A Environments

We describe below the three complex manipulation tasks that we selected from Dexterous Gym [19]. Other tasks in that environment are similar.

EggCatchUnderarm: The agent controls two dexterous manipulation hands in this environment. The goal is to throw the object with one hand from an initial in-hand position and catch the object at a desired pose with another hand. The observation space has 140 dimensions containing information of position, orientation and velocity about hands and the object. The continuous action space has 52 dimensions and the actions control the joints of the hands.

EggCatchOverarm: The main settings and the goal are the same as in EggCatchUnderarm. The only difference is that the the two hands are in the vertical plane instead of horizontal plane. This environment has the same state space (140 dimensions) and action space (52 dimensions) as EggCatchUnderarm.

PenSpin: This is an in-hand manipulation task with the goal of rotating a pen without dropping it. The observation space has 61 dimensions related to position, orientation and velocity of the pen and hand. The action space has 20 dimensions corresponding to the joints of the hand.

B Hyperparamters

We provide all the hyperparameters used in the different environments in Table 1 and Table 2.

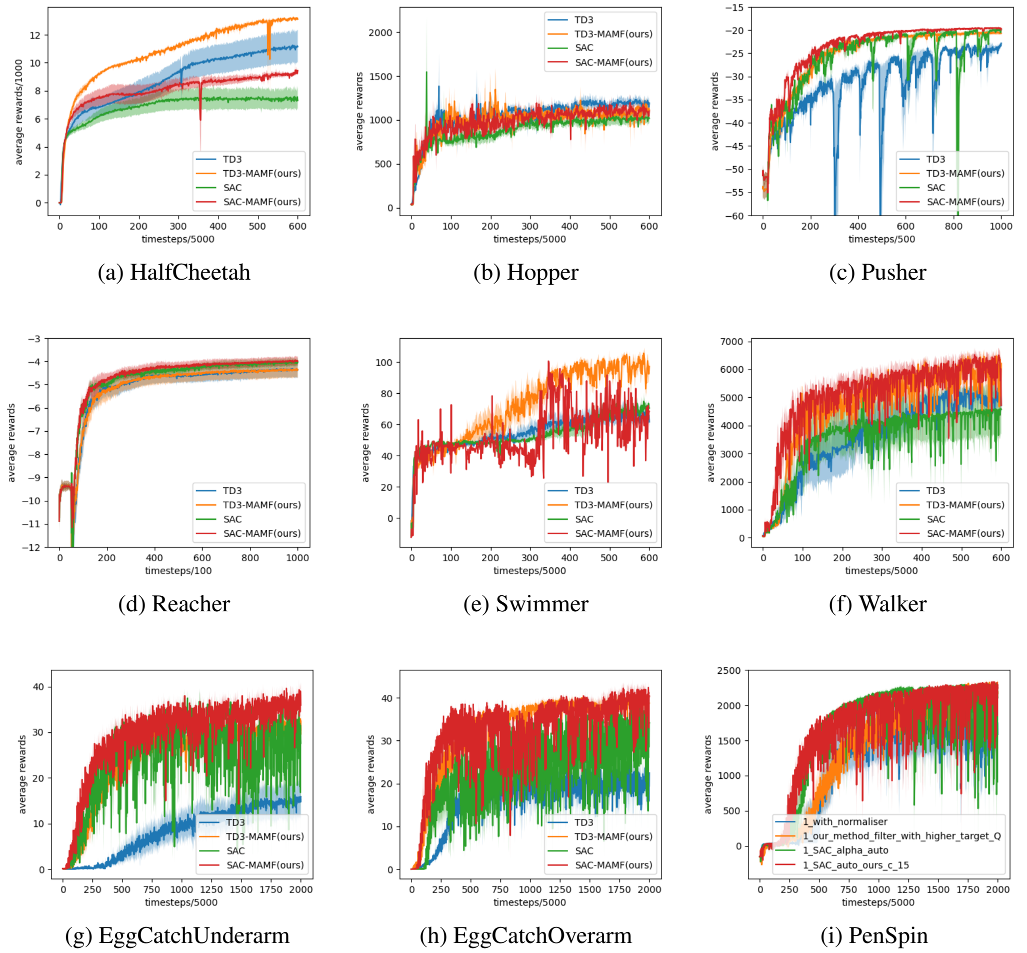
Table 1: Common hyperparameters in all the environments

Hyperparameter	Value
batch size N	256
discount γ	0.98
target network smoothing τ	0.005
frequency of delayed policy update	2
std of exploration noise	0.1
std of target policy noise	0.2
clipping bound of target policy noise in TD3	0.5
decay rate of clipping bound in our method ρ	0.9999996
number of layers in actors and critics	3
learning rate in actors and critics	3e-4
number of layers in dynamics model	4
learning rate in dynamics model	1e-4
number of nodes in each layer	256

Table 2: Different hyperparamters in different environments

Environment	Phase 1 training start h_0	Phase 1 duration h	c for the bias
Reacher	1000	5000	1/6
Pusher	5000	10000	1/6
Hopper	5000	25000	1/6
Walker	5000	25000	1/6
HalfCheetah	5000	25000	1/30
Swimmer	5000	25000	1/30
Dexterous gym	10000	25000	1/6

For the time step of starting to train the model and the actor-critic, a more complex environment requires more samples before we start training. For the constant c used for deciding the bias term, we initially want a heuristic way such that no more hyperparameter-tuning is needed for different environments. However, we find that the bias with $c = 1/6$ is too optimistic in HalfCheetah and Swimmer. For Hopper, Walker, HalfCheetah and Swimmer, each environment has a much larger max episode length $l = 1000$ than the others. Within these environments, the agent in HalfCheetah



(a) HalfCheetah

(b) Hopper

(c) Pusher

(d) Reacher

(e) Swimmer

(f) Walker

(g) EggCatchUnderarm

(h) EggCatchOverarm

(i) PenSpin

Figure 4: Comparison with SAC

and Swimmer will not fall down and thus has a high probability to get a large reward at some time step. So we decrease c in these two environments.

C Additional experiments

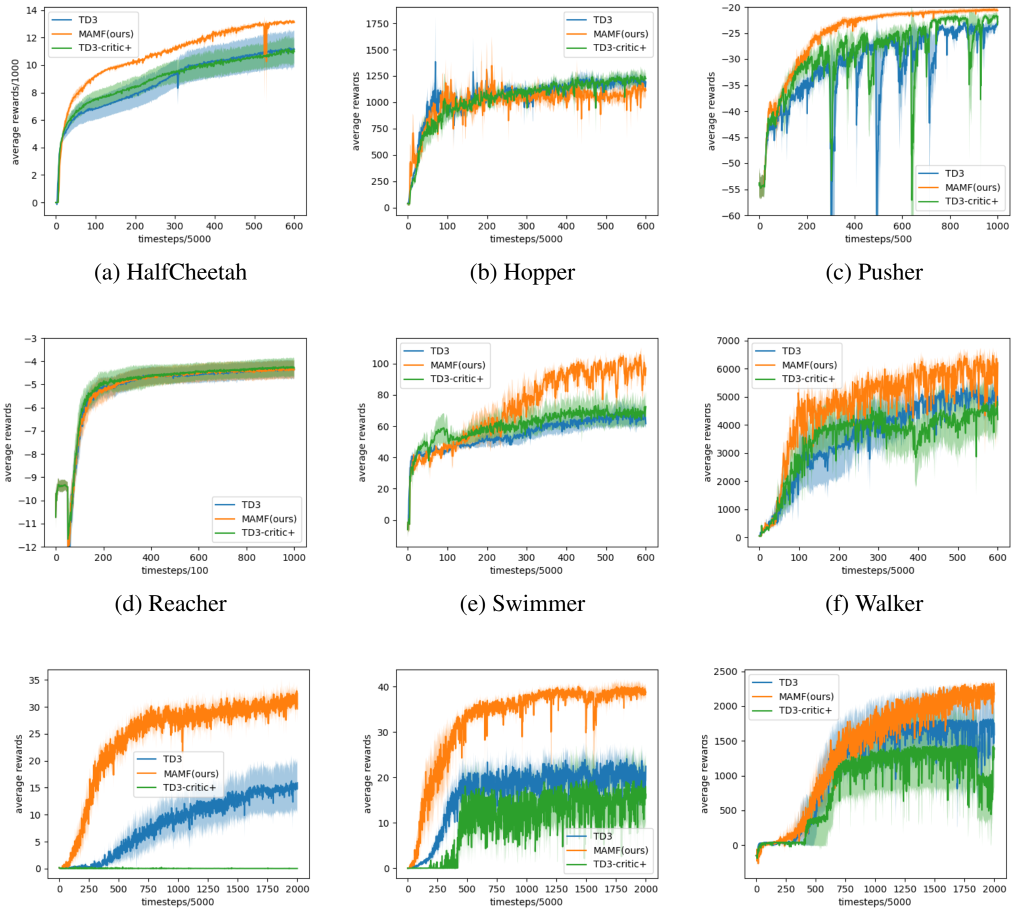
C.1 Comparison with SAC

We evaluate our proposition with SAC to further demonstrate its benefit. We run SAC and SAC augmented with our proposition (SAC-MAMF) in all the domains and the results are shown in Figure 4.

In the Mujoco tasks, SAC-MAMF generally improves over SAC, except in Swimmer where the performance of SAC-MAMF is worse. Interestingly, TD3-MAMF is generally better than SAC. In Dexterous gym, SAC works surprisingly well. In EggCatchUnderarm and EggCatchOverarm, SAC-MAMF generally improves over SAC. In PenSpin, the performances are similar. In all these tasks, the performances of SAC-MAMF have less variance.

C.2 Critic with a higher frequency of updates

For the experiments presented in the main paper, the critic is updated once at each time step. We compare our method with a variant of TD3 which has a higher frequency of updating the critic. The results of updating the critic twice at each time steps are shown in Figure 5. We can see that simply



(a) HalfCheetah

(b) Hopper

(c) Pusher

(d) Reacher

(e) Swimmer

(f) Walker

(g) EggCatchUnderarm

(h) EggCatchOverarm

(i) PenSpin

Figure 5: Results of increasing the frequency of updating the critic

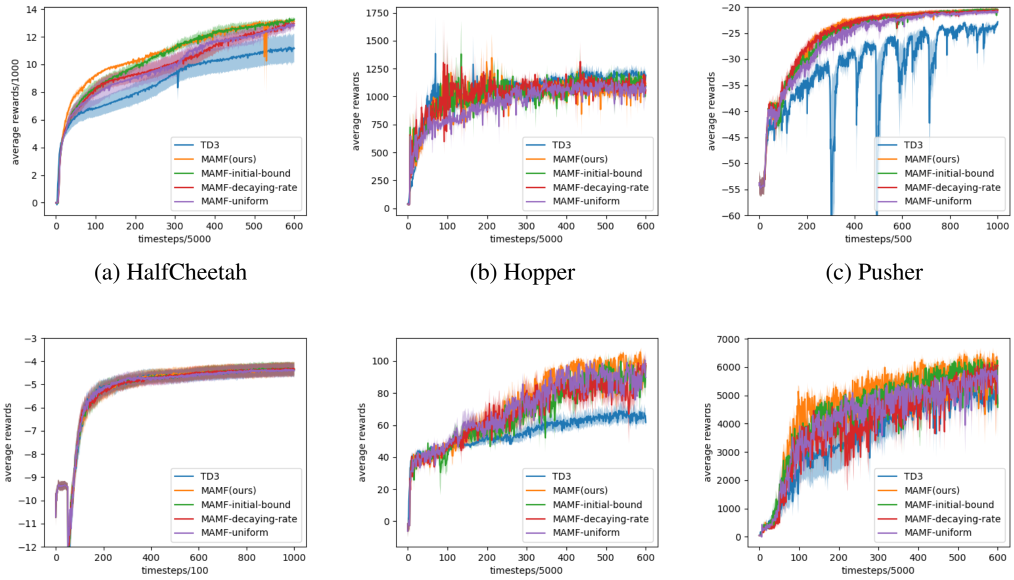
updating the critic more can not achieve a similar performance as our method. Including imaginary transitions as done in our method is more efficient than simply adding more samples from the replay buffer.

C.3 Decaying noise

We also run some experiments with different settings of the noise:

1. MAMF-initial-bound: use a smaller initial bound for the truncated Gaussian noise.
2. MAMF-decaying-rate: the bound decays faster.
3. MAMF-uniform: the noise comes from a uniform distribution over $(-a, a)$. a is initialized by the max action and exponentially decays with the same rate.

From the experiments, we can find that with different parameters of the noise, our method can still outperform the baseline TD3. However, using a smaller initial bound and decaying faster can slightly harm the performance. These two settings restrict the exploration in the model because the bound reaches 0 faster and the actions for generating the artificial transitions get closer to the policy action with fewer time steps.



(a) HalfCheetah

(b) Hopper

(c) Pusher

(d) Reacher

(e) Swimmer

(f) Walker

Figure 6: Results of using different noise setting