We thank the reviewers for their detailed comments. We hope this rebuttal addresses their concerns. First we clarify the problem we tackle of goal-reaching, its relevance, and properties. Then we provide a rigorous mathematical proof of the correctness of our Expert Relabeling technique. We also emphasize our other main contributions from an algorithm point of view. Finally, we include results on two considerably more complex environments, and further clarifications.

We tackle the problem of **learning a universal goal-reaching policy** [9] that, given **any goal** g, produces actions that lead to it. This can be specified by maximizing the indicator reward as defined in our Section 2: $r_t = 1[s_{t+1} == g]$.

- *[R1]"The idea of using sparse rewards depending on goal states"* is very extended in the literature [9, 12, 15]. As can be seen in these prior works, **many robotics problems** can be formulated as such. In particular, the FetchSlide task referred by *R1* was originally introduced in the HER paper [12], where such a sparse reward is used.
- *[R4] "the introduced goals, which are intermediate states, are different from the ground truth goals"*: in our problem statement, there are **not ground truth goals**. There is no "true reward" neither, and all our experiments only ever use the above-specified indicator reward. We are interested in learning to reach all goals equally well, and that is why our performance is evaluated in terms of the fraction of goals reached, as is common in this literature.

Here we provide a *[R5, 4] "**rigorously mathematical proof** of the statement"* that *"**guarantee the benefit** of augmenting data"* with **our Expert Relabeling** strategy, in the sense that it yields new (s, a, s', g) tuples that could have been produced by the expert. For a discrete state-action space, with deterministic dynamics, and assuming the demonstrations are optimal, the proof reduces to a shortest-path argument in graphs:

1. By the optimality of the demonstration $(s_0, a_0, s_1, a_1, s_2, \dots, g)$, there is no shorter path from s_0 to g.
2. By contradiction, there is no shorter path from s_0 to any encountered s_t neither, because if such path $P' = (s_0, s'_1, \dots s'_{t-1}, s_t)$ existed, then the path $(s_0, s'_1, \dots, s'_{t-1}, s_t, \dots, g)$ would be shorter than the demonstration.
3. By the same argument, there is no shorter path from s_t to s_{t+k} than the one that starts by going to s_{t+1}.
4. Therefore $(s_t, a_t, s_{t+1}, g' = s_{t+k})$ could also have been produced by the expert (the transition is optimal for g').

The argument can be extended to continuous stochastic case. We will include further details in the Appendix.

On top of our study of ER, we propose a **novel algorithm, goalGAIL**, that **combines and outperforms both HER and GAIL**. We also show that the algorithm is **robust to sub-optimal demonstrations** and that it can also **leverage state-only demonstrations**, which are very practical in robotics. Note that as long as the discriminator receives the state and next state (s, s', g) as input there is no concern that *[R2]"the notion of transition might be lost"* because this tuple captures the kind of transitions that the expert performs towards the goal g. As can be seen in Fig. 8 of our submitted Appendix, there are **no negative effects** on the studied tasks. BC + HER is not suited for these situations, and therefore the comparative performance of goalGAIL and BC + HER is of limited interest. We agree with R5 that our results should spark further research directions in the community about BC v.s. GAIL.

[R2] "Performing experiments with more complex domains": we added two more complex tasks: BlockPusher and Stack2. In BlockPusher a point-mass not only navigates itself, but also displaces a Block. In Stack2 a Fetch robot stacks two blocks on a desired spot, as done in [35]. These results bolster the conclusions of our paper. Furthermore *[R1] "scaling to real robot scenarios"* is not too far if we consider that 1M steps corresponds to 6h of real robot time [35].

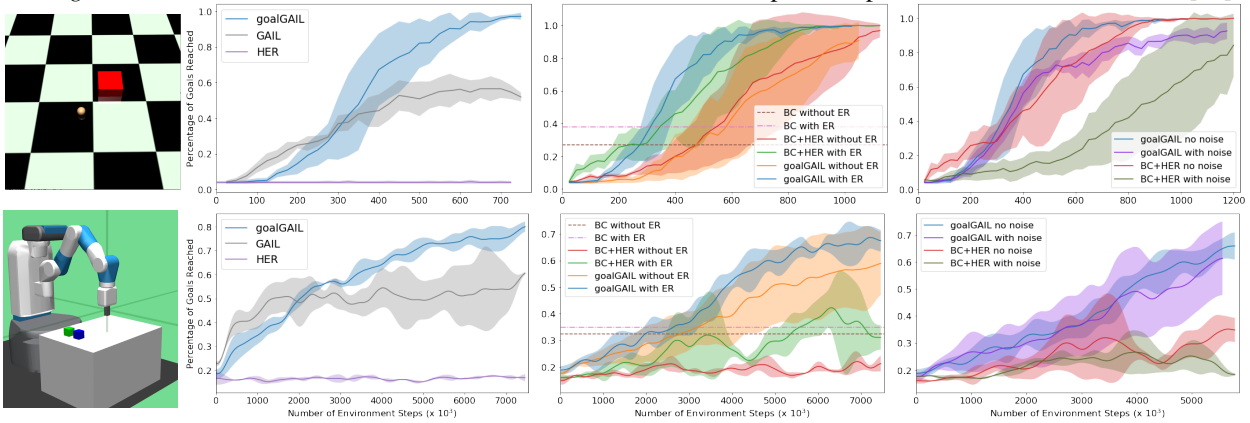

Figure 1: Experimental results on BlockPusher(row 1) and Stack2(row2). Column 2, 3, 4 correspond to the study in Fig. 3, 4 and 6 respectively in the submitted paper.

- *[R1] "What is the objective J"*: it's the expected cumulative reward. We will link this to line 20 of our algorithm.
- *[R1]* We will include a paragraph on GAIL in the background section to make the paper more self-contained.
- *[R1]* The work on "Task-parameterized movement learning" is very interesting, and we will explore this literature.
- *[R2]* **Quasi-static tasks** can be performed arbitrarily slow. This is the case for most robotics manipulation. If we also care about velocities, we can still use our framework by including velocities in the goal space.
- *[R2]* The system does **not need to start always from the same state**. This was the case only for the four-room experiment. In fetch robot experiments, the block positions are uniformly sampled at every rollout.