6 Supplementary materials

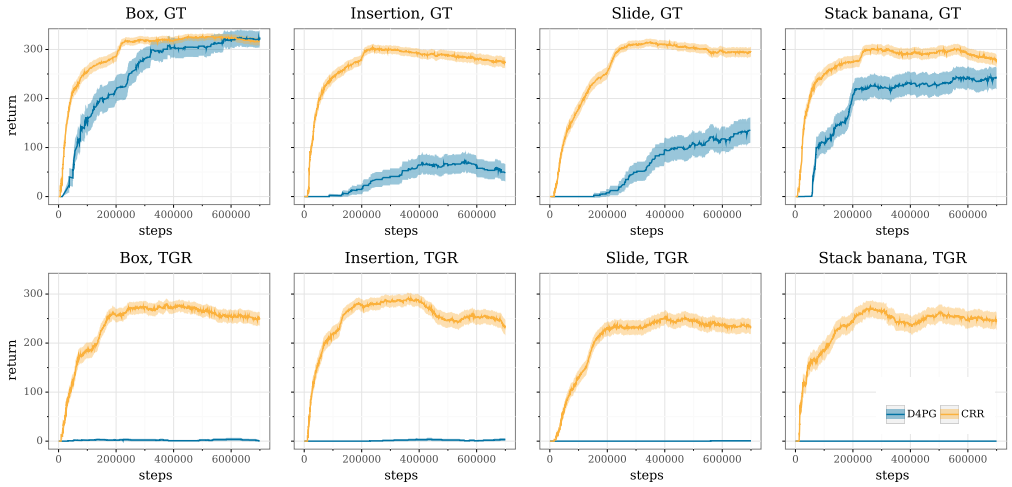

Figure 7: Results of policy training with two offline RL algorithms: CRR and D4PG for 4. The first row shows the performance when using the ground truth reward signal and the second row is with a learnt reward function. The performance of CRR with a learnt reward function stays close to the **GT**, however, D4PG with learnt reward fails completely.

Reward learning and different offline RL algorithms As the first step towards understanding how the type of offline RL algorithm influences the performance of the model with learnt reward, we conduct the following experiment. We study the performance of D4PG algorithm [32, 4] (which was used in the related work by Cabi et al. [6]) and CRR trained with ground truth rewards and learnt reward model. The first row of Fig. 7 shows the performance of CRR and D4PG when relying on the ground truth reward signal. Although CRR is clearly more efficient than D4PG, D4PG still reaches reasonable scores in two tasks and non-zero scores in two other tasks. The second row shows the performance of both algorithms with one of the learnt reward models. While CRR still attains reasonably high scores in all tasks, D4PG struggles to learn any useful policy from this imprecise reward signal. Our intuitive explanation of such difference in the behaviour of two offline algorithms

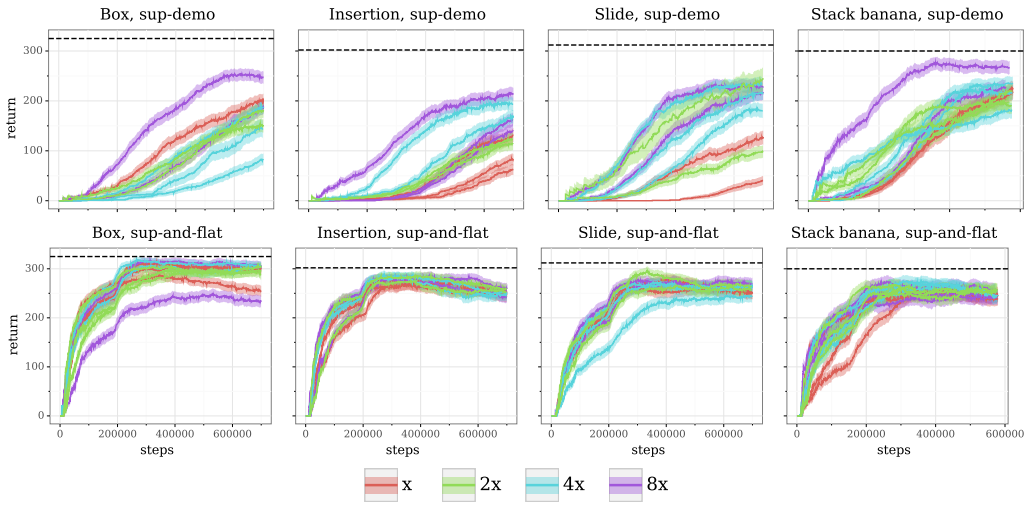

Figure 8: Results of policy training for 4 tasks using two reward model: **sup-demo** (first row) and **sup-and-flat** (second row) trained with varying amount of supervision. The performance of **sup-demo** can be potentially improved with more timesteps-level annotations, but the performance of **sup-and-flat** is almost indifferent to the amount of direct supervision and superior to **sup-demo**.

is that CRR only relies on rewards only when learning a critic and for acting it uses a loss similar to BC. This makes it more robust to the errors in reward predictions. However, further investigation is needed to understand how reward models inter-plays with policy training.

Varying amount of timestep-level annotations We look at the performance of methods with timestep-level annotations with the increasing amount of supervision. Fig. 8 shows the performance of **sup-demo** in the first row and the second row shows the performance of **sup-and-flat**. We use growing amounts of data: 1) the same amount of data as in Sec. 3.3 (**x**), 2) twice more (**2x**), 3) four times more (**4x**), and 4) eight times more (**8x**). The curves of the same colour correspond to different hyperparameters as discussed in Sec. 4.4. When we only use direct reward supervision in **sup-demo**, we notice that although the variance of the results with different hyperparameters is significant, it is possible in general to improve the performance substantially, but seldom up to the level of **sup-and-flat**. The performance of the policy trained with **sup-and-flat** seems to be much less affected by the amount of timestep-level supervision and works equally well with **x** and **8x** annotations. Thus, this **sup-and-flat** training algorithm is well adapted to working with limited supervision.