A Appendix

A.1 Experimental Details

All experiments were performed on an Alienware-Aurora-R9 with 8 Intel i7-9700 cores. We did not use a GPU. Code for the experiments is provided in the supplementary materials

A.1.1 Discrete Case

We perform 1000 episodes, using trajectories of 30 steps, a k of 8, and a γ of 0.825. Our learning rate begins at 0.01 and decays as $O(n^{-\frac{3}{4}})$.

A.1.2 For all continuous-space experiments

NNs: We use four-layer neural nets with RELU activations and 256 hidden units for both the actor and critic networks
Optimizer: Adam with default params
Learning rate: .001 for both actor and critic
Polyak averaging coefficient: .95
K (number of hindsight goals per non-hindsight goal): 8
Batch-size: 256
α_Q: 0.1
α_f: 0.5

N-torus with Freeze

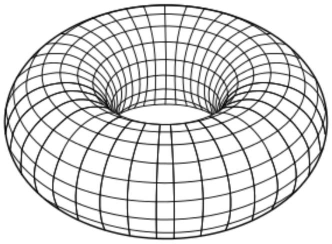

Figure 10: A Torus with a 2-dimensional surface

In this environment, robots move on a unit torus with a 4-dimensional surface 14. The robot can move up to 0.05 units in any direction on the surface. Alternatively, robots can take the "Freeze" action which randomly teleports them to a random location on the surface and permanently breaks the robot so it cannot move. The goal space is the 4-dimensional unit cube. The state space is the Cartesian product of the 4-dimensional unit cube with a boolean ($\mathbb{R}^4 \times [True, False]$). The boolean represents whether the robot is broken. Because this is a torus, the space loops around on itself as a torus would – robots that would move off the positive edge of any axis loop around and appear on the negative edge. The action is a 5-dimensional vector in the range [-1, 1]. The first four dimensions indicate the direction and distance to move, and the last axis indicates the probability with which to take the "freeze" action. Positive values are interpretted as a probability of taking the freeze action, and negative values are interpretted as "0 probability of taking the freeze action". We chose to make this value a scalar rather than a discrete value, because DDPG and SAC assume a continuous action space.

For the Torus with Freeze environment, we increase the ratio clipping factor c from 0.3 to 10, allowing the importance sampling weight to go as high as 11 and as low as $\frac{1}{11}$. We do this because the Freeze action was designed as a pathological counterexample to HER, and therefore the weights required to correct for its bias can be significantly higher than for more naturalistic environments.

Although this environment is very non-physical, it is the only benchmark we are aware of that assesses the bias of HER variants. For this reason, we felt reporting results with it was necessary.

γ : .98
Trajectory length: 50
Batches per epoch: 4
Episodes per epoch: 500

Entropy Regularization: 0.001
Car with Random Noise
γ : .95
Trajectory length: 20
Batches per epoch: 5
Episodes per epoch: 500
Entropy Regularization: 0.01
RedLight

The yellow tile in the figure represents the intersection that is dangerous during red lights.

γ : .9
Trajectory length: 50
Batches per epoch: 4
Episodes per epoch: 500
Entropy Regularization: 0.01
The light pattern was green: 1 second, yellow: 1 second, red: 4 seconds, with a randomized starting color. *FetchReach*
γ : .98
Trajectory length: 50
Batches per epoch: 40
Episodes per epoch: 50
Entropy Regularization: 0.001

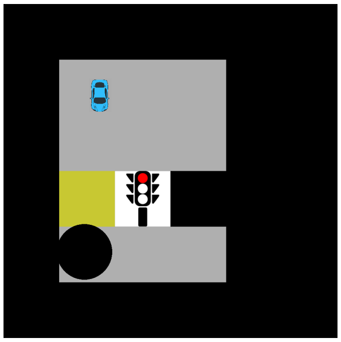

Figure 11: Red Light Environment

FetchPush
γ : .98
Trajectory length: 50
Batches per epoch: 40
Episodes per epoch: 50
Entropy Regularization: 0.01
FetchSlide
γ : .98
Trajectory length: 50
Batches per epoch: 40
Episodes per epoch: 50
Entropy Regularization: 0.001
Mobile Throwing Robot
γ : .9
Trajectory length: 20
Batches per epoch: 100
Episodes per epoch: 50
Entropy Regularization: 0.001
Mechanum robot – simulator
γ : .925
Trajectory length: 50
Batches per epoch: 100
Episodes per epoch: 50
Entropy Regularization: 0.01
Mechanum robot – analytic model
γ : .975
Trajectory length: 50
Batches per epoch: 10
Episodes per epoch: 50
Entropy Regularization: 0.01
Importance weight clipping value: 100

Our USHER and HER implementations are based on Tianhong Dai's implementation [16].

A.2 Additional Experiments

Due to space limitations, we were not able to include all of our experimental results. We have included these additional results here. We first trained the mechanum robot in simulation before transferring it to a real-world robot. Here are the training curves for the robot that was trained in simulation.

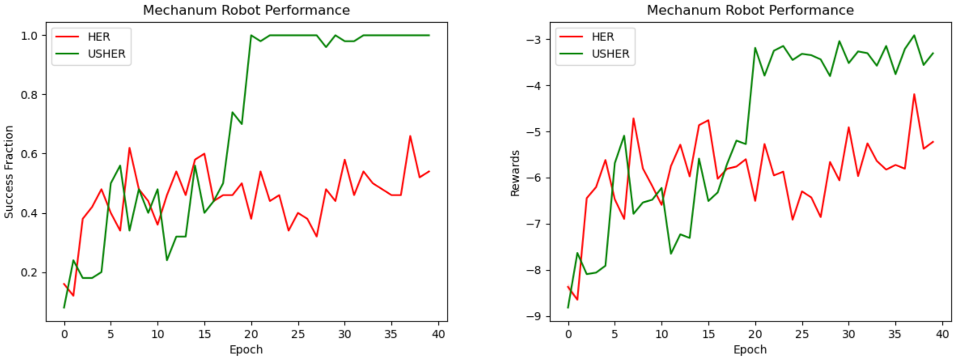

Figure 12: USHER performance with weight clipping (left) and without weight clipping (right)

A.3 Hyperparameter analysis

Here, we include an analysis of the performance of USHER as α_Q and α_f vary. We evaluated USHER's success rate on FetchReach after 30 episodes of training.

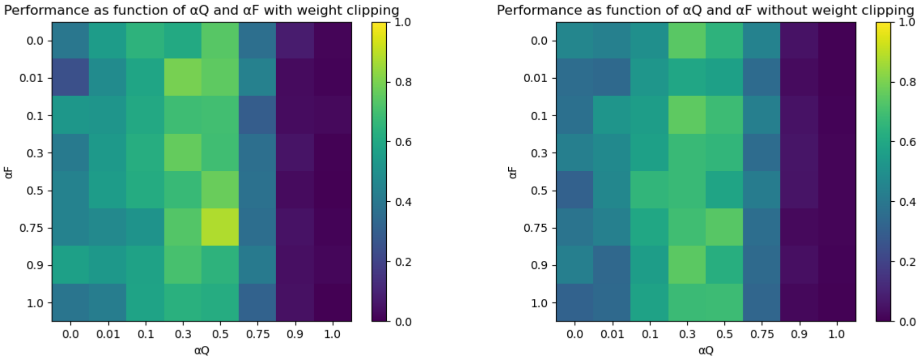

Figure 13: USHER performance with weight clipping (left) and without weight clipping (right)

Observe that the maximum value of W_{α_Q} is at most $\frac{1}{\alpha_Q}$. This means that for small values of α_Q, W_{α_Q} can potentially take on very large values, which can be a source of variance that harms USHER's performance. For large values of α_Q, hindsight goals have much less weight than random goals, which undercuts the source of HER's sample efficiency. For this reason, the best values of α_Q lie in the low-to-mid ranges, between $\alpha_Q = 0.1$ and $\alpha_Q = 0.5$. This seems to hold true for both clipped and unclipped weights. The selection of α_f mattered much less. We suspect this is because variance in the Q function matters more than variance in the future goal distribution, as we have to backpropagate through the Q value to get the policy gradient.

We did not carefully tune α_Q or α_f for our experiments. We simply guessed at a value of $\alpha_Q = 0.01$ and $\alpha_f = 0.5$ and used these values for all experiments. For c, we selected the value of 0.3 by trying a range of values from 0.1 to 10 on a sample environment similar to the stochastic car environment (6.2), but using simple displacement actions instead of dynamics that require numerical integration. We found that for $c > 1$ the bias induced by clipping was negligible.

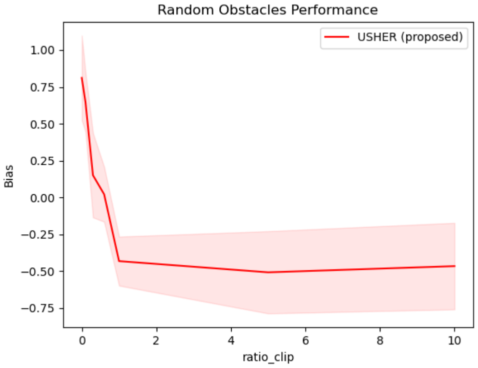

Figure 14: Bias as a function of the clipping parameter c

We found $c = 0.3$ worked well, so we used it for all experiments, except where the environment induced a very strong HER bias, in which case we set c high enough that clipping was effectively turned off.

A.4 Implementation

An implementation of USHER and our experiments can be found at: https://anonymous.4open.science/r/USHER_CoRL-0E16/README.md

A.5 Goal Selection Probability

Proposition 1:
Suppose g_π is fixed at the start of the trajectory, and g_r is sampled using HER. Then for any s', s, a, g_r, g_π, T,

$$f(s' \mid s, a, g_r, g_\pi, T) = \frac{f(g_r \mid s', \pi(s', g_\pi), g_\pi, T-1)}{f(g_r \mid s, a, g_\pi, T)} f(s' \mid s, a)$$

Proof. Suppose g_π is sampled before the trajectory begins, and is not changed at training time. Let $s, a,$ and s' be random variables representing a state, action, and subsequent state. Let $Q^\pi_{HER}(s, a, g_r, g_\pi)$ be the solution to the Bellman equation obtained using HER's sampling bias, with state s, hindsight goal g_r, policy goal g_π, and deterministic policy $\pi(s', g_\pi)$. Let t_s be the number of steps remaining in the trajectory when state s is sampled, t'_s be the number of steps remaining in the trajectory when state s' is sampled, and T be an integer

$$
\begin{aligned}
f(s' \mid s, a, g_r, g_\pi, t_s = T) &= \frac{f(s', s, a, g_r, g_\pi, t_s = T)}{f(s, a, g_r, g_\pi, t_s = T)} \\
&= \frac{f(g_r \mid s', s, a, g_\pi, t_s = T) f(s', s, a, g_\pi, t_s = T)}{f(s, a, g_r, g_\pi, t_s = T)} \\
&= \frac{f(g_r \mid s', s, a, g_\pi, t_s = T) f(s' \mid s, a, g_\pi, t_s = T)}{f(g_r \mid s, a, g_\pi, t_s = T)} \frac{f(s, a, g_\pi, t_s = T)}{f(s, a, g_\pi, t_s = T)} \\
&= f(s' \mid s, a, g_\pi, t_s = T) \frac{f(g_r \mid s', s, a, g_\pi, t_s = T)}{f(g_r \mid s, a, g_\pi, t_s = T)}
\end{aligned}
$$

Observe that with HER, g_r is either selected from the trajectory beginning with s', sampled independently of s' or left the same as the original goal given to the policy. In all three cases, $f(g_r \mid s', s, a, g_\pi, t_s = T) = f(g_r \mid s', g_\pi, t_s = T)$. In the first case where g_r comes from the future trajectory, the Markov property implies that given the most recent observed state s', g_r is independent of all earlier states and actions, including s and a, so $f(g_r \mid s', s, a, g_\pi, t_s = T) = f(g_r \mid s', g_\pi, t_s = T)$. In the second case where g_r is sampled independently of the trajectory, so $f(g_r \mid s', s, a, g_\pi, t_s = T) = f(g_r) = f(g_r \mid s', g_\pi, t_s = T)$. In the third case, $g_r = g_\pi$, so g_r has no dependence on s, a, or s'. In any case, $f(g_r \mid s', s, a, g_\pi, t_s = T) = f(g_r \mid s', g_\pi, t_s = T)$.

For the same reason that g_r depends only upon s' and not on s when s' is known, g_r depends only on $t_{s'}$ and not t_s when $t_{s'}$ is known. Thus we find that $f(g_r \mid s', g_\pi, t_s = T) = f(g_r \mid s', g_\pi, t_{s'} = T - 1)$.

$$
\begin{aligned}
f(s' \mid s, a, g_\pi, g_r, t_s = T) &= f(s' \mid s, a, g_\pi, t_s = T) \frac{f(g_r \mid s', g_\pi, t_{s'} = T - 1)}{f(g_r \mid s, a, g_\pi, t_s = T)} \\
&= f(s' \mid s, a, g_\pi, t_s = T) \frac{E_{a'}[f(g_r \mid s', a', g_\pi, t_{s'} = T - 1) \mid s' g_\pi]}{f(g_r \mid s, a, g_\pi, t_s = T)} \\
&= f(s' \mid s, a, g_\pi, t_s = T) \frac{f(g_r \mid s', \pi(s', g_\pi), g_\pi, t_{s'} = T - 1)}{f(g_r \mid s, a, g_\pi, t_s = T)}
\end{aligned}
$$

Observe that from the Markov assumption of the environment, the transition probability depends only on s, a, and does not depend on g_π nor t_s. g_π is sampled before the trajectory begins, independently of all other random variables. From this we can see that $f(s' \mid s, a)$ is independent of g_π and t_s.

We can then conclude that for all g_r, g_π

$$
f(s' \mid s, a, g_\pi, g_r, t_s = T) = f(s' \mid s, a) \frac{f(g_r \mid s', \pi(s', g_\pi), g_\pi, t_{s'} = T - 1)}{f(g_r \mid s, a, g_\pi, t_s = T)}
$$

For conciseness, we will abbreviate this to

$$
f(s' \mid s, a, g_\pi, g_r, T) = f(s' \mid s, a) \frac{f(g_r \mid s', \pi(s', g_\pi), g_\pi, T - 1)}{f(g_r \mid s, a, g_\pi, T)}
$$

□

A.6 2-goal HER is asymptotically unbiased

Corollary. *Suppose $Q^\pi_{HER}(s, a, g_\pi, g_\pi)$ satisfies the Bellman equation and the distribution of future achieved goals is absolutely continuous with respect to the goal space. Then $Q^\pi_{HER}(s, a, g_\pi, g_\pi) = Q^*(s, a, g_\pi)$ for all s, a, g_π, where $Q^*(s, a, g_\pi)$ is the optimal goal-conditioned Q function.*

Proof. Suppose that the distribution of future achieved goals is absolutely continuous with respect to the goal space. Furthermore, suppose the goal space is a continuous space of at least one dimension. Then the probability of arriving exactly at any given goal g_r given the policy goal g_π is infinitesimal. This means that the only time there is a non-zero probability of having $g_r = g_\pi$ is when g_r is not drawn from the distribution of future achieved goals and HER instead uses the same goal as during the data-gathering phase.

Let $P(g_r = g_\pi \mid s, a, g_\pi)$ be the probability that g_π is selected as the reward goal. Then

$$\begin{aligned}P(g_r = g_\pi \mid s, a, g_\pi, T) &= P(g_r = g_\pi \mid s, a, g_\pi, T, H)P(H) \\ &+ P(g_r = g_\pi \mid s, a, g_\pi, T, \neg H)P(\neg H) \\ &= P(g_r = g_\pi \mid s, a, g_\pi, T, H)P(H) + 1P(\neg H)\end{aligned}$$

Since $P(g_r = g_\pi \mid s, a, g_\pi, H)$ is infinitesimal and $P(\neg H)$ is not, this reduces to

$$P(g_r = g_\pi \mid s, a, g_\pi, T) = P(\neg H) = \frac{1}{k+1}$$

Thus,

$$\begin{aligned}Q^\pi_{HER}(s, a, g_\pi, g_\pi) &= E_{s'}[\frac{P(g_r = g_\pi \mid s', \pi(s', g_\pi), g_\pi, T)}{P(g_r = g_\pi \mid s, a, g_\pi, T)} \\ &(R(s', g_\pi) + \gamma Q^\pi_{HER}(s', \pi(s', g_\pi), g_\pi, g_\pi)) \mid s, a, g_r, g_\pi, T] \\ &= E_{s'}[\frac{1/(k+1)}{1/(k+1)} \\ &(R(s', g_\pi) + \gamma Q^\pi_{HER}(s', \pi(s', g_\pi), g_\pi, g_\pi)) \mid s, a, g_r, g_\pi, T] \\ &= E_{s'}[(R(s', g_\pi) + \gamma Q^\pi_{HER}(s', \pi(s', g_\pi), g_\pi, g_\pi)) \mid s, a, g_r, g_\pi, T]\end{aligned}$$

Now, observe that $Q^\pi_{HER}(s, a, g_\pi, g_\pi)$ satisfies the one-goal Bellman equation. Since the Bellman equation has a unique solution, and $Q^*(s, a, g)$ is a solution, $Q^\pi_{HER}(s, a, g_\pi, g_\pi) = Q^*(s, a, g_\pi)$. □

A.7 Importance Sampling for Mixed Sampling Method

Proposition 2:
Let $W(s', s, a, g_r, g_\pi, T) = \frac{f(g_r|s,a,g_\pi,T)}{\alpha f(g_r|s,a,g_\pi,T)+(1-\alpha)f(g_r|s',\pi(s',g_\pi),g_\pi,T)}$. Let α be a real value in the range $(0, 1]$. Then for any s', s, a, g_r, g_π,

$$f(s' \mid s, a) = W(s', s, a, g_r, g_\pi, T)(\alpha f(s' \mid s, a) + (1-\alpha)f(s' \mid s, a, g_\pi, g_r, T))$$

Furthermore, for any function F of the state s',

$$\begin{aligned}\mathbb{E}_{s'}[F(s') \mid s, a] &= \alpha\mathbb{E}_{s'}[W(s', s, a, g_r, g_\pi, T)F(s') \mid s, a] \\ &+(1-\alpha)\mathbb{E}_{s'}[W(s', s, a, g_r, g_\pi, T)F(s') \mid s, a, g_\pi, g_r, T]\end{aligned}$$

Proof. Let g_r, g_π, T be a reward goal, a policy goal, and the remaining steps left in the current trajectory, respectively Using Proposition 1, we can show that

$$
\begin{aligned}
f(s' \mid s, a) =& f(s' \mid s, a) \frac{\alpha f(s' \mid s, a) + (1-\alpha) f(s' \mid s, a, g_\pi, g_r, T)}{\alpha f(s' \mid s, a) + (1-\alpha) f(s' \mid s, a, g_\pi, g_r, T)} \\
=& (\alpha f(s' \mid s, a) + (1-\alpha) f(s' \mid s, a, g_\pi, g_r, T)) \\
& \frac{f(s' \mid s, a)}{\alpha f(s' \mid s, a) + (1-\alpha) f(s' \mid s, a, g_\pi, g_r, T)} \\
=& (\alpha f(s' \mid s, a) + (1-\alpha) f(s' \mid s, a, g_\pi, g_r, T)) \\
& \frac{f(s' \mid s, a)}{\alpha f(s' \mid s, a) + (1-\alpha) \frac{f(g_r \mid s', \pi(s', g_\pi), g_\pi, T)}{f(g_r \mid s, a, g_\pi, T)} f(s' \mid s, a)} \\
=& (\alpha f(s' \mid s, a) + (1-\alpha) f(s' \mid s, a, g_\pi, g_r, T)) \\
& \frac{1}{\alpha + (1-\alpha) \frac{f(g_r \mid s', \pi(s', g_\pi), g_\pi, T)}{f(g_r \mid s, a, g_\pi, T)}} \\
=& (\alpha f(s' \mid s, a) + (1-\alpha) f(s' \mid s, a, g_\pi, g_r, T)) \\
& \frac{f(g_r \mid s, a, g_\pi, T)}{\alpha f(g_r \mid s, a, g_\pi, T) + (1-\alpha) f(g_r \mid s', \pi(s', g_\pi), g_\pi, T)} \\
=& (\alpha f(s' \mid s, a) + (1-\alpha) f(s' \mid s, a, g_\pi, g_r, T)) \\
& W(s', s, a, g_r, g_\pi, T)
\end{aligned}
$$

It then follows that for any g_r, g_π, T, the expectation value $\mathbb{E}_{s'}[F(s') \mid s, a]$ may be written as follows:

$$
\begin{aligned}
\mathbb{E}_{s'}[F(s') \mid s, a] =& \int_S f(s' \mid s, a) F(s') ds' \\
=& \int_S (\alpha f(s' \mid s, a) + (1-\alpha) f(s' \mid s, a, g_\pi, g_r, T)) \\
& W(s', s, a, g_r, g_\pi, T) F(s') ds' \\
=& \int_S (\alpha f(s' \mid s, a) + (1-\alpha) f(s' \mid s, a, g_\pi, g_r, T)) \\
& W(s', s, a, g_r, g_\pi, T) F(s') ds' \\
=& \alpha \int_S f(s' \mid s, a) W(s', s, a, g_r, g_\pi, T) F(s') ds' \\
& + (1-\alpha) \int_S f(s' \mid s, a, g_\pi, g_r, T) W(s', s, a, g_r, g_\pi, T) F(s') ds' \\
=& \alpha \mathbb{E}_{s'}[W(s', s, a, g_r, g_\pi, T) F(s') \mid s, a] \\
& + (1-\alpha) \mathbb{E}_{s'}[W(s', s, a, g_r, g_\pi, T) F(s') \mid s, a, g_r, g_\pi, T] \\
=& \alpha \mathbb{E}_{s'}[W(s', s, a, g_r, g_\pi, T) F(s') \mid s, a] \\
& + (1-\alpha) \mathbb{E}_{s'}[W(s', s, a, g_r, g_\pi, T) F(s') \mid s, a, g_r, g_\pi, T]
\end{aligned}
$$

□