We thank all reviewers for their time and great feedback. We'll incorporate various suggestions and clarifications in the revision. Here, we first address the shared points, then individual comments.

Hyperparameters (R1 and R3). The learning rate α for the baseline was chosen to be the best value from $[0.1, 0.2, 0.3, 0.4]$, while our model hyperparameters (the learning rate α_h for h, and the number of bins n_b for the return version of HCA) were selected informally to be $\alpha = 0.3, \alpha_b = 0.4$, $n_b = 3$ for the results in Fig. 4, and $n_b = 10$ elsewhere. Return HCA is sensitive to n_b, but all variants are quite robust to the choice of learning rate. We'll report all of this.

Learning h at scale (R2 and R3). We are working on this. The simplest architecture is a standard A3C agent with an extra layer that takes the embeddings x, x' (e.g. outputs of the conv layers) of two observations as an input, and outputs a distribution over actions $h(\cdot|x, x')$. It can be trained with the cross-entropy loss on all x, x', a samples on observed trajectories of complete episodes. (For very long episodes, one may need to subsample). The return version is similar, but simply takes the return value as the second input. Alternatively, one could use ideas from noise contrastive estimation, and parametrize the ratio h/π directly, similarly e.g. to the recent CPC algorithm. The positive examples would be actions on observed trajectories, and the negative examples – independent samples from the current π. To use h, like in regular A3C, one needs to remember the trajectory up to unroll length. In the state version, the return cannot be recursively composed anymore, and so the complexity of the update becomes quadratic in the length of the trajectory (return version remains linear). Indeed, h is policy dependent, but our intuition is that it can be meaningful / helpful even without being exactly correct for a particular policy. See e.g. the response to R1.

Reviewer 1 Why show standard deviation in the error bars? wouldn't the standard error be more informative?

We care about how robust the learning performance is, the std of 100 independent runs captures the deviation in performance run-to-run, while the sem would measure the confidence in the mean (but not how likely it is to get it).

I found Figure 3 (right) a little confusing.

Thanks for spotting this, you are entirely correct. We will clarify and fix this.

In figure 4 (left) does HCA-state use bootstrapping? If it is I don't understand why it is able to perform better.

Thanks for the great comment! All variants use the same n and bootstrap, so it really is about using HCA. We spent a considerable amount of time working out exactly what it is that makes the state version learn here, since (as you rightly point out), if everything was learned perfectly the intermediate ratios would be 1, and no learning would happen at all. The key is that we are learning h, π and V at the same time, but their learning dynamics are different. In particular h moves quicker than π (regardless of learning rate) as it is updated towards 1 for any observed sample. Now consider some interim $V(y) < 0$. That means the policy at the initial state x prefers the bad action a over good action b: $\pi(a|x) > \pi(b|x)$. But this also means that $h(a|x, y)$ has been observed more frequently, and because the cross entropy loss is more aggressive: $h(a|x, y) > \pi(a|x)$. Therefore the HCA return $= (h(a|x, y)/\pi(a|x) - 1)V(y) < 0$ and *discourages* the bad action. Similarly, $(h(b|x, y)/\pi(b|x) - 1)V(y) > 0$ and the good action is encouraged. We tested different learning rates, and initializations, and the effect persisted. We'll add this discussion to the paper.

I assume this is an actor-critic algorithm?

Indeed, all baselines implement n-step advantage actor critic, with $n = \infty$ for Monte Carlo.

Reviewer 2 How to properly define and parametrize a function that predicts P(a|y), if the future state y remembers a

This is a great point, and something we are thinking about. Note that the return version doesn't suffer from this. We could consider other forms of future conditioning that are richer than the return and remain informative when past actions affect the representation.

Reviewer 3 what is the connection to hindsight experience replay (besides the use of the word "hindsight").

It really is mostly the word :) The idea behind HER is to use a trajectory τ to train not only with the goal pursued by the policy that generated τ, but also other (randomly sampled) goals (counterfactual goals), whereas we are concerned with efficient credit assignment for the same goal (counterfactual actions).

Why is h_k a higher entropy distribution in general?

Good catch! That's a typo and should say *lower* (or equal) entropy. This is because we never add uncertainty by conditioning on an additional random variable, so the result is a sharper distribution.

Why is Figure 4(center) cut off at 200 episodes? Does MC PG overtake both HCA curves?

This is simply an oversight. All variants reach the same asymptotic performance. We'll make the number of episodes consistent in the final version.