Maximum Entropy Gain Exploration for Long Horizon Multi-goal Reinforcement Learning

Silviu Pitis*
University of Toronto, Vector Institute
Toronto, Ontario, Canada
spitis@cs.toronto.edu

Harris Chan*
University of Toronto, Vector Institute
Toronto, Ontario, Canada
harris@cs.toronto.edu

Stephen Zhao
University of Toronto
Toronto, Ontario, Canada
stephen.zhao@mail.utoronto.ca

Bradly Stadie
Vector Institute
Toronto, Ontario, Canada
bstadie@vectorinstitute.ai

Jimmy Ba
University of Toronto, Vector Institute
Toronto, Ontario, Canada
jba@cs.toronto.edu

ABSTRACT

What goals should a multi-goal reinforcement learning agent pursue during training in long-horizon tasks? When the desired (test time) goal distribution is too distant to offer a useful learning signal, we argue that the agent should not pursue unobtainable goals. Instead, it should set its own intrinsic goals that maximize the entropy of the historical achieved goal distribution. We propose to optimize this objective by having the agent pursue past achieved goals in sparsely explored areas of the goal space, which focuses exploration on the frontier of the achievable goal set. We show that our strategy achieves an order of magnitude better sample efficiency than the prior state of the art on long-horizon multi-goal tasks including maze navigation and block stacking.

KEYWORDS

Multi-goal reinforcement learning; Curiosity-based Exploration; Empowerment; Long-horizon problem

ACM Reference Format:
Silviu Pitis, Harris Chan, Stephen Zhao, Bradly Stadie, and Jimmy Ba. 2020. Maximum Entropy Gain Exploration for Long Horizon Multi-goal Reinforcement Learning. In *ALA2020: Adaptive and Learning Agents 2020 , Auckland, New Zealand, May 2020,* IFAAMAS, 20 pages.

1 INTRODUCTION

Multi-goal reinforcement learning (RL) agents [25, 47, 53] learn goal-conditioned behaviors that can achieve and generalize across a range of different goals. Multi-goal RL forms a core component of hierarchical agents [35, 59], and has been shown to allow unsupervised agents to learn useful skills for downstream tasks [15, 21, 64]. Recent advances in goal relabeling [2] have made learning possible in complex, sparse-reward environments whose goal spaces are either dense in the initial state distribution [47] or structured as a curriculum [9]. But learning without demonstrations in sparse-reward, long-horizon environments remains a challenge [36, 62], as learning signal decreases exponentially with the horizon [39].

In this paper, we improve upon existing approaches to intrinsic goal setting and show how multi-goal agents can form an automatic behavioural goal curriculum that allows them to master long-horizon, sparse reward tasks. We begin with an algorithmic framework for goal-seeking agents that contextualizes prior work [4, 17, 37, 48, 64] and argue that past goal selection mechanisms are not well suited for long-horizon, sparse reward tasks (Section 2). By framing the long-horizon goal seeking task as optimizing an initially ill-conditioned distribution matching objective [30], we arrive at our unsupervised Maximum Entropy Goal Achievement (MEGA) objective, which maximizes the entropy of the past achieved goal set. This early unsupervised objective is annealed into the original supervised objective once the latter becomes tractable (Section 3).

We propose a practical algorithmic approach to maximizing entropy, which pursues past achieved goals in sparsely explored areas of the achieved goal distribution, as measured by a learned density model. The agent revisits and explores around these areas, pushing the frontier of achieved goals forward [14]. This strategy, similar in spirit to Baranes & Oudeyer [4] and Florensa et al. [17], encourages the agent to explore at the edge of its abilities, which avoids spending environment steps in pursuit of already mastered or unobtainable goals. When used in combination with hindsight experience replay and an off-policy learning algorithm, our method achieves more than an order of magnitude better sample efficiency than the prior state of the art on difficult exploration tasks, including long-horizon mazes and block stacking (Section 4). Finally, we draw connections between our approach and the empowerment objective [28, 51] and identify a key difference to prior work: rather than maximize empowerment on-policy by setting maximally diverse goals during training [20, 37, 48, 64], our proposed approach maximizes it off-policy by setting goals on the frontier of the past achieved goal set. We conclude with discussion of angles for future work (Sections 5-6).

2 THE LONG-HORIZON PROBLEM

2.1 Preliminaries

We consider the multi-goal reinforcement learning (RL) setting, described by a generalized Markov Decision Process (MDP) $\mathcal{M} = \langle S, A, T, G, [p_{dg}] \rangle$, where S, A, T, and G are the state space, action space, transition function and goal space, respectively [52, 58] and p_{dg} is an optional desired goal distribution. In the most general version of this problem each goal is a tuple $g = \langle R_g, \gamma_g \rangle$, where $R_g : S \to \mathbb{R}$ is a reward function and $\gamma_g \in [0, 1]$ is a discount factor [60], so that "solving" goal $g \in G$ amounts to finding an optimal policy in

[*] Equal contribution.

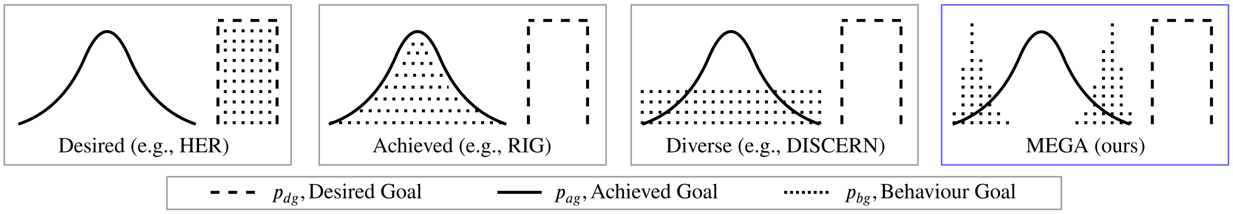

Fig. 1: Illustration of density-based Select mechanisms at start of training, when achieved (p_{ag}) and desired (p_{dg}) goal distributions are disconnected. HER samples goals from the desired distribution p_{dg}. RIG samples from the achieved distribution p_{ag}. DISCERN and Skew-Fit skew p_{ag} to sample diverse achieved goals. Our approach (MEGA) focuses on low density regions of p_{ag}. See Subsection 2.3.

the classical MDP $\mathcal{M}_g = \langle S, A, T, R_g, \gamma_g \rangle$. Although goal-oriented methods are general and could be applied to dense reward MDPs (including the standard RL problem, as done by Warde-Farley et al. [64], among others), we restrict our present attention to the sparse reward case, where each goal g corresponds to a set of "success" states, S_g, with $R_g : S \to \{-1, 0\}$ [47] defined as $R_g(s) = \mathbb{I}\{s \in S_g\} + c$. Following Plappert et al., we use base reward $c = -1$, which typically leads to more stable training than the more natural $c = 0$ (see Van Seijen et al. [63] for a possible explanation). We also adopt the usual form $S_g = \{s \mid d(\textsc{ag}(s), g) < \epsilon\}$, where $\textsc{ag} : S \to G$ maps state s to an "achieved goal" $\textsc{ag}(s)$ and d is a metric on G. An agent's "achieved goal distribution" p_{ag} is the distribution of goals achieved by states s (i.e., $\textsc{ag}(s)$) the agent visits (not necessarily the final state in a trajectory). Note that this may be on-policy (induced by the current policy) or historical, as we will specify below. The agent must learn to achieve success and, if the environment is not episodic, maintain it. In the episodic case, we can think of each goal $g \in G$ as specifying a skill or option $o \in \Omega$ [15, 59], so that multi-goal reinforcement learning is closely related to hierarchical reinforcement learning [35].

A common approach to multi-goal RL, which we adopt, trains a goal-conditioned state-action value function, $Q : S \times A \times G \to \mathbb{R}$, using an off-policy learning algorithm that can leverage data from other policies (past and exploratory) to optimize the current policy [53]. A goal-conditioned policy, $\pi : S \times G \to A$, is either induced via greedy action selection [33] or learned using policy gradients. Noise is added to π during exploration to form exploratory policy π_{explore}. Our continuous control experiments all use the DDPG algorithm [31], which parameterizes actor and critic separately, and trains both concurrently using Q-learning for the critic [65], and deterministic policy gradients [57] for the actor. DDPG uses a replay buffer to store past experiences, which is then sampled from to train the actor and critic networks.

2.2 Sparse rewards and the long horizon problem

Despite the success of vanilla off-policy algorithms in dense-reward tasks, standard agents learn very slowly—or not at all—in sparse-reward, goal-conditioned tasks [2]. In order for a vanilla agent to obtain a positive reward signal and learn about goal g, the agent must stumble upon g through random exploration *while it is trying to achieve* g. Since the chance of this happening when exploring randomly decreases exponentially with the horizon ("the long horizon problem") [39], successes are infrequent even for goals that are relatively close to the initial state, making learning difficult.

One way to ameliorate the long horizon problem is based on the observation that, regardless of the goal being pursued, (state, action, next state) transitions are unbiased samples from the environment dynamics. An agent is therefore free to pair transitions with any goal and corresponding reward, which allows it to use experiences gained in pursuit of one goal to learn about other goals ("goal relabeling") [25]. Hindsight Experience Replay (HER) [2] is a form of goal relabeling that relabels experiences with goals that are achieved later in the same trajectory. For every real experience, Andrychowicz et al. [2]'s `future` strategy produces k relabeled experiences, where the k goals are sampled uniformly from goals achieved by future states in the same trajectory. This forms an implicit optimization curriculum, and allows an agent to learn about any goal g it encounters during exploration.

Note, however, that a HER agent must still encounter g (or goals sufficiently similar to g) in order to learn about g, and the long horizon problem persists for goals that are too far removed from the agent's initial state distribution. This is illustrated in Figure 2, and is most easily understood by considering the tabular case, where no generalization occurs between a finite set of MDPs $\mathcal{M}_g$: since a learning signal is obtained only when transitioning into $s \in S_g$, the agent's achieved goal distribution must overlap with S_g for learning to occur. Empirically, this means that DDPG+HER agents that explore using only action noise or epsilon random actions fail to solve *long-horizon tasks*, whose desired goal distribution does not

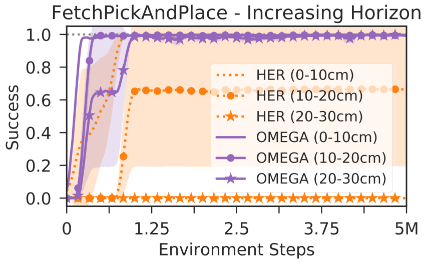

Fig. 2: Performance of a DDPG+HER agent that must lift a box to reach goals at increasing heights (3 seeds). As the horizon (desired height) increases, the agent loses the ability to solve the task in reasonable time. Our approach, OMEGA (Section 3), is robust to the horizon length. Specific details in Appendix.

Algorithm 1 Unified Framework for Multi-goal Agents

```
function TRAIN($*args$):
    Alternate between collecting experience using ROLLOUT and optimizing the parameters using OPTIMIZE.

function ROLLOUT (policy $\pi_{\text{explore}}$, buffer $\mathcal{B}$, $*args$):
    g ← SELECT($*args$)
    $s_0$ ← initial state
    for $t$ in $0 \dots T-1$ do
        $a_t, s_{t+1}$ ← execute $\pi_{\text{explore}}(s_t, g)$ in environment
        $r_t$ ← REWARD($s_t, a_t, s_{t+1}, g$)
        Store ($s_t, a_t, s_{t+1}, r_t, g$) in replay buffer $\mathcal{B}$

function OPTIMIZE (buffer $\mathcal{B}$, algorithm $\mathcal{A}$, parameters $\theta$):
    Sample mini-batch $B = \{(s, a, s', r, g)_i\}_{i=1}^{N} \sim \mathcal{B}$
    $B'$ ← RELABEL($B$, $*args$)
    Optimize $\theta$ using $\mathcal{A}$ (e.g., DDPG) and relabeled $B'$

function SELECT ($*args$):
    Returns a behavioural goal for the agent. Examples include the environment goal $g_{\text{ext}}$, a sample from the buffer of achieved goals $\mathcal{B}_{ag}$ [64], or samples from a generative model such as a GAN [17] or VAE [37]. Our approach (MEGA) selects previously achieved goals in sparsely explored areas of the goal space according to a learned density model.

function REWARD ($s_t, a_t, s_{t+1}, g$):
    Computes the environment reward or a learned reward function [37, 64].

function RELABEL ($B$, $*args$):
    Relabels goals and rewards in minibatch $B$ according to some strategy; e.g., don't relabel, future, mix future and generated goals [37], or rfaab (ours).
```

overlap with the initial state distribution. This includes the original version of `FetchPickAndPlace` (with all goals in the air) [2], block stacking [36], and mazes [62].

2.3 Setting intrinsic goals

We propose to approach the long-horizon problem by ignoring long-horizon goals: rather than try to achieve unobtainable goals, an agent can set its own intrinsic goals and slowly expand its domain of expertise in an unsupervised fashion. This is inspired by a number of recent papers on unsupervised multi-goal RL, to be described below. Our main contributions relative to past works are (1) a novel goal selection mechanism designed to address the long-horizon problem, and (2) a practical method to anneal initial unsupervised selection into training on the desired goals.

To capture the differences between various approaches, we present Algorithm 1, a unifying algorithmic framework for multi-goal agents. Variations occur in the subprocedures SELECT, REWARD, and RELABEL. The standard HER agent Andrychowicz et al. [2] SELECTS the environment goal g_{ext}, uses the environment REWARD and uses the `future` RELABEL strategy. Functions used by other agents are detailed in Appendix A. We assume access to the environment REWARD and propose a novel SELECT strategy—MaxEnt Goal Achievement (MEGA)—that initially samples goals from low-density regions of the achieved goal distribution. Our approach also leads to a novel RELABEL strategy, `rfaab`, which samples from Real, Future, Actual, Achieved, and behavioural goals (detailed in Appendix C).

Prior work also considers intrinsic SELECT functions. The approaches used by DISCERN [64], RIG [37] and Skew-Fit [48] select goals using a model of the past achieved goal distribution. DISCERN samples from a replay buffer (a non-parametric model), whereas RIG and Skew-Fit learn and sample from a variational autoencoder (VAE) [27]. These approaches are illustrated in Figure 1, alongside HER and MEGA. Prior density-based approaches were not tailored to the long-horizon problem; e.g., DISCERN was primarily focused on learning an intrinsic REWARD function, and left "the incorporation of more explicitly instantiated [SELECT] curricula to future work." By contrast, MEGA focuses on the low density, or sparsely explored, areas of the achieved goal distribution, forming a curriculum that crosses the gap between the initial state distribution and the desired goal distribution in record time. Although Diverse sampling (e.g., Skew-Fit) is less biased towards already mastered areas of the goal space than Achieved sampling (e.g., RIG), we show in our experiments that it still under-explores relative to MEGA.

MEGA's focus on the frontier of the achieved goal set makes it similar to SAGG-RIAC [4], which seeks goals that maximize learning progress, and Goal GAN [17], which seeks goals of intermediate difficulty.

3 MAXIMUM ENTROPY GOAL ACHIEVEMENT

3.1 The MEGA and OMEGA objectives

To motivate the MEGA objective, we frame exploration in episodic, multi-goal RL with goal relabeling as a distribution matching problem [30]. We note that the original distribution matching objective is ill-conditioned in long-horizon problems, which suggests maximizing the entropy of the achieved goal distribution (the MEGA objective). We then show how this can be annealed into the original objective (the OMEGA objective).

We start by noting that, for a truly off-policy agent, the actual goals used to produce the agent's experience do not matter, as the agent is free to relabel any experience with any goal. This implies that only the distribution of experience in the agent's replay buffer, along with the size of the buffer, matters for effective off-policy learning. How should an agent influence this distribution to accumulate useful data for achieving goals from the desired distribution p_{dg}?

Though we lack a precise characterization of which data is useful, we know that all successful policies for goal g pass through g, which suggests that useful data for achieving g monotonically increases with the number of times g is achieved during exploration. Past empirical results, such as the success of Andrychowicz et al. [2]'s `future` strategy and the effectiveness of adding expert demonstrations to the replay buffer [36], support this intuition. Assuming a relatively fixed initial state distribution and uniformly distributed p_{dg}[*], it follows that the intrinsic goal g^t at episode t should be chosen to bring the agent's historical achieved goal distribution p^t_{ag} closer to the desired distribution p_{dg}. We can formalize this as seeking g^t to minimize the following distribution matching objective:

$$J_{\text{original}}(p^t_{ag}) = \mathrm{D}_{\mathrm{KL}}(p_{dg} \parallel p^t_{ag}), \tag{1}$$

where p^t_{ag} represents the *historical* achieved goal distribution in the agent's replay buffer after executing its exploratory policy in pursuit of goal g^t. It is worth highlighting that objective (1) is a forward KL: we seek p_{ag} that "covers" p_{dg} [6]. If reversed, it would always be infinite when p_{dg} and the initial state distribution s_0 do not overlap, since p_{dg} cannot cover s_0.

So long as (1) is finite and non-increasing over time, the support of p_{ag} covers p_{dg} and the agent is accumulating data that can be used to learn about all goals in the desired distribution. In those multi-goal environments where HER has been successful (e.g., `FetchPush`), this is easily achieved by setting the behavioural goal distribution p_{bg} to equal p_{dg} and using action space exploration [47]. In long-horizon tasks, however, the objective (1) is usually ill-conditioned (even undefined) at the beginning of training when the supports of p_{dg} and p_{ag} do not overlap. While this explains why HER with action space exploration fails in these tasks, it isn't very helpful, as the ill-conditioned objective is difficult to optimize.

When p_{ag} does not cover p_{dg}, a natural objective is to expand the support of p_{ag}, in order to make the objective (1) finite as fast as possible. We often lack a useful inductive bias about which direction to expand the support in; e.g., a naive heuristic like Euclidean distance in feature space can be misleading due to dead-ends or teleportation [62], and should not be relied on for exploration. In absence of a useful inductive bias, it is sensible to expand the support as fast as possible, in any and all directions as in breadth-first search, which can be done by maximizing the entropy of the achieved goal distribution $H[p_{ag}]$. We call this the Maximum Entropy Goal Achievement (MEGA) objective:

$$J_{\text{MEGA}}(p^t_{ag}) = -H[p^t_{ag}], \tag{2}$$

The hope is that by maximizing $H[p_{ag}]$ (minimizing J_{MEGA}), the agent will follow a natural curriculum, expanding the size of its achievable goal set until it covers the desired distribution p_{dg} and objective (1) becomes tractable.

In the unsupervised case, where p_{dg} is not specified, the agent can stop at the MEGA objective. In the supervised case we would like the agent to somehow anneal objective (2) into objective (1). We can do this by approximating (2) using a distribution matching objective, where the desired distribution is uniform over the current support:

$$\tilde{J}_{\text{MEGA}}(p^t_{ag}) = \mathrm{D}_{\mathrm{KL}}(\mathcal{U}(\text{supp}(p^t_{ag})) \parallel p^t_{ag}). \tag{3}$$

This is a sensible approximation, as it shares a maximum with (2) when the uniform distribution over G is obtainable, and encourages the agent to "cover" the current support of the achieved goal distribution as broadly as possible, so that the diffusion caused by action space exploration will increase entropy. We may now form the mixture distribution $p^t_\alpha = \alpha p_{dg} + (1-\alpha)\mathcal{U}(\text{supp}(p^t_{ag}))$ and state our final "OMEGA" objective, which anneals the approximated MEGA into the original objective:

$$J_{\text{OMEGA}}(p^t_{ag}) = \mathrm{D}_{\mathrm{KL}}(p_\alpha \parallel p^t_{ag}). \tag{4}$$

The last remaining question is, how do we choose α? We would like $\alpha = 0$ when p_{ag} and p_{dg} are disconnected, and α close to 1 when p_{ag} well approximates p_{dg}. One way to achieve this, which we adopt in our experiments, is to set

$$\alpha = 1/\max(b + \mathrm{D}_{\mathrm{KL}}(p_{dg} \parallel p_{ag}), 1),$$

where $b \leq 1$. The divergence is infinite ($\alpha = 0$) when p_{ag} does not cover p_{dg} and approaches 0 ($\alpha = 1$) as p_{ag} approaches p_{dg}. Our experiments use $b = -3$, which we found sufficient to ensure $\alpha = 1$ at convergence (with $b = 1$, we may never have $\alpha = 1$, since p_{ag} is historical and biased towards the initial state distribution s_0).

3.2 Optimizing the MEGA objective

We now consider choosing behavioural goal $\hat{g} \sim p_{bg}$ in order to optimize the MEGA objective (2), as it is the critical component of (4) for early exploration in long-horizon tasks and general unsupervised goal-seeking. In supervised tasks, the OMEGA objective (4) can be approximately optimized by instead using the environment goal with probability α.

We first consider what behavioural goals we would pick if we had an oracle that could predict the conditional distribution $q(g' \mid \hat{g})$ of goals g' that would be achieved by conditioning the policy on $\hat{g}$. Then, noting that this may be too difficult to approximate in practice, we propose a minimum density heuristic that performs well empirically. The resulting Select functions are shown in Algorithm 2.

Oracle strategy. If we knew the conditional distribution $q(g' \mid \hat{g})$ of goals g' that would be achieved by conditioning behaviour on $\hat{g}$, we could compute the expected next step MEGA objective as the expected entropy of the new empirical $p_{ag \mid g'}$ after sampling g' and adding it to our buffer:

$$\begin{aligned} J_{\text{MEGA}}(p_{ag \mid g'}) &= -\mathbb{E}_{g' \sim q(g' \mid \hat{g})} H[p_{ag \mid g'}] \\ &= \sum_{g'} q(g' \vert \hat{g}) \sum_{g} p_{ag \mid g'}(g) \log p_{ag \mid g'}(g), \end{aligned}$$

To explicitly compute this objective one must compute both the new distribution and its entropy for each possible new achieved goal g'. The following result simplifies matters in the tabular case. Proofs may be found in Appendix B.

Proposition 1 (Discrete Entropy Gain). *Given buffer $\mathcal{B}$ with $\eta = \frac{1}{|\mathcal{B}|}$, maximizing expected next step entropy is equivalent to maximizing expected point-wise entropy gain $\Delta H(g')$:*

$$\begin{aligned} \hat{g}^* &= \arg\max_{\hat{g} \in \mathcal{B}} \mathbb{E}_{g' \sim q(g' \mid \hat{g})} H[p_{ag \mid g'}] \\ &= \arg\max_{\hat{g} \in \mathcal{B}} \mathbb{E}_{g' \sim q(g' \mid \hat{g})} \Delta H(g'), \end{aligned} \tag{5}$$

where $\Delta H(g') = p_{ag}(g') \log p_{ag}(g') -$
$(p_{ag}(g') + \eta) \log(p_{ag}(g') + \eta)$.

For most agents η will quickly approach 0 as they accumulate experience, so that choosing $\hat{g}$ according to (9) becomes equal (in the limit) to choosing $\hat{g}$ to maximize the directional derivative $\langle \nabla_{p_{ag}} H[p_{ag}], q(g' \mid \hat{g}) - p_{ag} \rangle$.

[*] For diverse initial state distributions, we would need to condition both p_{dg} and p_{ag} on the initial state. For non-uniform p_{dg}, we would likely want to soften the desired distribution as the marginal benefit of additional data is usually decreasing.

Algorithm 2 O/MEGA Select functions

```
function OMEGA_Select (env goal $g_{\text{ext}}$, bias $b$, $*args$):
    $\alpha \leftarrow 1/\max(b + D_{\text{KL}}(p_{dg} \parallel p_{ag}), 1)$
    if $x \sim \mathcal{U}(0,1) < \alpha$ then return $g_{\text{ext}}$
    else return MEGA_Select($*args$)

function MEGA_Select (buffer $\mathcal{B}$, num_candidates $N$):
    Sample $N$ candidates $\{g_i\}_{i=1}^{N} \sim \mathcal{B}$
    Eliminate unachievable candidates (see text)
    return $\hat{g} = \arg\min_{g_i} \hat{p}_{ag}(g_i)$                (*)
```

Proposition 2 (Discrete Entropy Gradient).

$$\begin{aligned}\lim_{\eta\to 0} \hat{g}^* &= \arg\max_{\hat{g}\in\mathcal{B}} \langle \nabla_{p_{ag}} H[p_{ag}], q(g' \mid \hat{g}) - p_{ag} \rangle \\ &= \arg\max_{\hat{g}\in\mathcal{B}} D_{\text{KL}}(q(g' \mid \hat{g}) \parallel p_{ag}) + H[q(g' \mid \hat{g})]\end{aligned} \tag{6}$$

This provides a nice intuition behind entropy gain exploration: we seek maximally diverse outcomes ($H[q(g' \mid \hat{g})]$) that are maximally different from historical experiences ($D_{\text{KL}}(q(g' \mid \hat{g}) \parallel p_{ag})$)—i.e., exploratory behavior should evenly explore under-explored regions of the state space. By choosing goals to maximize the entropy gain, an agent effectively performs constrained gradient ascent [18, 23] on the entropy objective.

Assuming the empirical p_{ag} is used to induce (abusing notation) a density p_{ag} with full support, Proposition 2 extends to the continuous case by taking the functional derivative of the differential entropy with respect to the variation $\eta(g) = q(g' \mid \hat{g})(g) - p_{ag}(g)$ (Appendix B).

Minimum density approximation. Because we do not know $q(g' \mid \hat{g})$, we must approximate it with either a learned model or an effective heuristic. The former solution is difficult, because by the time there is enough data to make an accurate prediction conditional on $\hat{g}$, $q(g' \mid \hat{g})$ will no longer represent a sparsely explored area of the goal space. While it might be possible to make accurate few- or zero-shot predictions if an agent accumulates enough data in a long-lived, continual learning setting with sufficient diversity for meta-learning [49], in our experiments we find that a simple, minimum-density approximation, which selects goals that have minimum density according to a learned density model, is at least as effective (Appendix D). We can view this approximation as a special case where the conditional $q(g' \mid \hat{g}) = \mathbb{1}[g' = \hat{g}]$, i.e. that the agent achieves the behaviour goal.

Proposition 3. *If $q(g'|\hat{g}) = \mathbb{1}[g' = \hat{g}]$, the discrete entropy gradient objective simplifies to a minimum density objective:*

$$\begin{aligned}\hat{g}^* &= \arg\max_{\hat{g}\in\mathcal{B}} - \log[p_{ag}(\hat{g})] \\ &= \arg\min_{\hat{g}\in\mathcal{B}} p_{ag}(\hat{g}).\end{aligned} \tag{7}$$

Our minimum density heuristic (Algorithm 2) fits a density model to the achieved goals in the buffer to form estimate $\hat{p}_{ag}$ of the historical achieved goal distribution p_{ag} and uses a generate and test strategy [38] that samples N candidate goals $\{g_i\}_{i=1}^{N} \sim \mathcal{B}$ from the achieved goal buffer (we use $N = 100$ in our experiments) and selects the minimum density candidate $\hat{g} = \arg\min_{g_i} \hat{p}_{ag}(g_i)$. We then adopt a Go Explore [14] style strategy, where the agent increases its action space exploration once a goal is achieved. Intuitively, this heuristic seeks out past achieved goals in sparsely explored areas, and explores around them, pushing the frontier of achieved goals forward.

It is important for $\hat{g}$ to be achievable. If it is not, then $q(g' \mid \hat{g})$ may be disconnected from $\hat{g}$, as is the case when the agent pursues unobtainable g_{ext} (Figure 2), which undermines the purpose of the minimum density heuristic. To promote achievability, our experiments make use of two different mechanisms. First, we only sample candidate goals from the past achieved goal buffer $\mathcal{B}$. Second, we eliminate candidates whose estimated value (according to the agent's goal-conditioned Q-function) falls below a dynamic cutoff, which is set according to agent's goal achievement percentage during exploration. The specifics of this cutoff mechanism may be found in Appendix C. Neither of these heuristics are core to our algorithm, and they might be be replaced with, e.g., a generative model designed to generate achievable goals [17], or a success predictor that eliminates unachievable candidates.

4 EXPERIMENTS

Having described our objectives and proposed approaches for optimizing them, we turn to evaluating our O/MEGA agents on four challenging, long-horizon environments that standard DDPG+HER agents fail to solve. We compare the performance of our algorithms with several goal selection baselines. To gain intuition on our method, we visualize qualitatively the behaviour goals selected and quantitatively the estimated entropy of the achieved goal distribution.

Environments. We consider four environments. In `PointMaze` [62], a point must navigate a 2d maze, from the bottom left corner to the top right corner. In `AntMaze` [35, 62], an ant must navigate a U-shaped hallway to reach the target. In `FetchPickAndPlace` (hard version) [47], a robotic arm must grasp a box and move it to the desired location that is at least 20cm in the air. In `FetchStack2` [36], a robotic arm must move each of the two blocks into the desired position, where one of the block rests on top of the other. In `PointMaze` and `AntMaze` goals are 2-dimensional and the agent is successful if it reaches the goal once. In `FetchPickAndPlace` and `FetchStack2`, goals are 3- and 6-dimensional, respectively, and the agent must maintain success until the end of the episode for it to count.

Baselines. We compare MEGA and OMEGA to the three density-based Select mechanisms shown in Figure 1 above: sampling from p_{dg} ("HER"), sampling from the historical p_{ag} as done approximately by RIG ("Achieved"), and sampling from a skewed historical p_{ag} that is approximately uniform on its support, as done by DISCERN and Skew-Fit ("Skewed"). We also compare against non density-based baselines as follows. `PointMaze` and `AntMaze` are the same environments used by the recent Sibling Rivalry paper [62]. Thus, our results are directly comparable to Figure 3 of their paper, which tested four algorithms: HER, PPO [56], PPO with intrinsic curiosity [43], and PPO with Sibling Rivalry (PPO+SR). The `AntMaze` environment uses the same simulation as the `MazeAnt` environment tested in the Goal GAN paper [17], but is four times larger. In Appendix D, we test MEGA on the smaller maze and obtain an almost

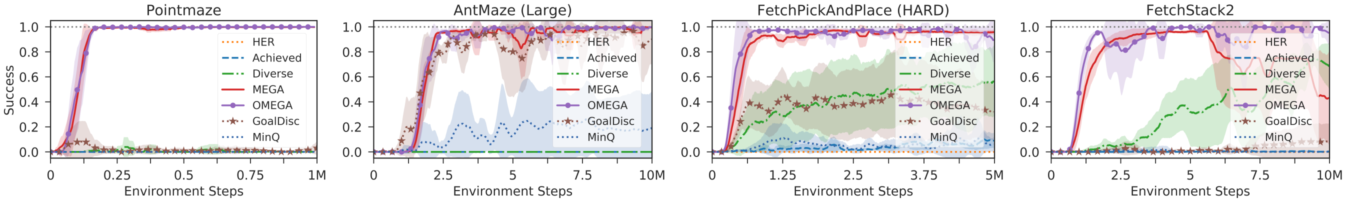

Fig. 3: Test success on the desired goal distribution, evaluated throughout training, for several behaviour goal selection methods (3 seeds each). Our methods (MEGA and OMEGA) are the only the methods which are able to solve the tasks with highest sample efficiency. In FetchStack2 we see that OMEGA's eventual focus on the desired goal distribution is necessary for long run stability.

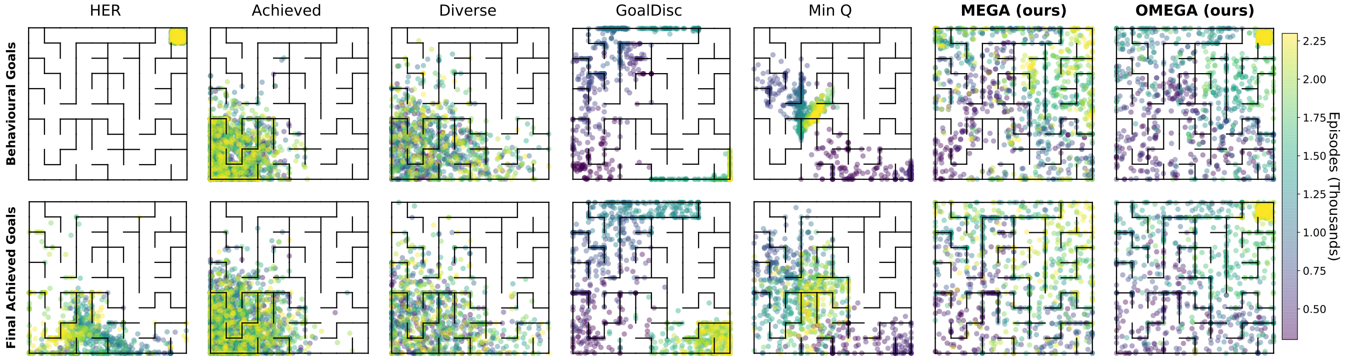

Fig. 4: Visualization of behavioural (top) and terminal achieved (bottom) goals in PointMaze, colour-coded for over the course of training for several behavioural goal sampling methods. Only our methods reach the desired goal area in top right hand corner in approximately 2000 episodes, beating the previous state of the art [62] by almost 2 orders of magnitude (100 times).

1000x increase in sample efficiency as compared to Goal GAN and the Goal GAN implementation of SAGG-RIAC [4]. Results are not directly comparable as Goal GAN uses an on-policy TRPO base [55], which is very sample inefficient relative to our off-policy DDPG base. Thus, we adapt the Goal GAN discriminator to our setting by training a success predictor to identify goals of intermediate difficulty (Appendix C) ("GoalDisc"). Finally, we compare against a minimum Q heuristic, which selects distant goals [22] ("MinQ").

We note a few things before moving on. First, Sibling Rivalry [62] is the only prior work that directly addresses the long-horizon, sparse reward problem (without imitation learning). Other baselines were motivated by and tested on other problems. Second, the generative parts of Goal GAN and RIG are orthogonal to our work, and could be combined with MEGA-style generate-and-test selection, as we noted above in Section 3.2. We adopted the generative mechanism of DISCERN (sampling from a buffer) as it is simple and has a built-in bias toward achievable samples. For a fair comparison, all of our implemented baselines use the same buffer-based generative model and benefit from our base DDPG+HER implementation (Appendix C). The key difference between MEGA and our implemented baselines is the Select mechanism (line (∗) of Algorithm 2).

Main Results. Our main results, shown in Figure 3 clearly demonstrate the advantage of minimum density sampling. We confirm that desired goal sampling (HER) is unable to solve the tasks, and observe that Achieved and Diverse goal sampling fail to place enough focus on the frontier of the achieved goal distribution to bridge the gap between the initial state and desired goal distributions. On PointMaze, none of the baselines were able to solve the environment within 1 million steps. The best performing algorithm from Trott et al. [62] is PPO+SR, which solves PointMaze to 90% success in approximately 7.5 million time steps (O/MEGA is almost 100 times faster). On AntMaze, only MEGA, OMEGA and the GoalDisc are able to solve the environment. The best performing algorithm from Trott et al. [62] is hierarchical PPO+SR, which solves AntMaze to 90% success in approximately 25 million time steps (O/MEGA is roughly 10 times faster). On a maze that is four times smaller, Florensa et al. [17] tested four algorithms, including SAGG-RIAC [4], which was implemented, along with Goal GAN, using a TRPO base. Their best performing result achieves 71% coverage of the maze in about 175 million time steps (O/MEGA is roughly 100 times faster on a larger maze). O/MEGA also demonstrates that FetchStack2 can be solved from scratch, without expert demonstrations [13, 36] or a task curriculum [9].

Maximizing Entropy. In Figure 5 (top), we observe that our approach increases the empirical entropy of the achieved goal buffer (the MEGA objective) much faster than other goal sampling methods. MEGA and OMEGA rapidly increase the entropy and begin to succeed with respect to the desired goals as the maximum entropy is reached. As OMEGA begins to shift towards sampling mainly from the desired goal distribution (Figure 5 (bottom)), the entropy declines as desired goal trajectories become over represented. We observe that the intermediate difficulty heuristic is a good optimizer of the MEGA objective on AntMaze, likely due to the linear structure of the environment. This explains its comparable performance to MEGA.

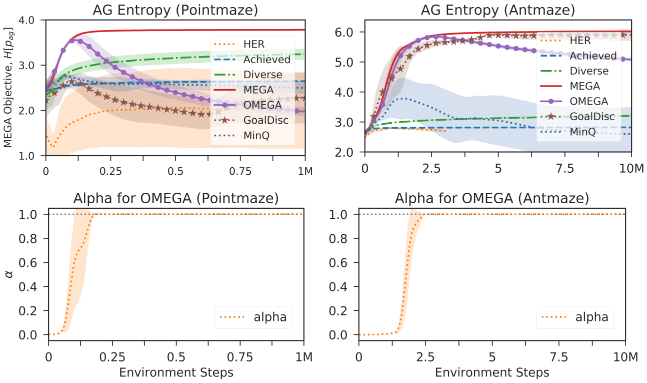

Fig. 5: Top: Entropy of the achieved goal buffer for `Pointmaze` (left) and `Antmaze` (right) over course of training, estimated using a Kernel Density Estimator. O/MEGA expand the entropy much faster than the baselines. Bottom: α computed by OMEGA, which transitions from intrinsic to extrinsic goals.

Visualization of Achieved Goals. To gain intuition for how our method compares to the baselines, we visualize the terminal achieved goal at the end of the episodes throughout the training for `PointMaze` in Figure 4. A corresponding figure for `AntMaze` can be found in Appendix E. Both MEGA and OMEGA set goals that spread outward from the starting location as training progresses, akin to a breadth-first search, with OMEGA eventually transitioning to goals from the desired goal distribution in the top right corner. Diverse sampling maintains a fairly uniform distribution at each iteration, but explores slowly as most goals are sampled from the interior of the support instead of the frontier. Achieved sampling oversamples goals near the starting location and suffers from a "rich get richer" problem. Difficulty-based GoalDisc and distance-based MinQ sampling explore deeply in certain directions, akin to a depth-first search, but ignore easier/closer goals that can uncover new paths.

5 OTHER RELATED WORK

Curiosity. Maximizing entropy in the goal space is closely related to general RL (not multi-goal) algorithms that seek to maximize entropy in the state space [23, 30] or grant the agent additional reward based on some measure of novelty, surprise or learning progress [5, 8, 29, 32, 41, 43, 54, 61]. A key difference is that our work learns and uses a goal-conditioned policy for exploration, rather than training a monolithic policy to optimize an exploration objective. In this sense, our work is similar to noise-conditioned [40, 46] and variational exploration algorithms (next paragraph). Future work might explore how one can automatically choose a good goal space for doing MEGA-style maximum entropy exploration.

Empowerment. Since the agent's off-policy, goal relabeling learning algorithm can be understood as minimizing the conditional entropy of (on-policy) achieved goals given some potential goal distribution p_g (not necessarily the behavioural goal distribution p_{bg}), simultaneously choosing p_{bg} to maximize entropy of historical achieved goals (the MEGA objective) results in an *empowerment*-like total objective: $\max_{p_{bg}} H[p_{ag}] - H[\mathrm{AG}(\tau \mid p_g) \mid p_g] \approx \max_{p_g} I[p_g; \mathrm{AG}(\tau \mid p_g)]$, where equality is approximate because the first max is with respect to p_{bg}, and also because $H[p_{ag}]$ is historical, rather than on-policy. Empowerment [28, 34, 51] has gained traction in recent years as an intrinsic, unsupervised objective due to its intuitive interpretation and empirical success [15, 21]. We can precisely define empowerment in the multi-goal case as the *channel capacity* between goals and achieved goals [10]:

$$\mathcal{E}(s_0) = \max_{p_g} \mathbb{E}_{p(\tau|g, s_0) p_g(g)} I[p_g; \mathrm{AG}(\tau \mid p_g)], \tag{8}$$

where s_0 represents the initial state distribution. To see the intuitive appeal of this objective, we reiterate the common argument and write: $I[p_g; \mathrm{AG}(\tau \mid p_g)] = H[p_g] - H[p_g \mid \mathrm{AG}(\tau \mid p_g)]$, where H is entropy. This now has an intuitive interpretation: letting $H[p_g]$ stand for the size of the goal set, and $H[p_g \mid \mathrm{AG}(\tau \mid p_g)]$ for the uncertainty of goal achievement, maximizing empowerment roughly amounts to maximizing the *size of the achievable goal set.*

The common approach to maximizing empowerment has been to either fix or parameterize the distribution p_g and maximize the objective $I[p_g; \mathrm{AG}(\tau \mid p_g)]$ *on-policy* [20, 48, 64]. We can think of this as approximating (8) using the behavioural goal distribution $p_{bg} \approx \arg\max_{p_g} I[p_g; \mathrm{AG}(s_T \mid p_g)]$. A key insight behind our work is that there is no reason for an off-policy agent to constrain itself to pursuing goals from the distribution it is trying to optimize. Instead, we argue that for off-policy agents seeking to optimize (8), the role of the behavioural goal distribution p_{bg} should be to produce useful empirical data for optimizing the true *off-policy* empowerment (8), where the maximum is taken over all possible p_g. Practically speaking, this means exploring to maximize entropy of the historical achieved goal distribution (i.e,. the MEGA objective), and letting our off-policy, goal relabeling algorithm minimize the conditional entropy term. Future work should investigate whether the off-policy gain of MEGA over the on-policy Diverse sampling can be transferred to general empowerment maximizing algorithms.

6 CONCLUSION

This paper proposes to address the long-horizon, sparse reward problem in multi-goal RL by having the agent maximize the entropy of the historical achieved goal distribution. We do this by setting intrinsic goals in sparsely explored areas of the state space, which focuses exploration on the frontier of the achieveable goal set. This strategy obtains results that are more than 10 times more sample efficient than prior approaches in four long-horizon multi-goal tasks.

We also identified two directions for future work. First, how can an agent automatically discover a good low-dimensional goal space for maximum entropy gain exploration? Second, can the idea of maximizing empowerment "off-policy" be extended to improve other empowerment maximizing algorithms? Other angles include combining MEGA exploration with hierarchical RL algorithms [35], applying MEGA in pixel-based tasks [37] and using MEGA exploration to optimize general (not multi-goal) tasks [21, 64].

LINK TO FULL APPENDIX

`https://www.dropbox.com/s/h5eliwabwf0jbwr/MEGA_Appendix.pdf`
Code to follow.

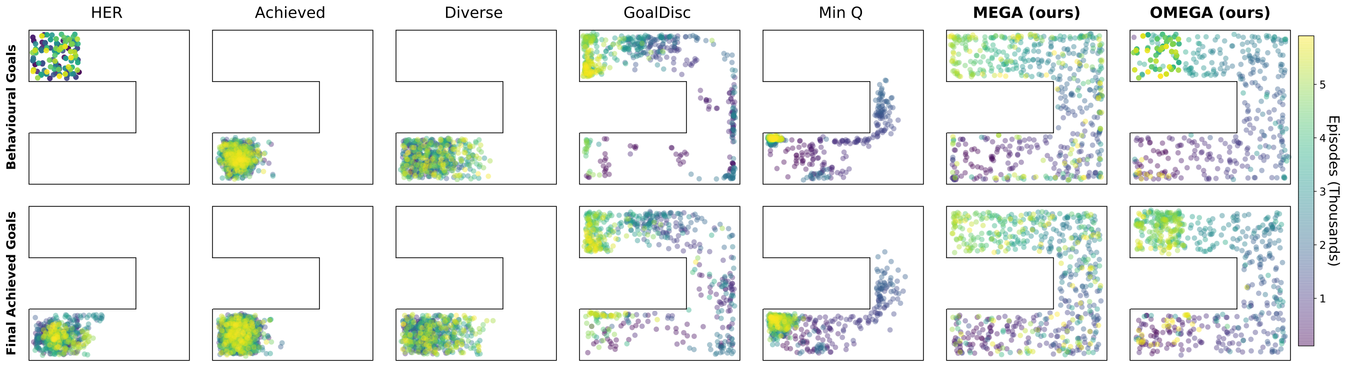

Fig. 6: Additional visualization: behavioural (top) and terminal achieved (bottom) goals in AntMaze.

ABRIDGED APPENDIX

Implementation details. We use a DDPG agent that acts in multiple parallel environments. Our agent keeps a single replay buffer and copy of its parameters and training is parallelized using a GPU. We utilize many of the same tricks as Plappert et al. [47], including clipping raw observations to [-200, 200], normalizing clipped observations, clipping normalized observations to [-5, 5], and clipping the Bellman targets to $[-\frac{1}{1-\gamma}, 0]$. Our agent uses independently parameterized, layer-normalized [3] actor and critic, each with 3 layers of 512 neurons with GeLU activations [24]. We apply gradient norm clipping of 5, and apply action l2 regularization with coefficient 1e-1 to the unscaled output of the actor. We apply action noise of 0.1 to the actor at all exploration steps, and also apply epsilon random exploration of 0.1.

We use Adam Optimizer [26] with a learning rate of 1e-3 for both actor and critic, and update the target networks every 40 training steps with a Polyak averaging coefficient of 0.05. We vary the frequency of training depending on the environment, which can stabilize training; we optimize every step in PointMaze, every two steps in Antmaze, ever four steps in FetchPickAndPlace, and every ten steps in FetchStack2. Optimization steps use a batch size of 2000, which is sampled uniformly from the buffer (no priorization). There is an initial "policy warm-up" period of 5000 steps, during which the agent acts randomly. Our buffer has infinite length.

We generalize the future strategy by additionally relabeling transitions with goals randomly sampled (uniformly) from buffers of actual (environment) goals, past achieved goals, and behavioral goals (i.e., goals that agent pursues during training). We call this the rfaab strategy, which stands for Real (do not relabel), Future, Actual, Achieved, and Behavioral. Intuitively, relabeling transitions with goals outside the current trajectory allows the agent to generalize across trajectories. All relabeling is done online. The rfaab strategy requires, as hyperparameters, relative ratios of each kind of experience, as it will appear in the minibatch. Thus, rfaab_1_4_3_1_1 keeps 10% real experiences, and relabels approximately 40% with future, 30% with actual, 10% with achieved, and 10% with behavioral. We use rfaab_1_4_3_1_1 in PointMaze and Antmaze and rfaab_1_5_2_1_1 in Fetch.

For density modeling, we considered three approaches: a kernel density estimator (KDE) [50], a normalizing flow (Flow) [42] based on RealNVP [12], and a random network distillation (RND) approach [8]. Based on the resulting performances and relatively complexity, we chose to use KDE throughout our experiments. We tested each approach in PointMaze only. Both the KDE and Flow models obtain similar performance, whereas the RND model makes very slow progress. Between KDE and Flow, we opted to use KDE throughout our experiments as it is fast, easy to implement, and equally effective in the chosen goal spaces (maximum 6 dimensions). It is possible that a Flow (or VAE-like model [37]) would be necessary in a higher dimensional space.

To encourage the agent to explore around the behavioral goal, we increase the agent's exploratory behaviors every time the behavioral goal is reachieved in any given episode. We refer to this as "Go Exploration" after Ecoffet et al. [14], who used a similar approach to reset the environment to a frontier state, and explored around that state. We use a very simple exploration bonus, which increases the agent's epsilon exploration by a fixed percentage. We use 10% (see next paragraph), which means that an agent which achieved the goal 10 times in an episode will be exploring purely at random. All baselines benefited from this feature.

Many of the parameters above were initially based on what has worked in the past Dhariwal et al. [11], Plappert et al. [47]. To finetune the base hyperparameters, we ran two random searches on PointMaze—one for rfaab and one for general hyperparameters—in order to tune a MEGA agent. The same base hyperparameters were used for all baselines.

All baselines are modify only line (∗) of MEGA_Select. The Diverse baseline scores candidates using $1/\hat{p}_{ag}$, where $\hat{p}_{ag}$ is estimated by the density model (KDE, see above), and then samples randomly from the candidates in proportion to their scores. This is similar to using Skew-Fit with $\alpha = -1$ [48] or using DISCERN's diverse strategy [64]. The Achieved baseline samples a random candidate uniformly. This is similar to RIG [37] and to DISCERN's naive strategy [64]. This GoalDisc baseline adapts Florensa et al. [17]'s GoalGAN. To select goals, it passes the goal candidates, along with starting states, to a trained goal discriminator, which predicts the likelihood that each candidate will be achieved from the starting state. The goals are ranked based on how close the output of the discriminator is to 0.5, choosing the goal with the minimum absolute value distance (the "most intermediate difficulty" goal). We do not use the cutoff mechanism based on Q-values in this strategy. The MinQ strategy uselects the goal with the lowest Q-value [22].

REFERENCES

[1] Abels, A., Roijers, D. M., Lenaerts, T., Now'e, A., and Steckelmacher, D. Dynamic weights in multi-objective deep reinforcement learning. *arXiv preprint arXiv:1809.07803*, 2018.

[2] Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, O. P., and Zaremba, W. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, pp. 5048–5058, 2017.

[3] Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] Baranes, A. and Oudeyer, P.-Y. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1): 49–73, 2013.

[5] Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. In *Advances in neural information processing systems*, pp. 1471–1479, 2016.

[6] Bishop, C. M. *Pattern recognition and machine learning*. springer, 2006.

[7] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016.

[8] Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

[9] Colas, C., Sigaud, O., and Oudeyer, P.-Y. Curious: Intrinsically motivated multi-task, multi-goal reinforcement learning. *arXiv preprint arXiv:1810.06284*, 2018.

[10] Cover, T. M. and Thomas, J. A. *Elements of information theory*. John Wiley & Sons, 2012.

[11] Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y., and Zhokhov, P. Openai baselines. https://github.com/openai/baselines, 2017.

[12] Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[13] Duan, Y., Andrychowicz, M., Stadie, B., Ho, O. J., Schneider, J., Sutskever, I., Abbeel, P., and Zaremba, W. One-shot imitation learning. In *Advances in neural information processing systems*, pp. 1087–1098, 2017.

[14] Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K. O., and Clune, J. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019.

[15] Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[16] Fang, M., Zhou, T., Du, Y., Han, L., and Zhang, Z. Curriculum-guided hindsight experience replay. In *Advances in Neural Information Processing Systems*, pp. 12602–12613, 2019.

[17] Florensa, C., Held, D., Geng, X., and Abbeel, P. Automatic goal generation for reinforcement learning agents. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1515–1528, 2018.

[18] Frank, M. and Wolfe, P. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[19] Fujimoto, S., Van Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.

[20] Gregor, K., Rezende, D. J., and Wierstra, D. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

[21] Hansen, S., Dabney, W., Barreto, A., Van de Wiele, T., Warde-Farley, D., and Mnih, V. Fast task inference with variational intrinsic successor features. In *International Conference on Learning Representations*, 2020.

[22] Hartikainen, K., Geng, X., Haarnoja, T., and Levine, S. Dynamical distance learning for semi-supervised and unsupervised skill discovery. In *International Conference on Learning Representations*, 2020.

[23] Hazan, E., Kakade, S. M., Singh, K., and Van Soest, A. Provably efficient maximum entropy exploration. *arXiv preprint arXiv:1812.02690*, 2018.

[24] Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[25] Kaelbling, L. P. Learning to achieve goals. In *IJCAI*, pp. 1094–1099. Citeseer, 1993.

[26] Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[27] Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[28] Klyubin, A. S., Polani, D., and Nehaniv, C. L. Empowerment: A universal agent-centric measure of control. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pp. 128–135. IEEE, 2005.

[29] Kolter, J. Z. and Ng, A. Y. Near-bayesian exploration in polynomial time. In *Proceedings of the 26th annual international conference on machine learning*, pp. 513–520, 2009.

[30] Lee, L., Eysenbach, B., Parisotto, E., Xing, E., Levine, S., and Salakhutdinov, R. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

[31] 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*, 2015.

[32] Lopes, M., Lang, T., Toussaint, M., and Oudeyer, P.-Y. Exploration in model-based reinforcement learning by empirically estimating learning progress. In *Advances in neural information processing systems*, pp. 206–214, 2012.

[33] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[34] Mohamed, S. and Rezende, D. J. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pp. 2125–2133, 2015.

[35] Nachum, O., Gu, S. S., Lee, H., and Levine, S. Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 3303–3313, 2018.

[36] Nair, A., McGrew, B., Andrychowicz, M., Zaremba, W., and Abbeel, P. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6292–6299. IEEE, 2018.

[37] Nair, A. V., Pong, V., Dalal, M., Bahl, S., Lin, S., and Levine, S. Visual reinforcement learning with imagined goals. In *Advances in Neural Information Processing Systems*, pp. 9209–9220, 2018.

[38] Newell, A. *Artificial intelligence and the concept of mind.* Carnegie-Mellon University, Department of Computer Science, 1969.

[39] Osband, I., Van Roy, B., and Wen, Z. Generalization and exploration via randomized value functions. *arXiv preprint arXiv:1402.0635*, 2014.

[40] Osband, I., Russo, D., Wen, Z., and Van Roy, B. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 2017.

[41] Ostrovski, G., Bellemare, M. G., van den Oord, A., and Munos, R. Count-based exploration with neural density models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2721–2730. JMLR. org, 2017.

[42] Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019.

[43] Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017.

[44] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[45] Pitis, S., Chan, H., and Ba, J. Protoge: Prototype goal encodings for multi-goal reinforcement learning. *The 4th Multidisciplinary Conference on Reinforcement Learning and Decision Making*, 2019.

[46] Plappert, M., Houthooft, R., Dhariwal, P., Sidor, S., Chen, R. Y., Chen, X., Asfour, T., Abbeel, P., and Andrychowicz, M. Parameter space noise for exploration. *arXiv preprint arXiv:1706.01905*, 2017.

[47] Plappert, M., Andrychowicz, M., Ray, A., McGrew, B., Baker, B., Powell, G., Schneider, J., Tobin, J., Chociej, M., Welinder, P., et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

[48] Pong, V. H., Dalal, M., Lin, S., Nair, A., Bahl, S., and Levine, S. Skew-Fit: State-Covering Self-Supervised Reinforcement Learning. *arXiv e-prints*, March 2019.

[49] Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H., and Zemel, R. S. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

[50] Rosenblatt, M. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, pp. 832–837, 1956.

[51] Salge, C., Glackin, C., and Polani, D. Empowerment–an introduction. In *Guided Self-Organization: Inception*, pp. 67–114. Springer, 2014.

[52] Schaul, T., Horgan, D., Gregor, K., and Silver, D. Universal value function approximators. In *International Conference on Machine Learning*, pp. 1312–1320, 2015.

[53] Schaul, T., Quan, J., Antonoglou, I., and Silver, D. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[54] Schmidhuber, J. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

[55] Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897, 2015.

[56] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[57] Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. In *ICML*, 2014.

[58] Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

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

[60] Sutton, R. S., Modayil, J., Delp, M., Degris, T., Pilarski, P. M., White, A., and Precup, D. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on*

Autonomous Agents and Multiagent Systems-Volume 2, pp. 761–768, 2011.

[61] Tang, H., Houthooft, R., Foote, D., Stooke, A., Chen, O. X., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in neural information processing systems*, pp. 2753–2762, 2017.

[62] Trott, A., Zheng, S., Xiong, C., and Socher, R. Keeping your distance: Solving sparse reward tasks using self-balancing shaped rewards. In *Advances in Neural Information Processing Systems 32*, pp. 10376–10386. 2019.

[63] Van Seijen, H., Fatemi, M., and Tavakoli, A. Using a logarithmic mapping to enable lower discount factors in reinforcement learning. In *Advances in Neural Information Processing Systems 32*, pp. 14111–14121. 2019.

[64] Warde-Farley, D., de Wiele, T. V., Kulkarni, T., Ionescu, C., Hansen, S., and Mnih, V. Unsupervised control through non-parametric discriminative rewards. In *International Conference on Learning Representations*, 2019.

[65] Watkins, C. J. and Dayan, P. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.