

## A Implementation Details

All experiments shown in this paper are conducted on a 10-core Intel i7 3.0 GHz desktop with 64 GB RAM and one GeForce GTX 1080 GPU. In our implementation of the PPO algorithm [38], we use a three hidden-layer fully-connect neural network with (128, 64, 32) units in each layer for both the policy network and the value network, and set  $\gamma = 0.99$  and  $\lambda = 0.95$ . We noticed that PPO training can be unstable due to the frequent curriculum switches especially in the block stacking environment, and found that in order to prevent collapsing during training, it is very helpful to use a small importance ratio clipping parameter in PPO (denoted as  $\epsilon$  in [38]) together with an optimizer with small learning rates and gradient clipping. In pick-and-place tasks, we set  $\epsilon = 0.2$  and use the Adam optimizer with a learning rate of  $2e-4$  without gradient clipping. In stacking tasks, we set  $\epsilon = 0.05$  and use the Adam optimizer with a learning rate of  $1e-4$  and gradient clipping-by-norm with a clipping factor of 0.05. In our DDPG [39] implementation, the learning rate is set to  $2e-4$ , and the same policy network is used as in the PPO implementation. The target update period in DDPG is set to 5.

In the ACED algorithm, we use  $\phi = 0.9$  as the curriculum switching threshold in pick-and-place tasks, and  $\phi = 0.85$  in block stacking tasks. The average return checking period is set to  $t = 120$ , and the number of episodes used to compute the average return is set to  $n = 3$ . 60 parallel rollout workers are used in both tasks. In pick-and-place tasks, the threshold for the object’s distance to the goal to assign reward  $r = 1$  is 0.05, and in stacking tasks the threshold is set to 0.04. The maximum number of steps in an episode is set to 50 in pick-and-place tasks, and 100 in block stacking tasks. In BC, an Adam optimizer with the learning rate of  $2e-4$  is used, and the loss function is negative log likelihood.

In the reverse curriculum implementation, we use 1000 random start states and a time horizon of 5 time steps for the Brownian motion. 200 old sampled start states are appended to the new start states at each training step. In our implementation of the Montezuma’s Revenge method, we randomly select one demonstration trajectory and set the curriculum switching threshold also to  $\phi = 0.85$ . Both the reverse curriculum method and the Montezuma’s Revenge method are implemented with the same PPO algorithm as used in the ACED implementation.

## B Additional Results

Due to limited space, additional experimental results are presented here in the Appendix. Since each experiment is terminated after convergence, the lengths of the learning curves may vary.

### B.1 Learning Curves

In order to show the learning progress and curriculum switches when using ACED, we use the pick-and-place task with 5 demonstration trajectories as an example to compare the learning curves of different algorithms, as shown in Figure 5. For each algorithm, we select one run whose convergence environment step is close to the mean for all 10 runs instead of directly using the mean in order to clearly show the learning progress and the curriculum switches. From Figure 5 we can see that for all ACED runs with PPO, the first few curricula are usually much easier than the last few and the majority of training time is spent on training the last few curricula. Without BC, the performance drop during curriculum switches is more obvious. If we compare the performance of ACED with DDPG and ACED with PPO, we can observe that ACED achieves a much higher sample efficiency with the off-policy DDPG.

All ACED runs are able to converge to almost 100% success rate, whereas vanilla PPO without ACED is not able to achieve a success rate higher than 10% during training. In addition to the comparison with vanilla PPO, we also compare ACED with Hindsight Experience Replay (HER) [33] in the block pick-and-place task. We use the OpenAI Baseline [42] implementation of HER with 2 MPI processes with 30 parallel environments each to make sure it is equivalent to the 60 parallel environments in other experiments. Other parameters for HER are set to default. However, all 10 runs with HER are only able to achieve a success rate of about 50%, and we show one representative learning curve in Figure 5. This is because in the Gym FetchPickAndPlace-V1 task, half of the goals are sampled from on the table and half are sampled in the air, thus agents that only learned to push can still reach

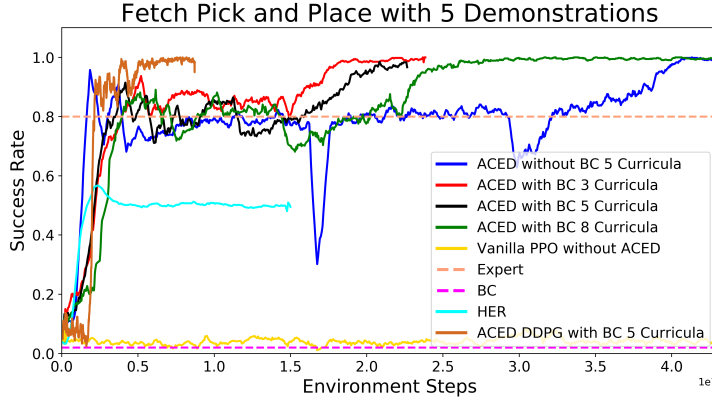


Figure 5: Learning curves of different algorithms in the pick-and-place environment with 5 demonstration trajectories. The horizontal axis represents the number of environment steps during training and the vertical axis represents the success rate. Expert and BC success rates are represented by dash lines because they didn’t have training processes and their success rates remain constant.

the goals close to the tabletop and receive a success rate of about 50%, but only agents that actually learned to pick and place will reach a success rate of 100%.

## B.2 Comparison with BC + RL without ACED

In order to further demonstrate the role of curriculum learning in ACED, this section compares the performance of ACED with an algorithm that only uses BC pre-trained policies to initialize the RL agent but doesn’t use curriculum learning during RL training. We refer to the RL algorithm that uses BC to pre-train the policy but doesn’t use ACED as “BC + RL”. The same PPO algorithm and BC pre-trained policy initializations as in the ACED experiments are used in all experiments presented in this section. We summarize the performance of both ACED with BC and BC + RL in Table 3 for convenient comparison, but the ACED with BC data in Table 3 are the same as the ones presented in Figure 2 and Table 1. As shown in Table 3, BC + RL only works better than ACED with BC when  $|\mathcal{T}| = 100$ , whereas its performance in terms of both convergence speed and success rate is worse than that of ACED with BC when  $|\mathcal{T}| = 50$  and  $|\mathcal{T}| = 20$ . When  $|\mathcal{T}| = 5$  or  $|\mathcal{T}| = 1$ , BC + RL is not able to learn pick-and-place and none of the runs converged to a success rate of 100%. Recorded videos show that when  $|\mathcal{T}| = 5$  and  $|\mathcal{T}| = 1$ , BC + RL can only learn to push the block to goal poses that are on the tabletop, but failed to learn pick-and-place when the goal pose is in the air. Since the goal pose has a 50% probability of being in the air in the pick-and-place environment, all the runs have converged to a success rate of around 50% during training.

We also evaluated BC + RL in the block stacking environment, but results show that none of the runs with  $|\mathcal{T}| = 100$  or  $|\mathcal{T}| = 20$  can converge to a success rate of 100%. In fact, the training curves remain zero throughout the entire training progress for all runs with BC + RL. The comparison between ACED with BC and BC + RL shows that ACED is especially helpful in scenarios where the target task is complicated or the number of demonstrations is small.

Table 3: Pick-and-Place Comparison with BC + RL

Algorithm <sup>1</sup>			$ \mathcal{T}  = 100$	$ \mathcal{T}  = 50$	$ \mathcal{T}  = 20$	$ \mathcal{T}  = 5^2$	$ \mathcal{T}  = 1^2$
ACED with BC	Convergence Env Steps (Million)	$C_{max} = 8$	2.78	3.40	8.40	24.00	41.21
		$C_{max} = 5$	2.32	3.55	5.97	16.50	38.85
		$C_{max} = 3$	4.15	6.53	7.88	17.67	25.56
		Average <sup>3</sup>	3.08	4.49	7.41	19.39	35.21
	Success Rate	$C_{max} = 8$	99%	100%	99%	97%	96%
		$C_{max} = 5$	96%	99%	99%	100%	95%
		$C_{max} = 3$	100%	99%	100%	100%	99%
		Average <sup>3</sup>	98.3%	99.3%	99.3%	99%	96.7%
BC + RL	Convergence Env Steps	2.47	14.66	13.58	(67.67)	(61.73)	
	Success Rate	100%	97%	99%	(37%)	(53%)	

<sup>1</sup> For each set of experiment except for BC + RL with  $|\mathcal{T}| = 5$  and  $|\mathcal{T}| = 1$ , we have 10 runs with different random seeds and the entries in the table are averaged from all runs. For BC + RL with  $|\mathcal{T}| = 5$  and  $|\mathcal{T}| = 1$ , we only conducted 3 runs each due to their long training time. For each run, we rollout 10 trajectories with the policy at convergence, and we compute the success rate by taking the average of all rollout trajectories for all runs.

<sup>2</sup> The entries for BC + RL with  $|\mathcal{T}| = 5$  and  $|\mathcal{T}| = 1$  are in brackets because none of these experiments have actually converged to a success rate of 100% during training. They instead converged to around 50% because they have only learned to push the block to the goal when the goal pose is on the tabletop, but they failed to learn how to pick up the block and lift them to the goal poses that are in the air.

<sup>3</sup> The average success rate for  $C_{max} = 8$ ,  $C_{max} = 5$  and  $C_{max} = 3$ .