# Goal-conditioned Imitation Learning

**Anonymous Authors**[1]

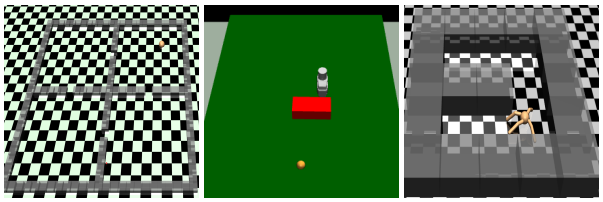
## Abstract

In robotics we are often interested in being able to reach a wide range of configurations. But designing multi-task rewards for Reinforcement Learning (RL) is challenging because it needs to convey the desired tasks, be efficient to optimize, and be easy to compute. The latter is particularly problematic because detecting whether the desired configuration is reached might require considerable supervision and instrumentation. Recently, methods like Hindsight Experience Replay (HER) have 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.

## 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., 2018). This is due to the difficulty of setting up the same learning environment that is enjoyed in simulation. Two of the most critical assumptions that are hard to obtain in the real world are the access to a reward function, and the reset capability.

A very versatile 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, although it used special resets, and the reward was in fact an $\epsilon$-ball around the goal, which only makes sense in lower-dimensional state-spaces. More recently the method was shown to work directly from vision (Nair et al., 2018b), and even only with the indicator reward of exactly matching observation and goal (Florensa et al., 2018a).



(a) Four rooms (b) Block Pusher (c) Ant Maze

*Figure 1.* Environments where we test the use of demonstrations

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 practicals examples of this are pick-and-place, or navigating narrow corridors between rooms, as illustrated in Fig. 1 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 towards

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

the bottlenecks, but this defeats the purpose of trying to learn without direct 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), 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, were 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 we observe that these methods can be run in a complete reset-free fashion, hence overcoming another limitation of RL in the real world.

## 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 $\pi_\theta$ 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 $\gamma^h$, 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. Related Work

Imitation Learning can be seen as an alternative to reward crafting to train a 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 (**?**), 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 leveraging a few demonstrations. The closes prior work is by Nair et al. (2018a), where they use a Behavioral Cloning loss with a Q-filter. We found that a simple annealing (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.

## 4. Methodology

In this section we describe the different algorithms we compare to running only Hindsight Experience Replay (Andrychowicz et al., 2017). First we revisit adding a Behavioral Cloning loss to the policy update as in (Nair et al., 2018a), then we propose a novel expert relabeling technique, and finally we formulate for the first time a goal-conditioned GAIL algorithm.

### 4.1. 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 $\pi_\theta$. 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:

$$\nabla_\theta \hat{J} = \frac{1}{N}\sum_{i=1}^{N} \nabla_a Q_\phi(a, s, g)\nabla_\theta \pi_\theta(s, g)$$

In our goal-conditioned case, the $Q$ fitting can also benefit form "relabeling" like done in HER (Andrychowicz et al., 2017). 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., 2018a), or by annealing it as we do in our experiments, which allows to eventually outperform the expert.

### 4.2. Goal-conditioned GAIL

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 all 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). Our experiments don't actually have high-dimensionality, so the benefit of Goal-conditioned GAIL is lesser, but we still introduce here the details of the algorithm. For goal-conditioned tasks, GAIL can be re-formulated with a discriminator $D_\psi(a, s, g)$ trained by minimizing

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

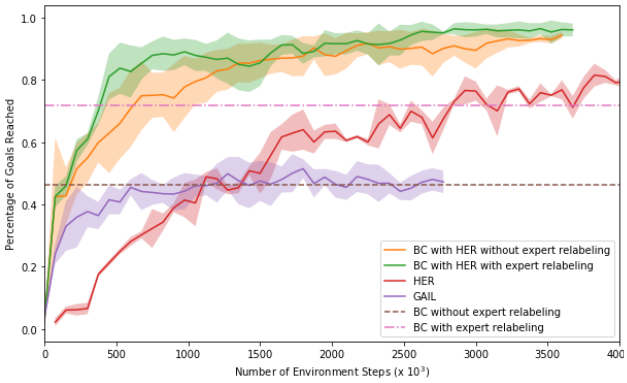
**Algorithm 1** Goal-conditioned Imitation Learning

**Input:** Demonstrations $\mathcal{D} = \{(s_0^j, a_0^j, s_1^j, ..., g^j)\}_{j=0}^D$, replay buffer $\mathcal{R} = \{\}$, policy $\pi_\theta(s, g)$, discount $\gamma$, 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*
  $\{(s_t^j, a_t^j, s_{t+1}^j, g^j)\} \sim \mathcal{D}$, $\{(s_t^i, a_t^i, s_{t+1}^i, g^i)\} \sim \mathcal{R}$
  *# 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**
    $g^j \leftarrow s_{t+k'}^j, \quad k' \sim Unif\{t+1, T_j\}$
  **end if**
  $r_t^h = \mathbb{1}[s_{t+1}^h == g^h]$
  **if** GAIL **then**
    $\Psi \leftarrow \min_\psi \mathcal{L}_{GAIL}(D_\psi, \mathcal{D}, \mathcal{R})$
    $r_t^h += \log D_\psi(a_t^h, s_t^h, g^h)$
  **end if**
  *# Fit $Q_\phi$*
  $y_t^h = r_t^h + \gamma Q_\phi(\pi(s_{t+1}^h), s_{t+1}^h, g^h)$
  $\phi \leftarrow \min_\phi \sum_h \|Q_\phi(a_t^h, s_t^h, g^h) - y_t^h\|$
  *# Update Policy*
  $\theta += \alpha\nabla_\theta \hat{J} - \beta \sum_h \nabla_\theta \mathcal{L}_{BC}(\theta, (a_t^h, s_t^h, g^h))$
**end while**

Once the discriminator is fitted, we can run our favorite RL algorithm on the reward $logD_\psi(a_t^h, s_t^h, g^h)$. In the goal-conditioned case we also supplement with the indicator reward $r_t^h = \mathbb{1}[s_{t+1}^h == g^h]$. This combination is slightly tricky because now the fitted $Q_\phi$ does not have the same clear interpretation it has when only one of the two rewards is used. Nevertheless, both rewards are pushing the policy towards the goals, so it shouldn't be too conflicting.

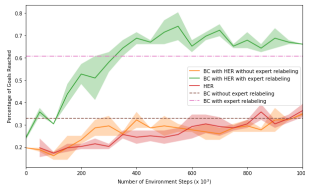
### 4.3. Relabeling the expert

The expert trajectories have been 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! Indeed that demonstration also went through the state $s_{t+k}^j$, so if that was the goal the expert would also have generated this transition. All possible variants we study are




(a) Four rooms

(b) Block Pusher



(c) Ant Maze

*Figure 2.* Learning curves based on 20 demos

detailed in Algorithm 1. In particular, $\alpha = 0$ falls back to pure Behavioral Cloning, $\beta = 0$ removes the BC component, $p = 0$ doesn't relabel agent trajectories, EXPERT RELABEL indicates whether the here explained expert relabeling should be performed, and GAIL indicates whether the reward should be supplemented with the discriminator output. In the next section we test all these variants in the diverse environments depicted in Fig. 1.

# 5. Experiments

For all experiments, we evaluate the percentage of goals in the goal space the agent is able to reach after each iteration.

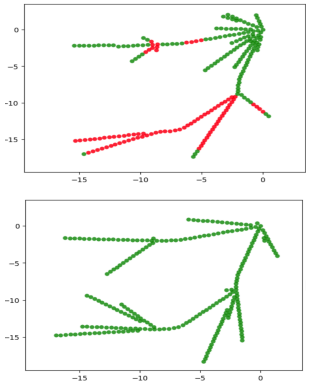
## 5.1. Tasks

Experiments are conducted in three continuous environments in MuJoCo (Todorov et al., 2012).

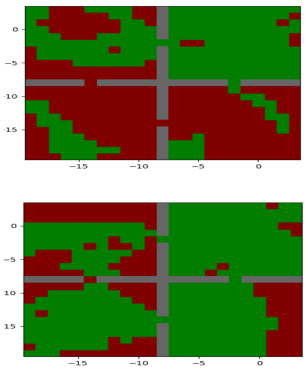
**Four rooms environment**: A point mass navigates in the four rooms from Fig. 1a. The action space of the agent is continuous and specifies the desired change in state space.

**Block pusher** A 3 DOF robot hand is trained to push a block to specified places on the table. The action space is 3-dimensional and specifies the desired change of velocity. The goal space is the position of the block. This environment is a simpler version than the one described in (Clavera et al., 2017), but with indicator reward instead of the original shaping.

**Ant navigation** The ant is trained to reach specified places in a U shaped maze as described in (Duan et al., 2016).



(a) Performance on reaching states visited in demonstrations. Each state is colored in green or red, according to whether the policy reaches it when conditioned on it



(b) Performance on reaching any possible state. Each cell is colored green if the policy can reach the center of it, when conditioned on it, and red otherwise.

*Figure 3.* 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)

## 5.2. Algorithm Comparison

The different algorithms proposed in Section 4 are compared to each other in Fig. 2. The performance of running pure Behavioral Cloning is plotted as a horizontal dotted line given that it does not require any additional environment steps. We observe that combining BC with HER always produces faster learning than running just HER, and it reaches higher final performance than running pure BC with only 20 demonstrations. Furthermore, no matter if combined with HER or not, our expert relabeling technique brings considerable performance boosts in all but the pusher environment where it's more indifferent to this relabeling. We also perform a further analysis of the benefit of the expert relabeling in the four-rooms environment because it is easy to visualize in 2D the goals the agent can reach. We see in Fig. 3 that without the expert relabeling sometimes the agent fails to learn how to reach intermediate states visited in the middle of a demonstration.

# 6. Conclusions and Future Work

We have investigated different self-supervised improvements on Imitation Learning. Our novel expert relabeling is proved to be effective in most environments. The biggest disappointment has been the under-performance of GAIL in these simple environments. We plan to test the many tricks described by (Blondé & Kalousis, 2019), as well as stronger regularizations like proposed by (Peng et al., 2019). We are also very excited to test the performance of these methods in vision-based tasks.

## 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.

Clavera, I., Held, D., and Abbeel, P. Policy Transfer via Modularity. *International Conference on Intelligent Robots and Systems*, pp. 1537–1544, 2017.

Duan, Y., Chen, X., Schulman, J., and Abbeel, P. Benchmarking Deep Reinforcement Learning for Continuous Control. *International Conference in Machine Learning*, 2016. URL http://arxiv.org/abs/1604.06778.

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.

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*, 2018a. ISSN 0969-2290. doi: 10.1080/09692290.2013.809781. URL http://arxiv.org/abs/1709.10089.

Nair, A., Pong, V., Dalal, M., Bahl, S., Lin, S., and Levine, S. Visual Reinforcement Learning with Imagined Goals. *Adavances in Neural Information Processing Systems*, 2018b.

Peng, X. B., Kanazawa, A., Toyer, S., Abbeel, P., and Levine, S. VARIATIONAL DISCRIMINATOR BOTTLENECK: IMPROVING IMITATION LEARNING, INVERSE RL, AND GANS BY CONSTRAINING INFORMATION FLOW. *International Conference in Learning Representations*, pp. 1–27, 2019. URL https://xbpeng.github.io/projects/VDB/2019_VDB.pdf.

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.

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 2018. URL http://arxiv.org/abs/1810.06045.

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

# A. Hyperparameters and Architectures

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 $p = 0.8$.

In the three environments (four rooms, block pushing, ant maze navigation), the horizon is set to $H = 500, 100, 1000$ respectively. The discount factors are set to $\gamma = 1 - \frac{1}{H}$.