Appendices

A Training Details and Hyperparameters

A.1 Continuous Control Experiments

Training details For reward learning experiments, we used the implementations of Preference Comparisons Algorithm from Imitation Learning Baseline Implementations [6] with a full list of hyperparameters in Table 1. For the RL component, we used soft actor-critic (SAC) [3] implementations from Stable-Baselines3 [5] in the locomotion control tasks with a list of hyperparameters in Table 2. For retraining evaluations, we use the same hyperparameters for SAC to train new agents against the frozen learned reward models.

Reward model The reward model consists of a single multi-layer perceptrons with two hidden layers of size 256 and LeakyReLU activations with slope 0.01. The input of the model consists of the state, action and next state vectors, and the input vector is normalized by running normalization. The output the the reward model is normalized by by exponential moving average. During relearning experiments, we directly use the raw reward output from the reward network while being normalzed by a VecNormalize layer in Stable-Baselines3 (`https://stable-baselines3.readthedocs.io/en/master/guide/vec_envs.html#vecenv`).

Hyperparameter	Value
Segment Length	50
Total Comparisons	2000
Number of Iteration	50
Reward Training Epochs	5
Query Schedule	constant

Table 1: Reward learning hyperparameters for continuous control experiments

Hyperparameter	Value
Learning Rate	0.0003
Batch Size	256
Discount	0.99
Learning Starts from	10000

Table 2: SAC hyperparameters for continuous control experiments

A.2 Tabular Experiments

Similarly to the continuous control experiments we use Imitation's implementation of preference comparison [6]. However, we use a tabular soft-q learning algorithm with a replay buffer [2] with reward relabling [4] to solve the environments. The reward network again uses a similar MLP architecture to the continuous control setting with a sightly smaller hidden size of 32. Finally, we normalize the reward functions before ensembling them using a simple running norm over sampled transitions which is frozen during retraining. Hyperparamaters can be found in Table 3.

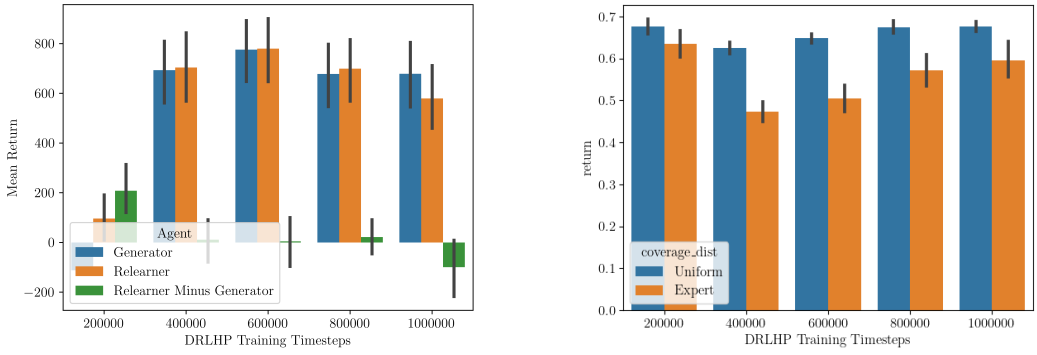
B Epic Distance as an Evaluation Metric

As an additional evaluation criterion, we considered using EPIC distance [1] to measure the distance between learned reward functions and the ground-truth reward. EPIC works by canonicalizing the rewards to be invariant to potential shaping, normalizing them to be invariant to scale, and then computing the L^2 norm of the difference of those functions over a *coverage* distribution of transitions. Here we consider two coverage distributions: uniform and expert distribution. The uniform distribution is uniform over feasible transitions. The expert distribution is the distribution of a soft-optimal policy with a temperature of 10 to give slightly more coverage.

Hyper Parameter	Value
Sampler Soft-Q Learning	
discount	0.99
learning rate	5e-2
replay buffer capacity	∞
temperature	0.1
samples from buffer per env sample	10
initial soft-q value	200
Reward Learning	
trajectory fragment length	30
total comparison budget	2500
rL budget	500000
frac. of comparisons from inital random trajs	0.1
select fragments for comparison	randomly
epochs of training per iteration	1
number of iterations	100
query schedule	constant
reward learning rate	1e-3
Reward Network	
reward network hidden layers	[32, 32]
activation function	ReLu
output normilization	Running Norm

Table 3: Tabular Experiment Hyperparamerers

C Additional Tabular Experiments



(a) (b)

Figure 1: Increasing the number of time steps of R.L. training does not seem to significantly effect relearning failures

To study the effects of training the sampler for a more time steps, we first consider a simple environment consisting of a 10x10 grid world. The agent begins in the lower left-hand corner of the environment and gains a ground-truth reward of 10 for reaching the lower right-hand cell, as seen in Figure 2.

The performance of the sampler and relearner initially increases with more training timesteps, with our relearners generalizing well and achieving slightly higher performance than their respective samplers. However, it quickly plateaus even though we *do not* see significant reductions in relearner performance with an increased number of time steps. The EPIC distances of our learned reward functions from the ground truth reward begin to increase after 400,000 timesteps Figure 1 (b).

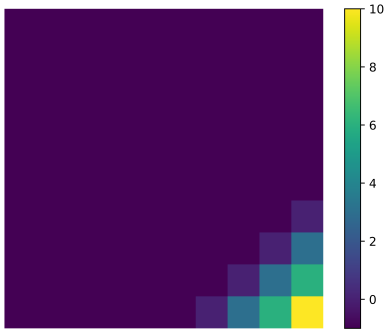


Figure 2: Tiny room environment
The ground-truth reward in the tiny room environment. Note that the reward only depends on the current state.

While increasing the number of total training timesteps used for DRLHP does seem to degrade the quality of the reward function according to EPIC distance. However, it does not appear to hurt relearning performance in the same way in this simple tabular environment.

References

[1] A. Gleave, M. D. Dennis, S. Legg, S. Russell, and J. Leike. Quantifying differences in reward functions. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=LwEQnp6CYev`.

[2] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine. Reinforcement learning with deep energy-based policies. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1352–1361. PMLR, 2017. URL `http://proceedings.mlr.press/v70/haarnoja17a.html`.

[3] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine. Soft actor-critic algorithms and applications. Technical report, 2018.

[4] K. Lee, L. M. Smith, and P. Abbeel. PEBBLE: feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 6152–6163. PMLR, 2021. URL `http://proceedings.mlr.press/v139/lee21i.html`.

[5] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22 (268):1–8, 2021. URL `http://jmlr.org/papers/v22/20-1364.html`.

[6] S. Wang, S. Toyer, A. Gleave, and S. Emmons. The `imitation` library for imitation learning and inverse reinforcement learning. `https://github.com/HumanCompatibleAI/imitation`, 2020.