We thank the reviewers for their feedback. We are excited to see that all four reviewers have rated our work positively and found that our paper addresses a very important problem (R1), is nicely motivated and builds logically on prior work (R2), works significantly better than baselines (R3), and is clearly contrasted with existing work (R4). We address the most salient points of feedback below, and will incorporate all of their feedback in the final version of our paper.

General Feedback

Novelty (R1, R2). "R2: *similar methods ... were published recently. It is very close to ... HER and the presented ideas for sampling new reward parameters are straightforward.*" Indeed, the power of our work is that we present simple methods to extend HER to general multi-task problems. Work by Eysenbach et al. 2020 was concurrent with ours, and is likely simultaneously in submission to NeurIPS 2020. Nevertheless, we contrast our work with HIPI as follows: our work explores different environments, has a different connection to inverse RL, and has extensive comparison of the pros and cons of each method. We're running HIPI as a baseline for all of our environments, which we will add when complete. However, Fig. 6 shows HIPI is worse on easy environments while taking more time to train; hence it is hard to see it working for more complex settings. Computationally, HIPI is $\approx 11\times$ slower than our method on environments that require bigger batch sizes. On Ant and Humanoid, this requires almost two weeks to run, which is limiting in academic labs such as ours. This analysis of computational complexity will be added to the paper.

Tasks (R3, R4). "R4: *the authors mostly use customized environments with dense rewards and goal-conditioned tasks where the task is represented by more than the goal state.*" For sparse goal-reaching, HER should work better, since the optimal relabeled task can be easily computed from the states reached. Our algorithms are designed for settings beyond pure goal reaching, where HER cannot be straightforwardly applied. We will clarify the categories of rewards that go into each environment, beyond the explicit reward formulae in Appendix B. And although our methods may not accelerate learning two highly dissimilar tasks, e.g. hammering a nail and opening a door, we show that they work well on a variety of meaningful and challenging robotics problems, such as control with safety or energy trade-offs.

Theory (R1). "R1: *There are no bounds that show how well both algorithms approximate the IRL problem and it's very unclear which one is more accurate.*" As with most prior work in hindsight relabeling [2, 46, 38, 45], we found it difficult to prove any meaningful bounds, likely due to the complexity of handling a distribution of policies $\{\pi(\cdot|z)|z \in \mathcal{Z}\}$ and the accuracy of the Q-function. Both are complicated by function approximation and complex dynamics and rewards. Intuitively, AIR may be more accurate because AIR directly maximizes the max-margin IRL objective when comparing against infinite random trajectories in the limit. We compared our relabeling algorithms empirically in Sec 4.4, and would love to see future work advancing our theoretical understanding of hindsight algorithms.

Algorithmic / Experimental details

Negative trajectories (R1). "R1: *the authors' algorithm would always pick the successful trajectories even though we know that informative negatives are crucial for off-policy RL algorithms.*" Indeed, hindsight relabeling in stochastic environments introduces hindsight bias, a tradeoff we discuss in Appendix C. In spite of this, our relabeling methods achieve good performance in the robotics environments that we test in. Extending our methods to more stochastic environments, e.g. by applying ARCHER [37], is an exciting avenue for future work.

Relabeling trajectories vs transitions (R1). "R1: *It is not clear why relabeling is done on the entire trajectory rather than on the individual transition.*" It's possible to relabel single transitions, based on (a) the transition reward or (b) the Q-value, but these both have problems. (a) Relabeling just based on the 1-step reward produces myopic relabeling that values short term rewards over large rewards that accumulate later. (b) In Sec 4.5, we present an in-depth discussion of the problems of using the Q-value to do transition relabeling. Relabeling the entire trajectory once is computationally efficient, simple, and already provides large performance gains. We will add this discussion earlier in the paper.

Distribution shift (R3). "R3: *The trajectories may be generated from a different policy parameter from that one being optimized.*" This is mostly fine for off-policy learning algorithms, such as SAC. We discuss bootstrap error in Appendix C, where we note that distribution shift from relabeled trajectories should at worst result in overestimation and encourage exploration of promising areas of the MDP. Empirically, our methods still perform well, and a promising extension of our work is to use offline RL methods to further reduce distribution shift error.

Hyperparameters for relabeling (R4). "R4: *There are some hyperparameters (e.g. number of tasks returned m) that need to be specified when deploying AIR.*" We simply select $m = 1$ relabeled trajectories out of $K = 100$ sampled task variables for all environments, and will add an ablation study on the effect of m and K in the camera-ready's appendix. With $m = 1$, our relabeling algorithms are empirically accurate in finding the proper relabeled task, as seen in Sec 4.4.

Error bars (R4). "R4: *I didn't notice anything specifying what this error bar signifies.*" The error bars show the standard deviation across seeds. When corrected by $1/\sqrt{n}$ for the number of runs, they have minimal overlap.

We again thank the reviewers for their detailed reviews, and even pointing out a few typos and missing citations, which we will add to the final version of the paper.