Goal-conditioned Imitation Learning

Yiming Ding [*1] **Carlos Florensa** [*1] **Mariano Phielipp** [2] **Pieter Abbeel** [1 3]

Abstract

Designing rewards for Reinforcement Learning (RL) is challenging because it needs to convey the desired task, be efficient to optimize, and be easy to compute. The latter is particularly problematic when applying RL to robotics, where detecting whether the desired configuration is reached might require considerable supervision and instrumentation. Furthermore, we are often interested in being able to reach a wide range of configurations, hence setting up a different reward every time might be unpractical. Methods like Hindsight Experience Replay (HER) have recently shown promise to learn policies able to reach many goals, without the need of a reward. Unfortunately, without tricks like resetting to points along the trajectory, HER might take a very long time to discover how to reach certain areas of the state-space. In this work we investigate different approaches to incorporate demonstrations to drastically speed up the convergence to a policy able to reach any goal, also surpassing the performance of an agent trained with other Imitation Learning algorithms. Furthermore, our method can be used when only trajectories without expert actions are available, which can leverage kinestetic or third person demonstration.

1. Introduction

Reinforcement Learning (RL) has shown impressive results in a plethora of simulated tasks, ranging from attaining super-human performance in video-games (Mnih et al., 2015; Vinyals et al., 2019) and board-games (Silver et al., 2017), to learning complex locomotion behaviors (Heess et al., 2017; Florensa et al., 2017a). Nevertheless, these successes are shyly echoed in real world robotics (Riedmiller et al., 2018; Zhu et al., 2018a). This is due to the difficulty of setting up the same learning environment that is enjoyed in simulation. One of the critical assumptions that are hard to obtain in the real world are the access to a reward function. Self-supervised methods have the power to overcome this limitation.

A very versatile and reusable form of self-supervision for robotics is to learn how to reach any previously observed state upon demand. This problem can be formulated as training a goal-conditioned policy (Kaelbling, 1993; Schaul et al., 2015) that seeks to obtain the indicator reward of having the observation exactly match the goal. Such a reward does not require any additional instrumentation of the environment beyond the sensors the robot already has. But in practice, this reward is never observed because in continuous spaces like the ones in robotics, the exact same observation is never observed twice. Luckily, if we are using an off-policy RL algorithm (Lillicrap et al., 2015; Haarnoja et al., 2018), we can "relabel" a collected trajectory by replacing its goal by a state actually visited during that trajectory, therefore observing the indicator reward as often as we wish. This method was introduced as Hindsight Experience Replay (Andrychowicz et al., 2017) or HER.

In theory these approaches could learn how to reach any goal, but the breadth-first nature of the algorithm makes that some areas of the space take a long time to be learned (Florensa et al., 2018b). This is specially challenging when there are bottlenecks between different areas of the state-space, and random motion might not traverse them easily (Florensa et al., 2017b). Some practical examples of this are pick-and-place, or navigating narrow corridors between rooms, as illustrated in Fig. 5 in appendix depicting the diverse set of environments we work with. In both cases a specific state needs to be reached (grasp the object, or enter the corridor) before a whole new area of the space is discovered (placing the object, or visiting the next room). This problem could be addressed by engineering a reward that guides the agent towards the bottlenecks, but this defeats the purpose of trying to learn without direct reward supervision. In this work we study how to leverage a few demonstrations that traverse those bottlenecks to boost the learning of goal-reaching policies.

Learning from Demonstrations, or Imitation Learning (IL),

[*]Equal contribution [1]Department of EECS, UC Berkeley, USA [2]Intel Corp. [3]Covariant. Correspondence to: Yiming Ding <dingyiming0427@berkeley.edu>, Carlos Florensa <florensa@berkeley.edu>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

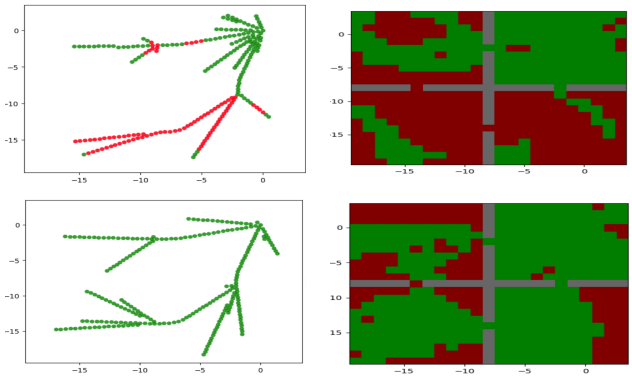
is a well-studied field in robotics (Kalakrishnan et al., 2009; Ross et al., 2011; Bojarski et al., 2016). In many cases it is easier to obtain a few demonstrations from an expert than to provide a good reward that describes the task. Most of the previous work on IL is centered around trajectory following, or doing a single task. Furthermore it is limited by the performance of the demonstrations, or relies on engineered rewards to improve upon them. In this work we study how IL methods can be extended to the goal-conditioned setting, and show that combined with techniques like HER it can outperform the demonstrator without the need of any additional reward. We also investigate how the different methods degrade when the trajectories of the expert become less optimal, or less abundant. Finally, the method we develop is able to leverage demonstrations that do not include the expert actions. This is very convenient in practical robotics where demonstrations might have been given by a motion planner, by kinestetic demonstrations (moving the agent externally, and not by actually actuating it), or even by another agent. To our knowledge, this is the first framework that can boost goal-conditioned policy learning with only state demonstrations.

2. Preliminaries

We define a discrete-time finite-horizon discounted Markov decision process (MDP) by a tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \rho_0, \gamma, H)$, where $\mathcal{S}$ is a state set, $\mathcal{A}$ is an action set, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}_+$ is a transition probability distribution, $\gamma \in [0, 1]$ is a discount factor, and H is the horizon. Our objective is to find a stochastic policy π_θ that maximizes the expected discounted reward within the MDP, $\eta(\pi_\theta) = \mathbb{E}_\tau[\sum_{t=0}^{T} \gamma^t r(s_t, a_t, s_{t+1})]$. We denote by $\tau = (s_0, a_0, ...,)$ the entire state-action trajectory, where $s_0 \sim \rho_0(s_0)$, $a_t \sim \pi_\theta(a_t|s_t)$, and $s_{t+1} \sim \mathcal{P}(s_{t+1}|s_t, a_t)$. In the goal-conditioned setting that we use here, the policy and the reward are also conditioned on a "goal" $g \in \mathcal{S}$. The reward is $r(s_t, a_t, s_{t+1}, g) = \mathbb{1}\big[s_{t+1} == g\big]$, and hence the return is the γ^h, where h is the number of time-steps to the goal. Given that the transition probability is not affected by the goal, g can be "relabeled" in hindsight, so a transition $(s_t, a_t, s_{t+1}, g, r = 0)$ can be treated as $(s_t, a_t, s_{t+1}, g' = s_{t+1}, r = 1)$. Finally, we also assume access to D trajectories $\big\{(s_0^j, a_0^j, s_1^j, ...)\big\}_{j=0}^{D}$ that were collected by an expert attempting to reach a goal g_j sampled uniformly among the feasible goals. Those trajectories must be approximately geodesics, meaning that the actions are taken such that the goal is reached as fast as possible.

3. Demonstrations in Goal-conditioned tasks

In this section we describe the different algorithms we compare to pure Hindsight Experience Replay (Andrychowicz et al., 2017). See the Appendix to prior work on adding a Behavioral Cloning loss to the policy update as in (Nair et al., 2018). Here we propose a novel expert relabeling technique, we formulate for the first time a goal-conditioned GAIL algorithm, and propose a method to train it with state-only demonstrations.



(a) Performance on reaching states visited in demonstrations. The state is colored in green if the policy reaches it when attempting so, and red otherwise.

(b) Performance on reaching feasible states. Each cell is colored green if the policy can reach the center of it when attempting so, and red otherwise.

Figure 1. Policy performance on reaching different goals in the four rooms, when training on 20 demonstrations with standard Behavioral Cloning (top row) or with our expert relabeling (bottom).

3.1. Relabeling the expert

The expert trajectories are collected by asking the expert to reach a specific goal g^j. But they are also valid trajectories to reach any other state visited within the demonstration! This is the key motivating insight to propose a new type of relabeling: if we have the transitions $(s_t^j, a_t^j, s_{t+1}^j, g^j)$ in a demonstration, we can also consider the transition $(s_t^j, a_t^j, s_{t+1}^j, g' = s_{t+k}^j)$ as also coming from the expert! This can be understood as a type of data augmentation leveraging the assumption that the tasks we work on are quasi-static. It will be particularly effective when not many demonstrations are available. In Fig. 1 we compare the final performance of two agents for Four Rooms environment, one trained with pure Behavioral Cloning, and the other one also using expert relabeling.

3.2. Goal-conditioned GAIL with Hindsight

The compounding error in Behavioral Cloning might make the policy deviate arbitrarily from the demonstrations, and it requires too many demonstrations when the state dimension increases. The first problem is less severe in our goal-conditioned case because in fact we do want to visit and be able to purposefully reach all states, even the ones that the expert did not visited. But the second drawback will become pressing when attempting to scale this method to practical robotics tasks where the observations might be

high-dimensional sensory input like images. Both problems can be mitigated by using other Imitation Learning algorithms that can leverage additional rollouts collected by the learning agent in a self-supervised manner, like GAIL (Ho & Ermon, 2016). In this section we extend the formulation of GAIL to tackle goal-conditioned tasks, and then we detail how it can be combined with HER (Andrychowicz et al., 2017), which allows to outperform the demonstrator and generalize to all goals. We call this algorithm *goal-GAIL*.

First of all, the discriminator needs to also be conditioned on the goal $D_\psi(a, s, g)$. Once the discriminator is fitted, we can run our favorite RL algorithm on the reward $\log D_\psi(a^h_t, s^h_t, g^h)$. In our case we used the off-policy algorithm DDPG (Lillicrap et al., 2015) to allow for the relabeling techniques outlined above. In the goal-conditioned case we also supplement with the indicator reward $r^h_t = \mathbb{1}\left[s^h_{t+1} == g^h\right]$. This combination is slightly tricky because now the fitted Q_ϕ does not have the same clear interpretation it has when only one of the two rewards is used (Florensa et al., 2018a) . Nevertheless, both rewards are pushing the policy towards the goals, so it shouldn't be too conflicting. Furthermore, to avoid any drop in final performance, the weight of the reward coming from GAIL (δ_{GAIL}) can be annealed. See Appendix for details.

3.3. Use of state-only Demonstrations

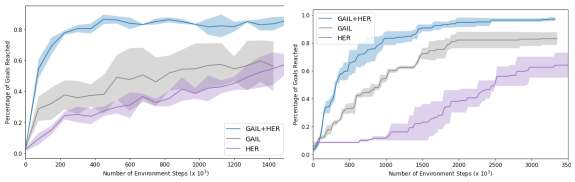
Both Behavioral Cloning and GAIL use state-action pairs from the expert. This limits the use of the methods, combined or not with HER, to setups where the exact same agent was actuated to reach different goals. Nevertheless, much more data could be cheaply available if the action was not required. For example, kinestetic demonstration or third-person imitation (Stadie et al., 2017). The main insight we have here is that we can replace the action in the GAIL formulation by the next state s', and in most environments this should be as informative as having access to the action directly. Intuitively, given a desired goal g, it should be possible to determine if a transition $s \rightarrow s'$ is taking the agent in the right direction. The loss function to train a discriminator able to tell apart the current agent and demonstrations (always transitioning towards the goal) is simply:

$$\mathcal{L}_{GAIL^s}(D^s_\psi, \mathcal{D}, \mathcal{R}) = \mathbb{E}_{(s,s',g)\sim\mathcal{R}}[\log D^s_\psi(s, s', g)] + \mathbb{E}_{(s,s',g)\sim\mathcal{D}}[\log(1 - D^s_\psi(s, s', g))].$$

4. Experiments

We are interested in answering the following questions:

1. Can the use of demonstrations accelerate the learning of goal-conditioned tasks without reward?
2. Is the *Expert Relabeling* an efficient way of doing data-augmentation on the demonstrations?



(a) Continuous Four rooms (b) Fetch Pick & Place

Figure 2. Performance of Goal-conditioned GAIL compared to only GAIL and HER

3. Can state-only demonstrations be leveraged equally well as full trajectories?
4. Compared to Behavorial Cloning methods, is GAIL more robust to noise in the expert actions?

We evaluate these questions in two different simulated robotic goal-conditioned tasks that are detailed in the next subsection. All the results use 20 demonstrations. All curves have 5 random seeds and the shaded area is one standard deviation

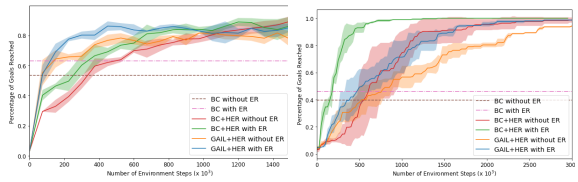
4.1. Tasks

Experiments are conducted in two continuous environments in MuJoCo (Todorov et al., 2012). The performance metric we use in all our experiments is the percentage of goals in the feasible goal space the agent is able to reach.

Four rooms environment: A point mass is placed in an environment with four rooms connected through small openings. The action space is continuous and specifies the desired change in state space which corresponds to the goal space. **Pick and Place**: A fetch robot needs to pick a block and place it in a desired point in space as described in Nair et al. (2018). The control is four-dimensional, corresponding to a change in position of the end-effector and a change in gripper opening. The goal space is the position of the block.

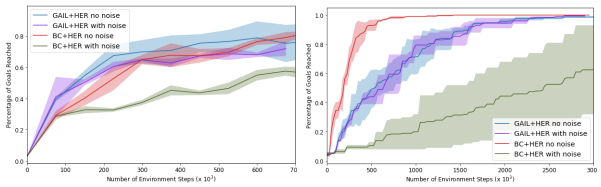
4.2. Goal-conditioned Imitation Learning

In goal-conditioned tasks, HER (Andrychowicz et al., 2017) should eventually converge to a policy able to reach any desired goal. Nevertheless, this might take a long time, specially in environments where there are bottlenecks that need to be traversed before accessing a whole new area of the goal space. In this section we show how the methods introduced in the previous section can leverage a few demonstrations to improve the convergence speed of HER. This was already studied for the case of Behavioral Cloning by (Nair et al., 2018), and in this work we show we also get a benefit when using GAIL as the Imitation Learning algorithm. In both environments, we observe that running GAIL with relabeling (GAIL+HER) considerably outperforms running each of them in isolation. HER alone has a very slow convergence,



(a) Continuous Four rooms (b) Fetch Pick & Place

Figure 3. Effect of our Expert Relabeling technique on different Goal-Conditioned Imitation Learning algorithms.



(a) Continuous Four rooms (b) Fetch Pick & Place

Figure 4. Learning with sub-optimal demonstrations

although as expected it ends up reaching the same final performance if run long enough. On the other hand GAIL by itself learns fast at the beginning, but its final performance is capped. This is because despite collecting more samples on the environment, those come with no reward of any kind indicating what is the task to perform (reach the given goals). Therefore, once it has extracted all the information it can from the demonstrations it cannot keep learning and generalize to goals further from the demonstrations. This is not an issue anymore when combined with HER, as our results show.

4.3. Expert relabeling

Here we show that the Expert Relabeling technique introduced in Section 3.1 is beneficial in the goal-conditioned imitation learning framework. As shown in Fig. 3, our expert relabeling technique brings considerable performance boosts for both Behavioral Cloning methods and goal-GAIL in both environments. We also perform a further analysis of expert relabeling in the four-rooms environment. We see in Fig. 1 that without the expert relabeling, the agent fails to learn how to reach many intermediate states visited in the middle of a demonstration.

4.4. Using state-only demonstrations

Behavioral Cloning and standard GAIL rely on the state-action (s, a) tuples from the expert. Nevertheless there are many cases in robotics where we only have access to observation-only demonstrations. In this section we want to emphasize that all the results obtained with our goal-GAIL method and reported in Fig. 2 and Fig. 3 do *not* require actions that the expert took. Surprisingly, in the four rooms environment, despite the more restricted information goal-GAIL has access to, it outperforms BC combined with HER. This might be due to the superior imitation learning performance of GAIL, and also to the fact that these tasks might be possible to solve by only matching the state-distribution of the expert. With GAIL conditioned only on current state but not action (as also done in other non-goal-conditioned works (Fu et al., 2018)), we observe that the discriminator learns a very well shaped reward that encourages the agent to go towards the goal, as pictured in Fig. 6 in appendix. See the Appendix for more details.

4.5. Robustness to sub-optimal expert

In the above sections we assumed access to optimal experts. Nevertheless, in practical applications the experts might have a more erratic behavior. In this section we study how the different methods perform with a sub-optimal expert. To do so we collect trajectories attempting goals g by modifying our optimal expert $\pi^*(a|s, g)$ in two ways: We add noise α to the optimal actions and make it be ϵ-greedy. The sub-optimal expert is then $a = \mathbb{1}[r < \epsilon]u + \mathbb{1}[r > \epsilon](\pi^*(a|s, g) + \alpha)$, where $r \sim \texttt{Unif}(0, 1)$, $\alpha \sim \mathcal{N}(0, \sigma_\alpha^2 I)$ and u is a uniformly sampled random action.

In Fig. 4 we observe that approaches that copy the action of the expert, like Behavioral Cloning, greatly suffer under a sub-optimal expert. On the other hand, discriminator-based methods are able to leverage noisier experts. A possible explanation is that a discriminator approach can give a positive signal as long as the transition is "in the right direction", without trying to exactly enforce a single action. Under this lens, having some noise in the expert might actually improve the performance of these adversarial approaches, as it has been observed in many generative models literature (Goodfellow et al.).

5. Conclusions and Future Work

Hindsight relabeling can be used to learn useful behaviors without any reward supervision for goal-conditioned tasks, but they are inefficient when the state-space is large or includes exploration bottlenecks. In this work we show how only a few demonstrations can be leveraged to improve the convergence speed of these methods. We introduce a novel algorithm, goal-GAIL, that converges faster than HER and to a better final performance than a naive goal-conditioned GAIL. We also study the effect of doing expert relabeling as a type of data augmentation on the provided demonstrations, and demonstrate it improves the performance of our goal-GAIL as well as goal-conditioned Behavioral Cloning. We emphasize that our goal-GAIL method only needs state demonstrations, without using expert actions like other Behavioral Cloning methods. Finally, we show that goal-GAIL is robust to sub-optimalities in the expert behavior.

References

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W. Hindsight Experience Replay. *Advances in Neural Information Processing Systems*, 2017. ISSN 10495258. doi: 10.1016/j.surfcoat.2018.06.018. URL http://arxiv.org/abs/1707.01495.

Blondé, L. and Kalousis, A. Sample-Efficient Imitation Learning via Generative Adversarial Nets. *AISTATS*, 2019. URL https://youtu.be/-nCsqUJnRKU.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K. End to End Learning for Self-Driving Cars. 2016. URL http://arxiv.org/abs/1604.07316.

Finn, C., Levine, S., and Abbeel, P. Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization. *Internation Conference in Machine Learning*, 3 2016. URL http://arxiv.org/abs/1603.00448.

Florensa, C., Duan, Y., and Abbeel, P. Stochastic Neural Networks for Hierarchical Reinforcement Learning. *International Conference in Learning Representations*, pp. 1–17, 2017a. ISSN 14779129. doi: 10.1002/rcm.765. URL http://arxiv.org/abs/1704.03012.

Florensa, C., Held, D., Wulfmeier, M., Zhang, M., and Abbeel, P. Reverse Curriculum Generation for Reinforcement Learning. *Conference on Robot Learning*, pp. 1–16, 2017b. ISSN 1938-7228. doi: 10.1080/00908319208908727. URL http://arxiv.org/abs/1707.05300.

Florensa, C., Degrave, J., Heess, N., Springenberg, J. T., and Riedmiller, M. Self-supervised Learning of Image Embedding for Continuous Control. In *Workshop on Inference to Control at NeurIPS*, 2018a. URL http://arxiv.org/abs/1901.00943.

Florensa, C., Held, D., Geng, X., and Abbeel, P. Automatic Goal Generation for Reinforcement Learning Agents. *International Conference in Machine Learning*, 2018b. URL http://arxiv.org/abs/1705.06366.

Fu, J., Luo, K., and Levine, S. Learning Robust Rewards with Adversarial Inverse Reinforcement Learning. *International Conference in Learning Representations*, 10 2018. URL http://arxiv.org/abs/1710.11248.

Gao, Y., Harry, H., Ji, X., Fisher, L., Sergey, Y., and Trevor, L. Reinforcement Learning from Imperfect Demonstrations. *Internation Conference in Machine Learning*, 2018.

Goodfellow, I. J., Pouget-abadie, J., Mirza, M., Xu, B., and Warde-farley, D. Generative Adversarial Nets. pp. 1–9.

Haarnoja, T., Zhou, A., Abbeel, P., Levine, S., and Sciences, C. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *Internation Conference in Machine Learning*, pp. 1–15, 2018.

Heess, N., TB, D., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, S. M. A., Riedmiller, M., and Silver, D. Emergence of Locomotion Behaviours in Rich Environments. 7 2017. URL http://arxiv.org/abs/1707.02286.

Ho, J. and Ermon, S. Generative Adversarial Imitation Learning. *Advances in Neural Information Processing Systems*, 2016. URL http://arxiv.org/abs/1606.03476.

Kaelbling, L. P. Learning to Achieve Goals. *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1094–1098, 1993.

Kalakrishnan, M., Buchli, J., Pastor, P., and Schaal, S. Learning locomotion over rough terrain using terrain templates. In *International Conference on Intelligent Robots and Systems*, pp. 167–172. IEEE, 2009. ISBN 978-1-4244-3803-7. doi: 10.1109/IROS.2009.5354701. URL http://ieeexplore.ieee.org/document/5354701/.

Kostrikov, I., Agrawal2, K. K., Dwibedi, D., Levine, S., and Tompson, J. DISCRIMINATOR-ACTOR-CRITIC: ADDRESSING SAMPLE INEFFICIENCY AND REWARD BIAS IN ADVERSARIAL IMITATION LEARNING. *International Conference in Learning Representations*, 2019.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, pp. 1–14, 2015. URL http://arxiv.org/abs/1509.02971.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. a., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.

Nair, A., McGrew, B., Andrychowicz, M., Zaremba, W., and Abbeel, P. Overcoming Exploration in Reinforcement Learning with Demonstrations. *International Conference on Robotics and Automation*, 2018. ISSN 0969-2290. doi: 10.1080/09692290.2013.809781. URL http://arxiv.org/abs/1709.10089.

Peng, X. B., Abbeel, P., Levine, S., and van de Panne, M. DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills. *Transactions on Graphics (Proc. ACM SIGGRAPH)*, 37(4), 2018. doi: 10.1145/3197517.3201311. URL http://arxiv.org/abs/1804.02717%0Ahttp://dx.doi.org/10.1145/3197517.3201311.

Pomerleau, D. A. ALVINN: an autonomous land vehicle in a neural network. *Advances in Neural Information Processing Systems*, pp. 305–313, 1989. URL https://papers.nips.cc/paper/95-alvinn-an-autonomous-land-vehicle-in-a-neural-network.pdfhttp://dl.acm.org/citation.cfm?id=89851.89891.

Rajeswaran, A., Kumar, V., Gupta, A., Schulman, J., and Sep, L. G. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. *Robotics: Science and Systems*, 2018.

Riedmiller, M., Hafner, R., Lampe, T., Neunert, M., Degrave, J., Wiele, T. V. D., Mnih, V., Heess, N., and Springenberg, T. Learning by Playing – Solving Sparse Reward Tasks from Scratch. *Internation Conference in Machine Learning*, 2018.

Ross, S., Gordon, G. J., and Bagnell, J. A. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. *International Conference on Artificial Intelligence and Statistics*, 2011.

Sasaki, F., Yohira, T., and Kawaguchi, A. Sample Efficient Imitation Learning for Continuous Control. *International Conference in Learning Representationsa*, pp. 1–15, 2019.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. Universal Value Function Approximators. *Internation Conference in Machine Learning*, 2015. URL http://jmlr.org/proceedings/papers/v37/schaul15.pdf.

Schroecker, Y., Vecerik, M., and Scholz, J. Generative predecessor models for sample-efficient imitation learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkeVsiAcYm.

Schulman, J., Moritz, P., Jordan, M., and Abbeel, P. Trust Region Policy Optimization. *International Conference in Machine Learning*, 2015.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., Van Den Driessche, G., Graepel, T., and Hassabis, D. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 10 2017. ISSN 14764687. doi: 10.1038/nature24270. URL http://arxiv.org/abs/1610.00633.

Stadie, B. C., Abbeel, P., and Sutskever, I. Third-Person Imitation Learning. *International Conference in Learning Representations*, 3 2017. URL http://arxiv.org/abs/1703.01703.

Sun, W., Bagnell, J. A., and Boots, B. Truncated Horizon Policy Search: Combining Reinforcement Learning & Imitation Learning. pp. 1–14, 2018. ISSN 0004-6361. doi: 10.1051/0004-6361/201527329. URL http://arxiv.org/abs/1805.11240.

Todorov, E., Erez, T., and Tassa, Y. MuJoCo : A physics engine for model-based control. pp. 5026–5033, 2012.

Vecerik, M., Hester, T., Scholz, J., Wang, F., Pietquin, O., Piot, B., Heess, N., Rothörl, T., Lampe, T., and Riedmiller, M. Leveraging Demonstrations for Deep Reinforcement Learning on Robotics Problems with Sparse Rewards. pp. 1–11, 2017.

Vinyals, O., Babuschkin, I., Chung, J., Mathieu, M., Jaderberg, M., Czarnecki, W. M., Dudzik, A., Huang, A., Georgiev, P., Powell, R., Ewalds, T., Horgan, D., Kroiss, M., Danihelka, I., Agapiou, J., Oh, J., Dalibard, V., Choi, D., Sifre, L., Sulsky, Y., Vezhnevets, S., Molloy, J., Cai, T., Budden, D., Paine, T., Gulcehre, C., Wang, Z., Pfaff, T., Pohlen, T., Wu, Y., Yogatama, D., Cohen, J., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Apps, C., Kavukcuoglu, K., Hassabis, D., and Silver, D. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. Technical report, 2019.

Zhu, H., Gupta, A., Rajeswaran, A., Levine, S., and Kumar, V. Dexterous Manipulation with Deep Reinforcement Learning: Efficient, General, and Low-Cost. 10 2018a. URL http://arxiv.org/abs/1810.06045.

Zhu, Y., Wang, Z., Merel, J., Rusu, A., Erez, T., Cabi, S., Tunyasuvunakool, S., Kramár, J., Hadsell, R., de Freitas, N., and Heess, N. Reinforcement and Imitation Learning for Diverse Visuomotor Skills. *Robotics: Science and Systems*, 2018b. URL http://arxiv.org/abs/1802.09564.

Ziebart, B. D., Maas, A., Bagnell, J. A., and Dey, A. K. Maximum Entropy Inverse Reinforcement Learning. pp. 1433–1438, 2008.

A. Related Work

Imitation Learning can be seen as an alternative to reward crafting to train desired behaviors. There are many ways to leverage demonstrations, from Behavioral Cloning (Pomerleau, 1989) that directly maximizes the likelihood of the expert actions under the training agent policy, to Inverse Reinforcement Learning that extracts a reward function from those demonstrations and then trains a policy to maximize it (Ziebart et al., 2008; Finn et al., 2016; Fu et al., 2018). Another formulation close to the later introduced by Ho & Ermon (2016) is Generative Adversarial Imitation Learning (GAIL), explained in details in the next section. Originally, the algorithms used to optimize the policy were on-policy methods like Trust Region Policy Optimization (Schulman et al., 2015), but recently there has been a wake of works leveraging the efficiency of off-policy algorithms without loss in stability (Blondé & Kalousis, 2019; Sasaki et al., 2019; Schroecker et al., 2019; Kostrikov et al., 2019). This is a key capability that we are going to exploit later on.

Unfortunately most work in the field cannot outperform the expert, unless another reward is available during training (Vecerik et al., 2017; Gao et al., 2018; Sun et al., 2018), which might defeat the purpose of using demonstrations in the first place. Furthermore, most tasks tackled with these methods consist on tracking expert state trajectories (Zhu et al., 2018b; Peng et al., 2018), but can't adapt to unseen situations.

In this work we are interested in goal-conditioned tasks, where the objective is to be able to reach any state upon demand. This kind of multi-task learning are pervasive in robotics, but challenging if no reward-shaping is applied. Relabeling methods like Hindsight Experience Replay (Andrychowicz et al., 2017) unlock the learning even in the sparse reward case (Florensa et al., 2018a). Nevertheless, the inherent breath-first nature of the algorithm might still make very inefficient learning to learn complex policies. To overcome the exploration issue we investigate the effect of leveraging a few demonstrations. The closest prior work is by Nair et al. (2018), where a Behavioral Cloning loss is used with a Q-filter. We found that a simple annealing of the Behavioral Cloning loss (Rajeswaran et al., 2018) works better. Furthermore, we also introduce a new relabeling technique of the expert trajectories that is particularly useful when only few demonstrations are available. We also experiment with Goal-conditioned GAIL, leveraging the recently shown compatibility with off-policy algorithms. For a more comprehensive review of related work, please see Appendix.

B. Goal-conditioned Behavioral Cloning

The most direct way to leverage demonstrations $\{(s_0^j, a_0^j, s_1^j, ...)\}_{j=0}^D$ is to construct a data-set $\mathcal{D}$ of all state-action-goal tuples (s_t^j, a_t^j, g^j), and run a supervised regression algorithm. In the goal-conditioned case and assuming a deterministic policy $\pi_\theta(s, g)$, the loss is:

$$\mathcal{L}_{BC}(\theta, \mathcal{D}) = \mathbb{E}_{(s_t^j, a_t^j, g^j)\sim\mathcal{D}}\left[\|\pi_\theta(s_t^j, g^j) - a_t^j\|_2^2\right]$$

This loss and its gradient are computed without any additional environments samples from the trained policy π_θ. This makes it particularly convenient to combine a gradient descend step based on this loss with other policy updates. In particular we can use a standard off-policy Reinforcement Learning algorithm like DDPG (Lillicrap et al., 2015), where we fit the $Q_\phi(a, s, g)$, and then estimate the gradient of the expected return as: $\nabla_\theta \hat{J} = \frac{1}{N}\sum_{i=1}^N \nabla_a Q_\phi(a, s, g)\nabla_\theta \pi_\theta(s, g)$. The improvement guarantees with respect to the task reward are lost when we combine the BC and the deterministic policy gradient updates, but this can be side-stepped by either applying a Q-filter: $\mathbb{1}\{Q(s_t, a_t, g) > Q(s_t, \pi(s_t, g), g)\}$ to the BC loss as proposed in (Nair et al., 2018), or by annealing it as we do in our experiments, which allows the agent to eventually outperform the expert.

C. Algorithm

All possible variants we study are detailed in Algorithm 1 as presented in appendix. In particular, $\alpha = 0$ falls back to pure Behavioral Cloning, $\beta = 0$ removes the BC component, $p = 0$ doesn't relabel agent trajectories, $\delta_{GAIL} = 0$ removes the discriminator output from the reward, and EXPERT RELABEL indicates whether the here explained expert relabeling should be performed.

D. Environments, Hyperparameters and Architectures

In the two environments, i.e. Four Rooms environment and Fetch Pick & Place, the task horizons are set to 300 and 100 respectively. The discount factors are $\gamma = 1 - \frac{1}{H}$. In all experiments, the Q function, policy and discriminator are paramaterized by fully connected neural networks with two hidden layers of size 256. DDPG is used for policy optimization and hindsight probability is set to $p = 0.8$. The initial value of the behavior cloning loss weight β is set to 0.1 and is annealed by 0.9 per 250 rollouts collected. The initial value of the discriminator reward weight δ_{GAIL} is set to 0.1. We found empirically that there is no need to anneal δ_{GAIL} .

Algorithm 1 Goal-conditioned Imitation Learning

Input: Demonstrations $\mathcal{D} = \left\{(s_0^j, a_0^j, s_1^j, ..., g^j)\right\}_{j=0}^{D}$, replay buffer $\mathcal{R} = \{\}$, policy $\pi_\theta(s, g)$, discount γ, hindsight probability p
while not done **do**
  # Sample rollout
  $g \sim \mathcal{R} \cup \mathcal{D}$
  Use $\pi(\cdot, g)$ to sample $(s_0, a_0, s_1, ...) \rightarrow \cup\mathcal{R}$
  # Sample from buffers
  $\left\{(s_t^j, a_t^j, s_{t+1}^j, g^j)\right\} \sim \mathcal{D}$, $\left\{(s_t^i, a_t^i, s_{t+1}^i, g^i)\right\} \sim \mathcal{R}$
  # Relabel agent
  if HER **then**
    for each i, with probability p **do**
      $g^i \leftarrow s_{t+k}^i, \quad k \sim \text{Unif}\{t+1, \ldots, T^i\}$
    end for
  end if
  if EXPERT RELABEL **then**
    $g^j \leftarrow s_{t+k'}^j, \quad k' \sim \text{Unif}\{t+1, \ldots, T^j\}$
  end if
  $r_t^h = \mathbb{1}\left[s_{t+1}^h == g^h\right]$
  if $\delta_{GAIL} > 0$ **then**
    $\Psi \leftarrow \min_\psi \mathcal{L}_{GAIL}(D_\psi, \mathcal{D}, \mathcal{R})$
    $r_t^h = (1 - \delta_{GAIL}) r_t^h + \delta_{GAIL} \log D_\psi(a_t^h, s_t^h, g^h)$
  end if
  # Fit Q_ϕ
  $y_t^h = r_t^h + \gamma Q_\phi(\pi(s_{t+1}^h, g^h), s_{t+1}^h, g^h)$
  $\phi \leftarrow \min_\phi \sum_h \|Q_\phi(a_t^h, s_t^h, g^h) - y_t^h\|$
  # Update Policy
  $\theta += \alpha \nabla_\theta \hat{J} - \beta \sum_h \nabla_\theta \mathcal{L}_{BC}(\theta, (a_t^h, s_t^h, g^h))$
  Anneal δ_{GAIL} and β
end while

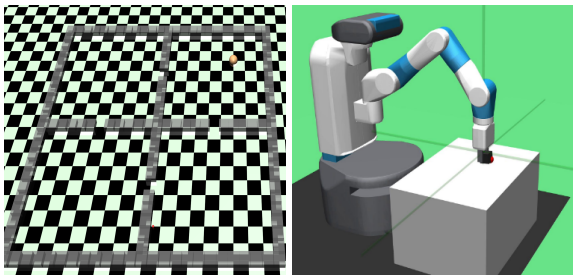
For experiments with sub-optimal expert in section 4.5, ϵ is set to 0.4 and 0.5, and σ_α is set to 1.5 and 0.3 respectively for Four Rooms environment and Fetch Pick & Place.

E. Effect of Different Input of Discriminator

We trained the discriminator in three settings:

- current state and goal: (s, g)
- current state, next state and goal: (s, s', g)
- current state, action and goal: (s, a, g)

We compare the three different setups in Fig. 7 and 8.



(a) Continuous Four rooms (b) Fetch Pick & Place

Figure 5. Environments where we test the use of demonstrations

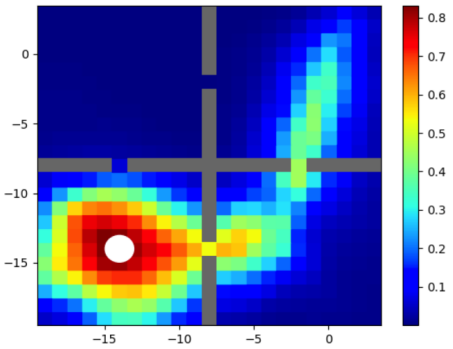
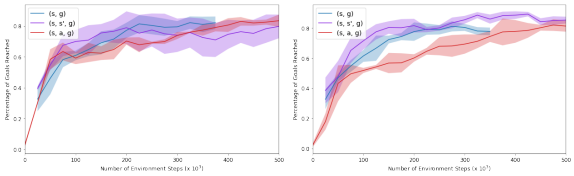
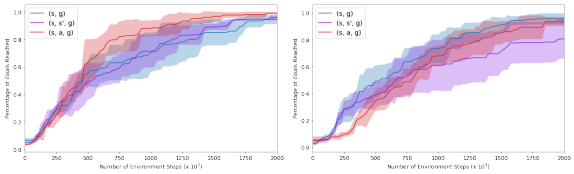


Figure 6. Output of the Discriminator $D(\cdot, g)$ when the goal is the white point in the lower left, and the start is always at the top right.



(a) 12 demos (b) 6 demos

Figure 7. Study of different discriminator inputs for goal-GAIL in Continuous Four Rooms



(a) 12 demos (b) 6 demos

Figure 8. Study of different discriminator inputs for goal-GAIL in Fetch Pick & Place