TRANSFER LEARNING VIA DIVERSE POLICIES IN VALUE-RELEVANT FEATURES

Matthew Smith*, Jelena Luketina*, Maximilian Igl & Shimon Whiteson
Department of Computer Science
University of Oxford
Oxford, United Kingdom
{msmith,jelena.luketina}@cs.ox.ac.uk

ABSTRACT

Directly optimizing for reward is impractical in complex, sparse reward environments where agents are unlikely to encounter any reward unless they follow near optimal policies. Various information-theoretic methods (Gregor et al. (2016); Eysenbach et al. (2019); Sharma et al. (2020)) remedy this problem by learning a set of diverse, identifiable behaviors (or skills) independent of reward from the environment. However, being entirely task-agnostic, these methods often discover sets of trivially diverse instead of useful behaviours, and often rely on heuristics to define the space in which skills need to be diverse. To overcome this limitation, we propose a transfer learning approach that leverages learning of features relevant to value prediction to specify the space in which skills should be diverse. Concretely, our method employs three phases. First, we utilize standard reinforcement learning to learn a good representation by optimizing for reward on a source task. That representation is then employed in a phase of unsupervised, information-theoretic optimization of policies, which constructs skills that are both diverse, and able to manipulate reward relevant features. Lastly, these skills are used in the final phase to inform exploration on a target task. Our preliminary results demonstrate how this approach leads to the emergence of task-relevant skills.

1 INTRODUCTION

The automatic learning of useful skills is a fundamental challenge in scaling reinforcement learning (RL), particularly in temporally extended, or sparse-reward environments where exploration and credit assignment are difficult. Such behavioural abstraction reduces the effective time horizon of RL problems, allowing for faster propagation of rewards, as well as more coordinated exploration. If the skills are well constructed, these advantages can even speed up reward discovery and credit assignment enough to outweigh the burden of employing more complex policies (Nachum et al., 2019). Furthermore, if learned skills are modular, they can also provide an effective vehicle for transfer learning, as each skill constitutes a partial solution to the environment.

Recent work (Gregor et al., 2016; Eysenbach et al., 2019; Achiam et al., 2018; Sharma et al., 2020) has investigated the skill discovery problem from an unsupervised perspective, leading to a set of related information-theoretic objectives which optimize a set of distinct skills according to the set of discriminable states and transitions that they visit. However these methods often require learning large sets of skills in order to get useful or interesting policies, as many of the skills learned can trivially satisfy the optimization criteria, that is, be discriminable from one another, but in task irrelevant ways. A common solution to this problem has been hand specifying the space in which the skills should be diverse, such as using only location features when the downstream task is navigation Eysenbach et al. (2019); Sharma et al. (2020), an approach which may not be viable for many tasks of interest.

Consider an agent learning to cook a stew, though without a thermometer. Here, the temperature of the food is clearly a task-relevant feature that must be learned. After learning to make stew, if the agent can learn to manipulate this learned temperature feature in different ways, it becomes much

*Denotes equal contribution authors

easier to add new, more complex dishes to the agent's repertoire, which may involve sequences of temperatures. Furthermore, it will be easier for the agent to learn to make dishes that need to be chilled, a skill that would yield negative rewards in the original task. In a fully unsupervised setting, the agent may never learn this particularly concise feature, in favor of arbitrary combinations of ingredients, which could be easier to manipulate.

In this paper, we propose a transfer learning approach to skill discovery by combining supervised representation learning and unsupervised skill discovery. We first train an agent on a small set of source tasks to learn a state representation that captures features relevant to the value function. The agent can then be trained in the target or source task using reward-agnostic information-theoretic skill discovery to learn a diverse set of skills that manipulate this representation. We find that this procedure leads to skills are more meaningful to humans and consistently reach states that are diverse in features relevant for the set of tasks of interest.

2 PRELIMINARIES

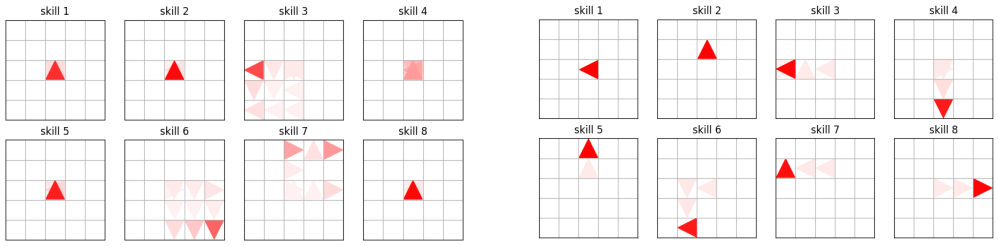
We define a discounted Markov Decision Process (MDP) as a tuple $M = \langle S, A, T, r, P_0, \gamma \rangle$ where S is the set of states, A the set of actions, T the transition probability function $T : S \times A \times S \to [0, 1]$, r the reward function $r : S \times A \times S \to \mathbb{R}$, P_0 the initial state distribution $P_0 : S \to [0, 1]$, and $\gamma \in [0, 1)$ is a discount factor. The objective is to find a parametric policy $\pi_\theta(a|s) = p(A = a|S = s)$ that maximizes the expected discounted cumulative return $\mathcal{J}(\theta) = \mathbb{E}_\tau[\sum_{t=0}^{\infty} \gamma^t r(a_t, s_t)]$, where $\tau = \{s_0, a_0, s_1, ...\}$ is a trajectory generated under π_θ: $s_0 \sim P_0$, $a_t \sim \pi_\theta(a_t|s_t)$, $s_{t+1} \sim T(s_{t+1}|s_t, a_t)$. We are interested in a case where the agent is given access to an environment $\langle S, A, T, P_0 \rangle$ which can be freely explored, with the goal of being able to later solve a range of specific tasks as specified by $\langle r_i, \gamma_i \rangle_{i=1...N_i}$. The aim is to leverage access to this environment to learn a set of behaviors or skills that allow faster learning on those specific tasks.

A recently proposed family of methods for unsupervised skill discovery are variational option discovery (VOD) algorithms (Gregor et al., 2016; Eysenbach et al., 2019; Achiam et al., 2018). Here, a skill is defined as a policy $\pi_\theta(a|s, z)$ conditioned on a discrete latent variable z. VOD algoritms aim to discover a set of diverse and distinguishable skills, such that each skill can be uniquely identified from some statistic ξ_τ derived from the trajectory τ generated by $\pi_\theta(a|s, z)$. Typical choices for ξ_τ are a sub-sequence of visited states $\{s_0, s_k, s_{2k}...\}$ (Achiam et al., 2018), individual states s_t (Eysenbach et al., 2019) or terminal states $\xi_\tau = s_T$ (Gregor et al., 2016). Formally, the skills are maximizing the mutual information between z and the statistic ξ_τ, $I(z; \xi_\tau)$, sometimes including maximization of the entropy $H[\pi_\theta(a|s, z)]$ to ensure good exploration. Since computing mutual information is intractable, these methods maximize the variational lower bound instead, resulting in the objective:

$$\max_{\theta,\phi} \mathbb{E}_{z\sim P(z)}[\mathbb{E}_{\tau\sim\pi,T,P_0}[\log q_\phi(z|\xi_\tau)] - \log P(z) + \beta H[\pi_\theta(a|s, z)]], \tag{1}$$

where the latent distribution $P(z)$ is not learned and set to be uniform in order to prevent trivial solutions. The training procedure consists of two simultaneous optimization processes: training a classifier $q_\phi(z|\xi_\tau)$, referred to as discriminator, to minimize the negative log-likelihood loss $-\log q_\phi(z|\xi_\tau)$; and training a skill-conditional policy using a pseudo-reward that is proportional to the negative discriminator loss. In other words, discriminator is trained to correctly predict skills from the trajectories, and policies are rewarded for producing trajectories that discriminator can distinguish.

In sufficiently large state spaces where agent has good control over the state, diversity in the full state is not sufficient to learn skills that are useful, i.e., relevant for future tasks. For example, the set of skills discovered by MuJoCo Ant agent trained with DIAYN (Eysenbach et al., 2019), tends to rely on easily controllable features like joint positions and orientation to differentiate. Consequentially, resulting skill are not capturing behaviours like different walking gaits or even involve significant movement in the x-y plane, rendering them useless for most tasks of interest, like navigation. In order to find skills that are useful for navigation tasks, the common approach has been limiting the observation space of the discriminator to the position of the agent's centre of mass (referred to as x-y prior) (Eysenbach et al., 2019; Sharma et al., 2020). However, there are many tasks in which the relevant features of the state space are unknown or hard to specify in this manner.



(a) DIAYN (b) DIAYN with pretraining

Figure 1: Visualizing state visitation in the subspace of interest under eight different skills trained using DIAYN (Eysenbach et al., 2019) (a) and DIAYN with pretrained representation (b). Color intensity of the agent indicates the probability of an agent occupying given direction and position during an episode.

3 Our Method

We propose to utilize a small number of "example" or *source* tasks to extract a latent representation that only captures variations in the parts of the state which matter for the tasks of interest. By restricting the discriminator to such a latent representation, we learn skills that are not only diverse, but diverse in relevant features. The resulting method consists of three phases of learning.

In the first phase, the goal is to discover a set of relevant features $\psi(s)$ that defines a space for learning a more constrained and useful diversity metric. We train an actor-critic agent, which results in a policy $\pi(a|s)$ and a corresponding value function $V(s)$. While the features $\psi(s)$ can come from any hidden layer of the policy or value function, we use the last hidden layer of the value function network. We hypothesize this is a good choice when the value functions of target tasks are dependent on the similar sets of features, and by taking state representations from the higher layers of the network, we are likely capturing more abstract and compressed state features. In order to ensure generality of features and ensure the resulting representation no longer contains projections from the irrelevant parts of the state space, one could use regularization methods during training (Cobbe et al., 2019) or take the state representation from the distilled value function (Rusu et al., 2015).

Next, without access to the reward function, we train a set of skills using VOD methods. Instead of providing the discriminator with privileged information about the space in which the skills should be diverse (e.g. x-y prior), our approach instead employs the features $\psi(s)$ learned in the first phase. While our approach is compatible with any VOD method, in our experiments we employ DIAYN. The discriminator and skills are trained in parallel, with the discriminator trained to minimize the prediction loss at each state independently–i.e., $\mathcal{L}_\tau(\phi) = \sum_{t\in\{1..T\}} \mathcal{L}_t(\phi)$, where:

$$\mathcal{L}_t(\phi) = -\log q_\phi(z|\psi(s_t)). \tag{2}$$

The resulting pseudo-reward for the skill is:

$$r_t = \log q_\phi(z|\psi(s_t)) - \log P(z), \tag{3}$$

with the parameters of the state embedding $\psi(s)$ fixed and uniform distribution $P(z)$.

The result is a set of skills $\pi_\theta(a|s,z)$ parametrized through z, which can be used on downstream tasks either by selecting and fine-tuning the best performing skills or by training a meta-controller $\pi(z|s)$ as in Eysenbach et al. (2019).

4 Empirical Evaluation

Gridworld Experiments We start by testing the proposed approach on a modification of a simple 7×7 gridworld from (Chevalier-Boisvert et al., 2018), which has been expanded with extra controllable dimensions. In addition to being able to rotate and move forward, the agent can increase or decrease the value of one of the four added "shadow" dimensions, depending on which of the four cardinal directions it is facing. Analogous to limb positions in Ant MuJoCo, these added dimensions are not informative for value functions on navigation tasks, but are easier to control than the agent's

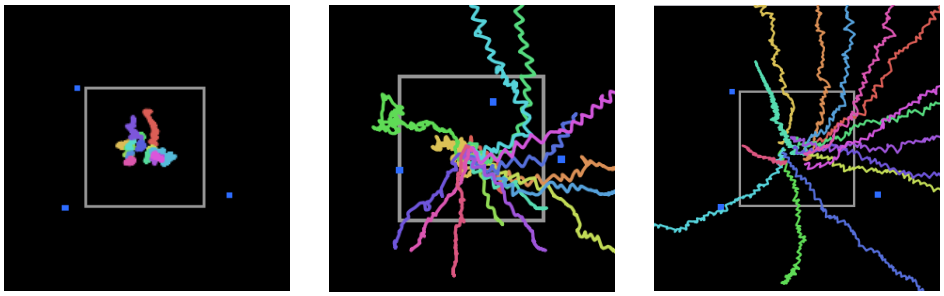


Figure 2: Comparison of trajectory traces in x-y plane of 16 different Ant skills: DIAYN (left), DIAYN with x-y prior (middle), DIAYN with pre-trained representation (right). The grey square is for scale, denoting a 2m$\times$2m area.

position. These dimensions could be interpreted as, e.g., controlling the brightness of the wall the agent is currently facing.

To learn which state features are relevant for navigation tasks, we train a PPO agent on a generic navigation task which involves reaching to one of the two corners of the room. Proceeding to the unsupervised training phase, we fix the learned value function embeddings, and use them as input features to the discriminator.

In Figure 1, we visualize which states are visited by resulting skills. With the added, more easily controllable space, most DIAYN skills never leave the initial position (3,3), and instead learn to differentiate by controlling the added space or agent orientation. Skills that do leave the initial position do not reliably reach a particular grid position. By pre-training on navigation tasks, we obtain a representation that ignores added dimensions of the state space as they do not affect the value function. In this case, this leads to differentiation of skills based on the position and orientation of the agent.

Mujoco Experiments Next we evaluate skills trained on MuJoCo Ant which, as shown in the literature (Eysenbach et al., 2019; Sharma et al., 2020), requires hard coding restrictions on the observation space of the discriminator (an x-y prior) in order to obtain skills useful for complex navigation tasks. As source tasks, we use three simple navigational tasks. In each, the agent needs to reach a specific goal, chosen from three fixed goal positions. The agent receives a dense reward proportional to it's distance from the chosen goal. This, the value of a state in these tasks is determined mostly by the distance of the agent's center of mass to the goal. Here, we transfer the learned state embedding from the penultimate value network layer, before incorporation of goal information.

We compare the x-y traces of the agent's center of mass on rollouts of vanilla DIAYN skills, skills learned with the x-y prior and skills learned with pretraining (see Figure 2). The x-y traces show whether skills cover the x-y plane well, a good indicator for the usefulness of skills on downstream tasks. As reported in the literature, baseline skills trained with DIAYN do not venture far from the initial state position, whereas both DIAYN with the x-y prior and DIAYN with pretraining, explore the x-y plane well.

5 Related Work

A detailed overview of related work is provided in the appendix.

6 Discussion

In this work we investigated a particular failure mode of information-theoretic unsupervised skill discovery methods, which is that discriminability objective can often be trivially satisfied, leading to the discovery of skills that are useless for most tasks of interest. We proposed to address this problem via transfer of state representations, where the discriminator's state representation is transferred from the value function trained on a set of source tasks. Our empirical results demonstrate that skills obtained using this approach learn to manipulate more relevant features and better cover the space of interest. In future work, we plan to investigate the under-specification problem in more detail and examine the advantages of proposed approach for training on downstream tasks.

REFERENCES

Joshua Achiam, Harrison Edwards, Dario Amodei, and Pieter Abbeel. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*, 2018.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. `https://github.com/maximecb/gym-minigrid`, 2018.

Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. *arXiv preprint arXiv:1912.01588*, 2019.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *ICLR*, 2019.

Kevin Frans, Jonathan Ho, Xi Chen, Pieter Abbeel, and John Schulman. META LEARNING SHARED HIERARCHIES. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=SyX0IeWAW`.

Alexandre Galashov, Siddhant Jayakumar, Leonard Hasenclever, Dhruva Tirumala, Jonathan Schwarz, Guillaume Desjardins, Wojtek M Czarnecki, Yee Whye Teh, Razvan Pascanu, and Nicolas Heess. Information asymmetry in kl-regularized rl. 2018.

Sandeep Goel and Manfred Huber. Subgoal discovery for hierarchical reinforcement learning using learned policies. In *FLAIRS conference*, pp. 346–350, 2003.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Anna Harutyunyan, Will Dabney, Diana Borsa, Nicolas Heess, Remi Munos, and Doina Precup. The termination critic. *arXiv preprint arXiv:1902.09996*, 2019.

Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an embedding space for transferable robot skills. 2018.

Maximilian Igl, Andrew Gambardella, Nantas Nardelli, N Siddharth, Wendelin Böhmer, and Shimon Whiteson. Multitask soft option learning. *arXiv preprint arXiv:1904.01033*, 2019.

Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in neural information processing systems*, pp. 3675–3683, 2016.

Amy McGovern and Andrew G. Barto. Automatic discovery of subgoals in reinforcement learning using diverse density. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pp. 361–368, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1. URL `http://dl.acm.org/citation.cfm?id=645530.655681`.

Ofir Nachum, Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *arXiv preprint arXiv:1805.08296*, 2018a.

Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. Near-optimal representation learning for hierarchical reinforcement learning. *arXiv preprint arXiv:1810.01257*, 2018b.

Ofir Nachum, Haoran Tang, Xingyu Lu, Shixiang Gu, Honglak Lee, and Sergey Levine. Why does hierarchy (sometimes) work so well in reinforcement learning? *arXiv preprint arXiv:1909.10618*, 2019.

Marc Pickett and Andrew G Barto. Policyblocks: An algorithm for creating useful macro-actions in reinforcement learning. In *ICML*, pp. 506–513, 2002.

Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised skill discovery. *ICLR*, 2020.

Özgür Şimşek and Andrew G Barto. Skill characterization based on betweenness. In *Advances in neural information processing systems*, pp. 1497–1504, 2009.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 4499–4509, 2017.

Chen Tessler, Shahar Givony, Tom Zahavy, Daniel J Mankowitz, and Shie Mannor. A deep hierarchical approach to lifelong learning in minecraft. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Sebastian Thrun and Anton Schwartz. Finding structure in reinforcement learning. In *Advances in neural information processing systems*, pp. 385–392, 1995.

Dhruva Tirumala, Hyeonwoo Noh, Alexandre Galashov, Leonard Hasenclever, Arun Ahuja, Greg Wayne, Razvan Pascanu, Yee Whye Teh, and Nicolas Heess. Exploiting hierarchy for learning and transfer in kl-regularized rl. *arXiv preprint arXiv:1903.07438*, 2019.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pp. 3540–3549, 2017.

Markus Wulfmeier, Abbas Abdolmaleki, Roland Hafner, Jost Tobias Springenberg, Michael Neunert, Tim Hertweck, Thomas Lampe, Noah Siegel, Nicolas Heess, and Martin Riedmiller. Regularized hierarchical policies for compositional transfer in robotics. *arXiv preprint arXiv:1906.11228*, 2019.

A Related work

Several methods have been developed for the automatic discovery of skills in RL. Of particular recent interest, due to their unsupervised nature, is a category of information-theoretically motivated, reward-agnostic methods, which optimise policies according to various criteria, centered around notions of discriminability (Gregor et al., 2016; Eysenbach et al., 2019; Achiam et al., 2018; Sharma et al., 2020). However, these techniques are generally highly underconstrained, in that many sets of skills satisfy the optimization objective. This means that most of them can produce uninteresting or useless skills in practice, if significant heuristic guidance is not provided (Achiam et al., 2018). Hausman et al. (2018) combine dscriminability and task rewards, however, they aim to find multiple solutions to the specified task.

Often, skills are learned as part of a hierarchical policy. To learn a useful and diverse set of skills, one can rely on specified subgoals for each skill (Sutton et al., 1999; Kulkarni et al., 2016), which can be found analyzing the structure of the MDP (McGovern & Barto, 2001; Şimşek & Barto, 2009), previous policies (Goel & Huber, 2003; Tessler et al., 2017) or predictability (Harutyunyan et al., 2019). In the last case, predictability, in contrast to discriminability, describes the notion that skills should *terminate* in a predictable state, instead of *behave* in a way that is different from other skills. Subgoals can also be generated by a higher-level policy, either set in the state space (Nachum et al., 2018a) or in a latent space with learned semantics (Vezhnevets et al., 2017; Nachum et al., 2018b).

Several prior work also learn skills from a set of tasks (Thrun & Schwartz, 1995; Pickett & Barto, 2002). Frans et al. (2018) extract skills from a set of tasks, but contrary to our appraoch, are restricted to downstream tasks which can be achieved by a composition of the source task policies. Igl et al. (2019) relax this restriction by also learning termination functions, allowing the composition of sub-policies from the source tasks. More widely, the use of KL-regularization to transfer knowledge between tasks is investigated by (Teh et al., 2017) and by (Galashov et al., 2018; Tirumala et al., 2019), who explore how hierarchies, i.e. skills, can be used to implement various inductive biases. Without the complications that can arise by learning temporal abstractions, (Wulfmeier et al., 2019) show that sharing policies across closely related tasks can significantly speed up training.

However, all of the above approaches are restricted to policies that are exactly the same or 'close', as measured by the KL-divergence between policies, to the policies on the source tasks. In contrast, we aim to learn policies that are as diverse as possible, while using source tasks only to identify the relevant part of the state space we want to be diverse in.

B Implementation Details

Here we provide further implementation details for the experiments.

B.1 Gridworld Experiments

During RL training, we also initialize the agent at a random state to ensure generalization of the value function. The grid observation is embedded with convolutional layers, while the non-location observations – direction and the added dimensions – are concatenated and embedded with fully connected layers. The two resulting embeddings are concatenated before being passed to actor and critic heads. The goal, which identifies which of the two tasks the agent is currently in, is also embedded with a fully connected network and concatenated with the rest of the embeddings.

For the unsupervised training, we fix both the convolutional and fully-connected subnetworks for the value function and pass the resulting embeddings to the discriminator. The discriminator is parametrized as a single hidden layer neural network on top of those features. The skills architecture is similar to the policy in pre-training phase with the addition of a one-hot encoding of skill identity, which is embedded with fully-connected layers and concatenated with the location and non-location embeddings. In experiments with DIAYN, we use the same architecture for the discriminator and skills.

B.2 MuJoCo Experiments

Agents are trained with PPO, using standard MLPs for both actor and critic networks. We employ a variant of the Tanh squashing trick for MuJoCo envs (Haarnoja et al., 2018). The gear ratio for the ant agent is reduced to 30, as in Eysenbach et al. (2019). Agents receive reward proportional to the negative log distance from the goal, with a small epsilon added to prevent instability, as well as the control penalty that is standard in the Ant environment.