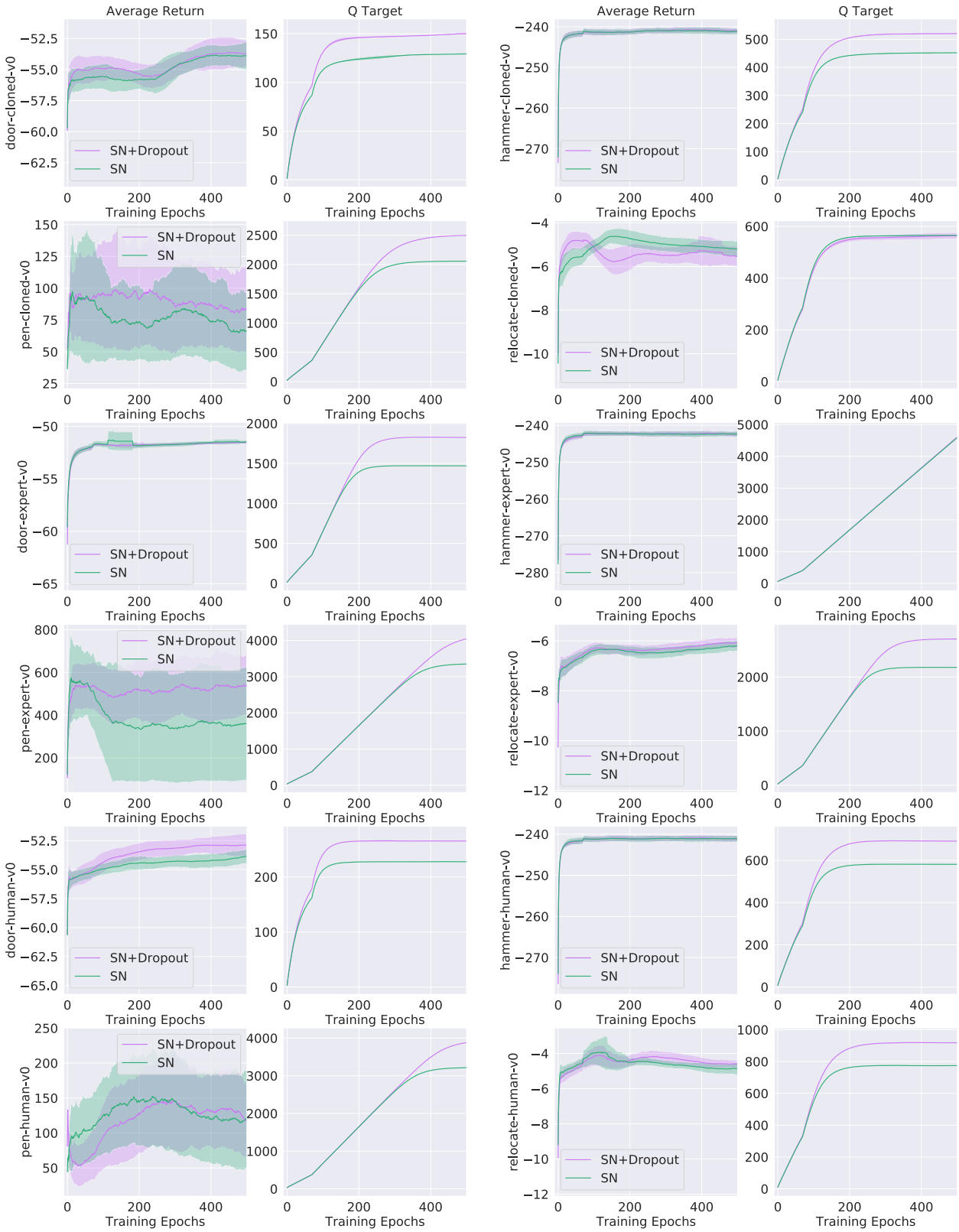

Figure 11: **Ablation:** Plot of average return v.s. training epochs for BEAR+Spectral Norm v.s. BEAR+Dropout+Spectral Norm, together with the corresponding average Q Target over training epochs on the D4RL Adroit hand offline data set. The results are averaged across 5 random seeds. Without the UWAC reweighing loss, performing dropout on the Q function does not lead to improved performance.

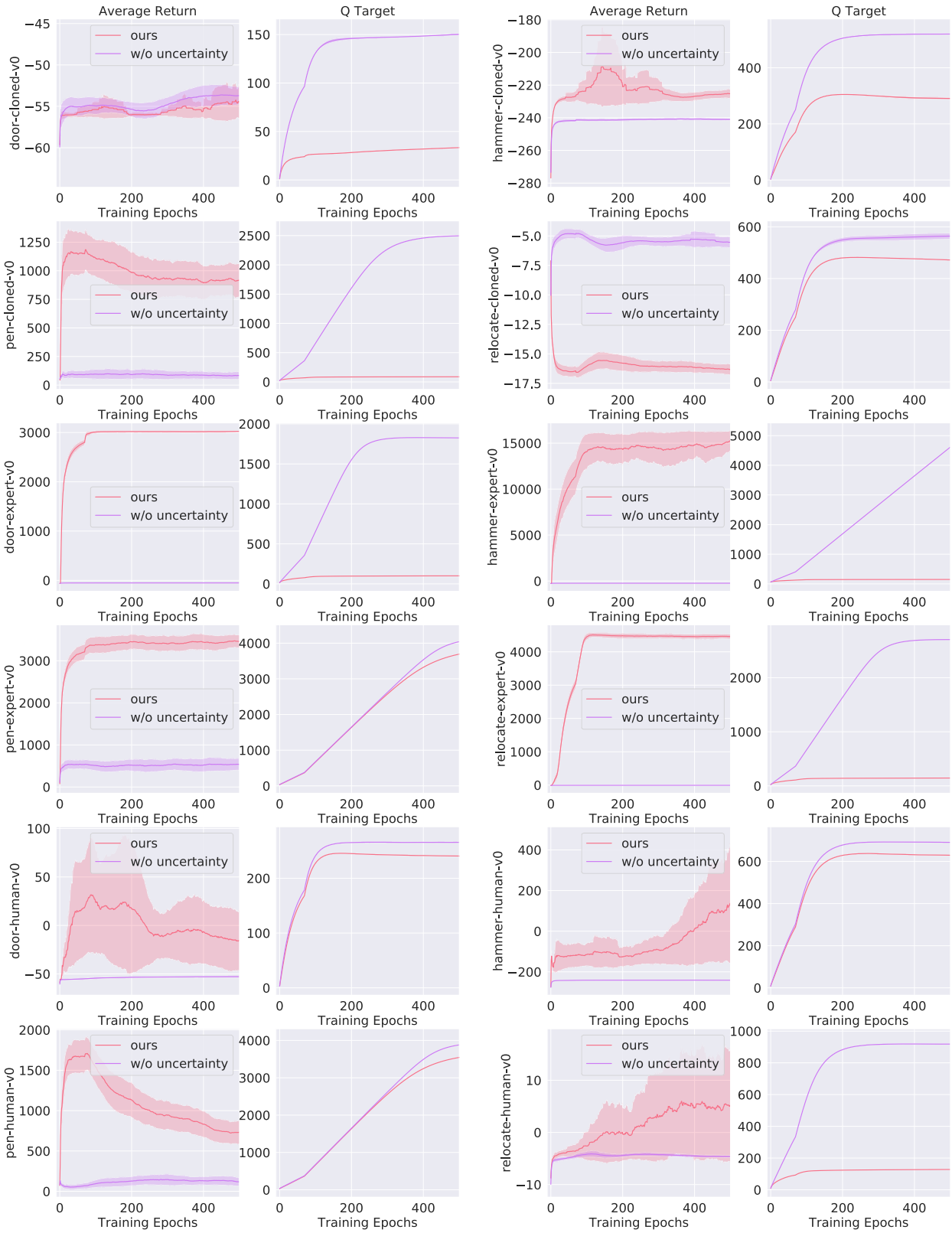

Figure 12: **Ablation:** Plot of average return v.s. training epochs for UWAC (ours) v.s. ours without uncertainty weighing but with dropout in the Q function, together with the corresponding average Q Target over training epochs on the D4RL Adroit hand offline data set. The results are averaged across 5 random seeds. Without the weighing loss, performance of the agent drops drastically. Note that low performance on hammer-cloned, door-cloned, and relocated-cloned may be attributed to the bad quality of the datasets caused by data collection (explained in section 5.3)

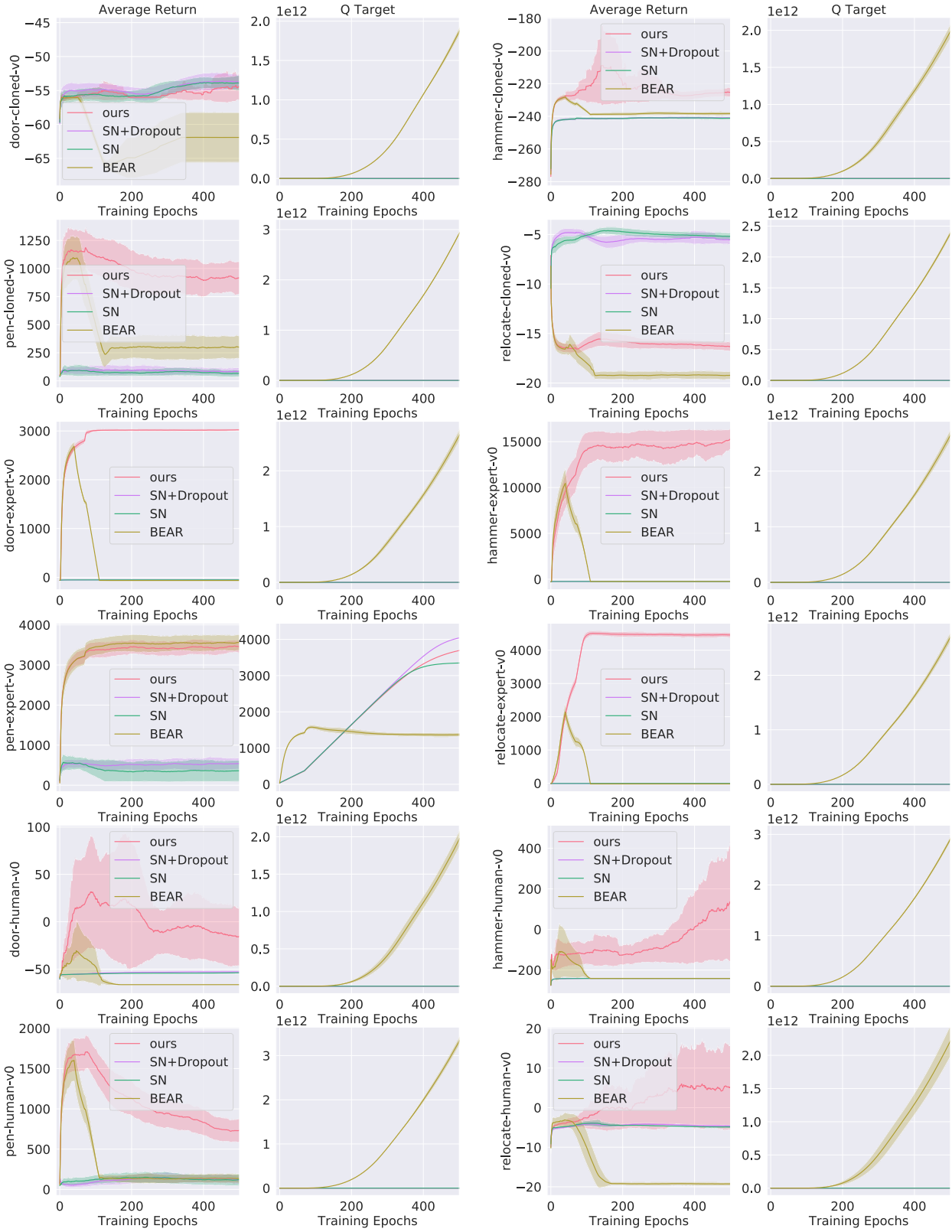

Figure 13: **Ablation:** Figure 10, 11, 12 plotted together. Note that SN+Dropout (purple) is also denoted as ours-w/o-uncertainty in Figure 12.

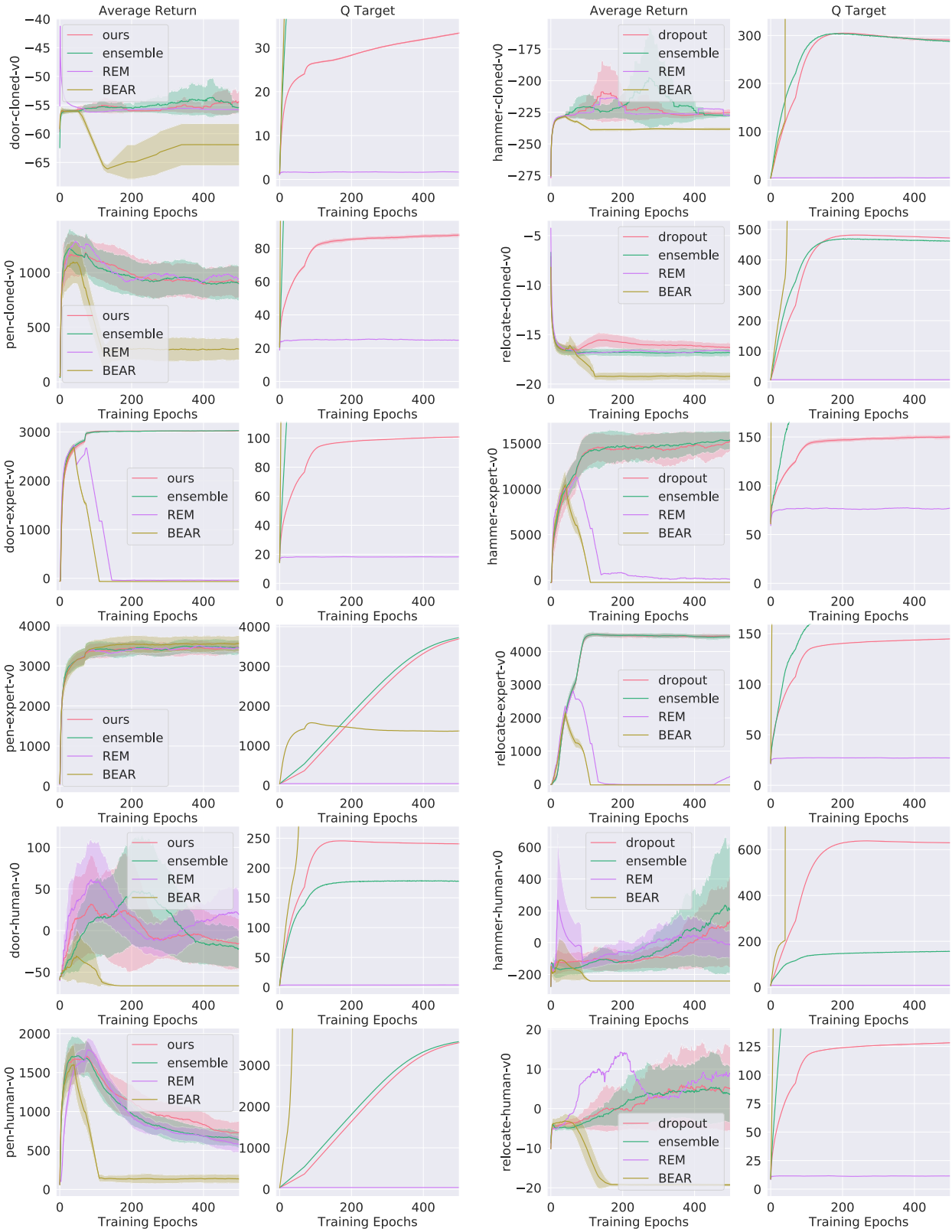

Figure 14: **Ablation:** Plot of UWAC under dropout (ours), Averaged-DQN ensembles, REM against baseline BEAR.

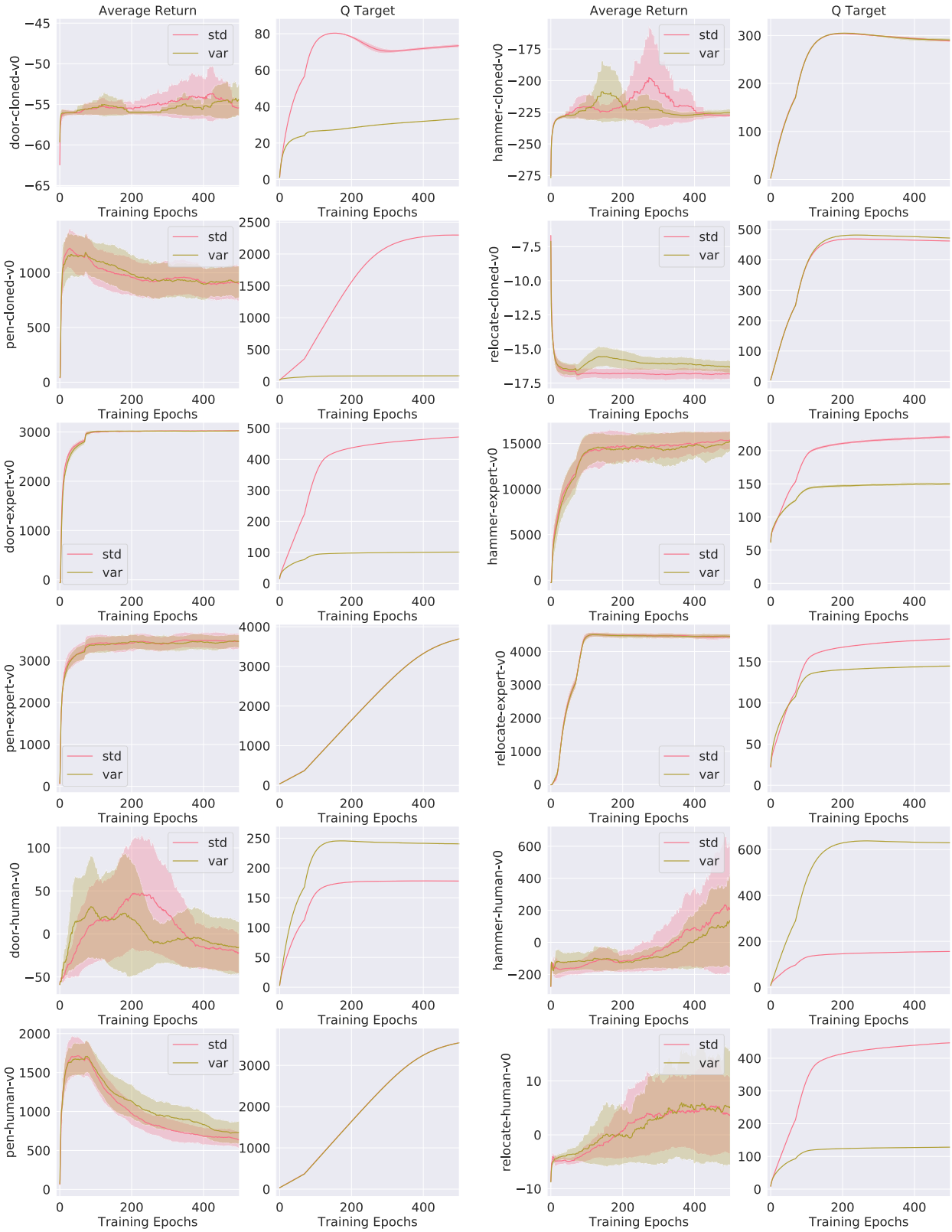

Figure 15: **Ablation:** Plot of UWAC with variance down-weighing v.s. standard deviation down-weighing.

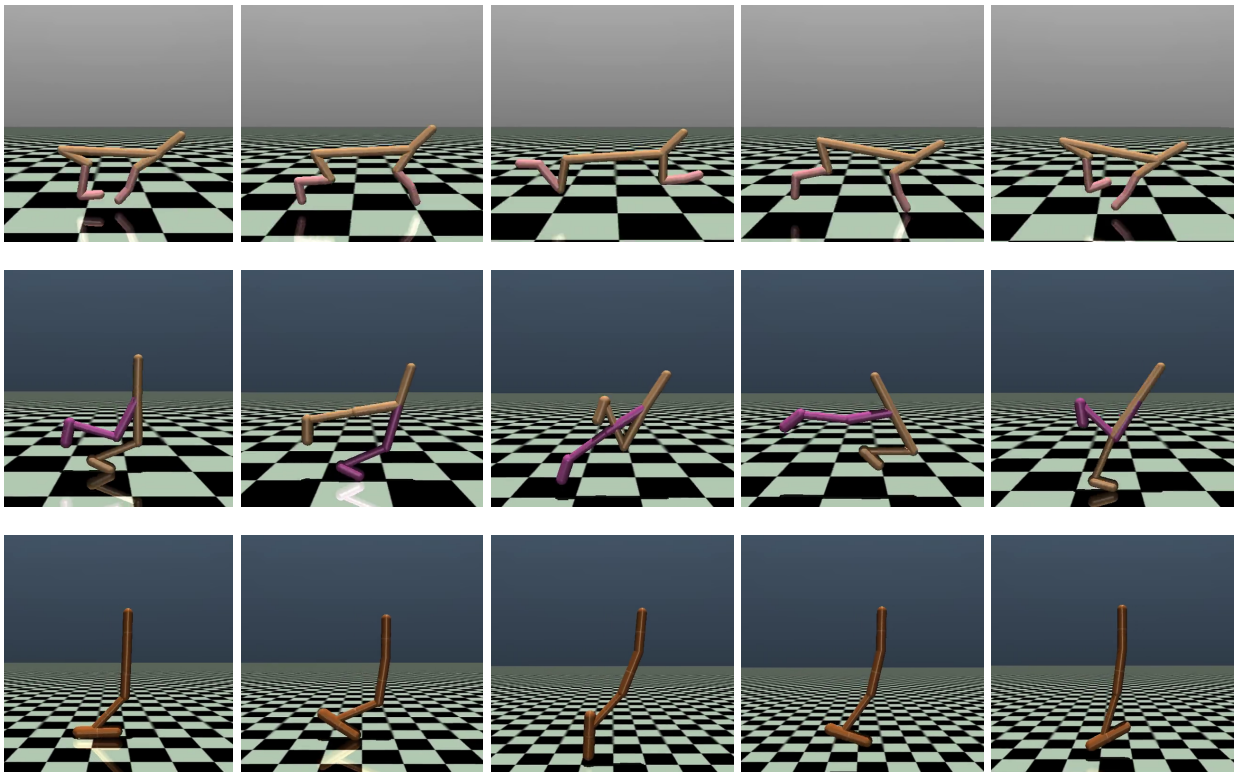

Figure 16: Sequences of our offline agent trained from expert demonstrations executing learned policies performing on the halfcheetah, walker2d, and hopper tasks in the MuJuCo Gym environment. See the videos attached in the supplementary.

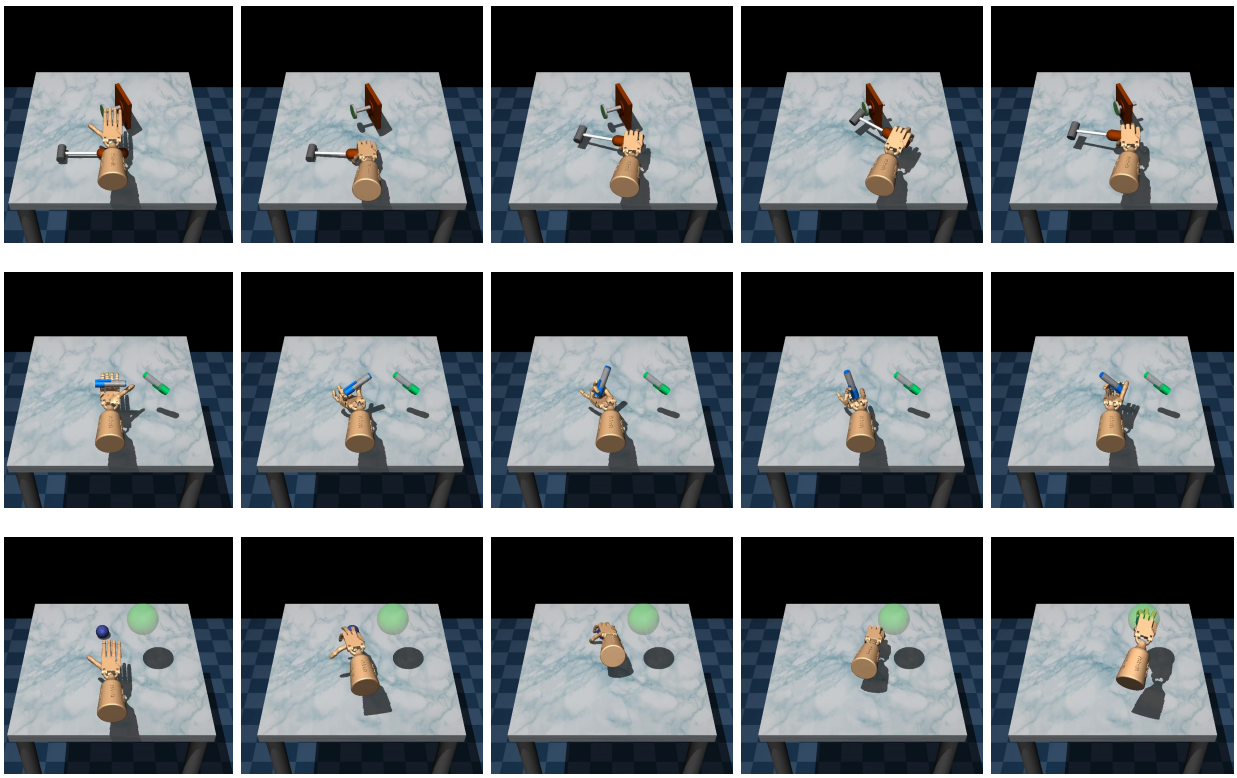

Figure 17: Sequences of the agent trained from human demonstrations executing learned policies performing the Adroit tasks of hammering a nail, twirling a pen and picking/moving a ball. The task of opening a door is shown in Figure 5. See the videos attached in the supplementary.