Task-Agnostic Exploration via Policy Gradient of a Non-Parametric State Entropy Estimate

Mirco Mutti[1,2,*], **Lorenzo Pratissoli**[1,*], and **Marcello Restelli**[1]
[1] Politecnico di Milano, Milan, Italy
[2] Università di Bologna, Bologna, Italy
mirco.mutti@polimi.it, lorenzo.pratissoli@mail.polimi.it, marcello.restelli@polimi.it

Abstract

In a reward-free environment, what is a suitable intrinsic objective for an agent to pursue so that it can learn an optimal task-agnostic exploration policy? In this paper, we argue that the entropy of the state distribution induced by finite-horizon trajectories is a sensible target. Especially, we present a novel and practical policy-search algorithm, Maximum Entropy POLicy optimization (MEPOL), to learn a policy that maximizes a non-parametric, k-nearest neighbors estimate of the state distribution entropy. In contrast to known methods, MEPOL is completely model-free as it requires neither to estimate the state distribution of any policy nor to model transition dynamics. Then, we empirically show that MEPOL allows learning a maximum-entropy exploration policy in high-dimensional, continuous-control domains, and how this policy facilitates learning meaningful reward-based tasks downstream.

1 Introduction

In recent years, Reinforcement Learning (RL) (Sutton and Barto 2018) has achieved outstanding results in remarkable tasks, such as Atari games (Mnih et al. 2015), Go (Silver et al. 2016), Dota 2 (Berner et al. 2019), and dexterous manipulation (Andrychowicz et al. 2020). To accomplish these feats, the learning process usually requires a considerable amount of human supervision, especially a hand-crafted reward function (Hadfield-Menell et al. 2017), while the outcome rarely generalizes beyond a single task (Cobbe et al. 2019). This barely mirrors human-like learning, which is far less dependent on exogenous guidance and exceptionally general. Notably, an infant would go through an intrinsically-driven, nearly exhaustive, exploration of the environment in an early stage, without knowing much about the tasks she/he will face. Still, this same unsupervised process will be consequential to solve those complex, externally-driven tasks, when they will eventually arise. In this perspective, what is a suitable *task-agnostic* exploration objective to set for the agent in an *unsupervised* phase, so that the acquired knowledge would facilitate learning a variety of reward-based tasks afterwards?

Lately, several works have addressed this question in different directions. In (Bechtle et al. 2019; Zheng et al. 2020), authors investigate how to embed task-agnostic knowledge into a transferable meta-reward function. Other works (Jin et al. 2020; Tarbouriech et al. 2020) consider the active estimation of the environment dynamics as an unsupervised objective. Another promising approach, which is the one we focus in this paper, is to incorporate the unsupervised knowledge into a *task-agnostic exploration policy*, obtained by maximizing some entropic measure over the state space (Hazan et al. 2019; Tarbouriech and Lazaric 2019; Mutti and Restelli 2020; Lee et al. 2019). Intuitively, an exploration policy might be easier to transfer than a transition model, which would be hardly robust to changes in the environment, and more ready to use than a meta-reward function, which would still require optimizing a policy as an intermediate step. An ideal maximum-entropy policy, thus inducing a uniform distribution over states, is an extremely general starting point to solve any (unknown) subsequent goal-reaching task, as it minimizes the so-called worst-case regret (Gupta et al. 2018, Lemma 1). In addition, by providing an efficient estimation of any, possibly sparse, reward function, it significantly reduces the burden on reward design. In tabular settings, Tarbouriech and Lazaric (2019); Mutti and Restelli (2020) propose theoretically-grounded methods for learning an exploration policy that maximizes the entropy of the asymptotic state distribution, while Mutti and Restelli (2020) concurrently consider the minimization of the mixing time as a secondary objective. In (Hazan et al. 2019), authors present a principled algorithm (MaxEnt) to optimize the entropy of the discounted state distribution of a tabular policy, and a theoretically-relaxed implementation to deal with function approximation. Similarly, Lee et al. (2019) design a method (SMM) to maximize the entropy of the finite-horizon state distribution. Both SMM and MaxEnt learn a maximum-entropy mixture of policies following this iterative procedure: first, they estimate the state distribution induced by the current mixture to define an intrinsic reward, then, they learn a policy that optimizes this reward to be added to the mixture. Unfortunately, the literature approaches to state entropy maximization either consider impractical infinite-horizon settings (Tarbouriech and Lazaric 2019; Mutti and Restelli 2020), or output a mixture of policies that would be inadequate for non-episodic tasks (Hazan et al. 2019; Lee et al. 2019). In addition, they would still require a full model of the transition dynamics (Tarbouriech and Lazaric 2019; Mutti and Restelli 2020), or a state density estimation (Hazan et al. 2019; Lee

[*]Equal contribution.

et al. 2019), which hardly scale to complex domains.

In this paper, we present a novel policy-search algorithm (Deisenroth et al. 2013), to deal with task-agnostic exploration via state entropy maximization over a *finite horizon*, which gracefully scales to continuous, high-dimensional domains. The algorithm, which we call *Maximum Entropy POLicy optimization* (MEPOL), allows learning a *single* maximum-entropy parameterized policy from mere interactions with the environment, combining non-parametric state entropy estimation and function approximation. It is completely *model-free* as it requires neither to model the environment transition dynamics nor to directly estimate the state distribution of any policy. The entropy of continuous distributions can be speculated by looking at how random samples drawn from them laid out over the support surface (Beirlant et al. 1997). Intuitively, samples from a high entropy distribution would evenly cover the surface, while samples drawn from low entropy distributions would concentrate over narrow regions. Backed by this intuition, MEPOL relies on a k-nearest neighbors entropy estimator (Singh et al. 2003) to asses the quality of a given policy from a batch of interactions. Hence, it searches for a policy that maximizes this entropy index within a parametric policy space. To do so, it combines ideas from two successful, state-of-the-art policy-search methods: TRPO (Schulman et al. 2015), as it performs iterative optimizations of the entropy index within trust regions around the current policies, and POIS (Metelli et al. 2018), as these optimizations are performed offline via importance sampling. This recipe allows MEPOL to learn a maximum-entropy task-agnostic exploration policy while showing stable behavior during optimization.

The paper is organized as follows. First, we report the basic background (Section 2) and some relevant theoretical properties (Section 3) that will be instrumental to subsequent sections. Then, we present the task-agnostic exploration objective (Section 4), and a learning algorithm, MEPOL, to optimize it (Section 5), which is empirically evaluated in Section 6. In Appendix A,[1] we discuss related work. The proofs of the theorems are reported in Appendix B. The implementation of MEPOL can be found at https://github.com/muttimirco/mepol.

2 Preliminaries

In this section, we report background and notation.

Markov Decision Processes A discrete-time Markov Decision Process (MDP) (Puterman 2014) is defined by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, d_0)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state space and the action space respectively, $P(s'|s,a)$ is a Markovian transition model that defines the conditional probability of the next state s' given the current state s and action a, $R(s)$ is the expected immediate reward when arriving in state s, and d_0 is the initial state distribution. A trajectory $\tau \in \mathcal{T}$ is a sequence of state-action pairs $\tau = (s_0, a_0, s_1, a_1, \ldots)$. A policy $\pi(a|s)$ defines the probability of taking action a given the current state s. We denote by Π the set of all stationary Markovian policies. A policy π that interacts with an MDP, induces a t-step state distribution defined as (let $d_0^\pi = d_0$):

$$d_t^\pi(s) = Pr(s_t = s|\pi) = \int_{\mathcal{T}} Pr(\tau|\pi, s_t = s)\,\mathrm{d}\tau,$$

$$d_t^\pi(s) = \int_{\mathcal{S}} d_{t-1}^\pi(s') \int_{\mathcal{A}} \pi(a|s') P(s|s',a)\,\mathrm{d}a\,\mathrm{d}s',$$

for every $t > 0$. If the MDP is ergodic, it admits a unique steady-state distribution which is $\lim_{t\to\infty} d_t^\pi(s) = d^\pi(s)$. The mixing time t_{mix} describes how fast the state distribution d_t^π converges to its limit, given a mixing threshold ϵ:

$$t_{\text{mix}} = \big\{t \in \mathbb{N} : \sup_{s\in\mathcal{S}} \big|d_t^\pi(s) - d_{t-1}^\pi(s)\big| \leq \epsilon\big\}.$$

Differential Entropy Let $f(x)$ be a probability density function of a random vector $\boldsymbol{X}$ taking values in $\mathbb{R}^p$, then its differential entropy (Shannon 1948) is defined as:

$$H(f) = -\int f(x) \ln f(x)\,\mathrm{d}x.$$

When the distribution f is not available, this quantity can be estimated given a realization of $\boldsymbol{X} = \{x_i\}_{i=1}^N$ (Beirlant et al. 1997). In particular, to deal with high-dimensional data, we can turn to non-parametric, k-Nearest Neighbors (k-NN) entropy estimators of the form (Singh et al. 2003):

$$\widehat{H}_k(f) = -\frac{1}{N}\sum_{i=1}^N \ln \frac{k}{N V_i^k} + \ln k - \Psi(k), \qquad (1)$$

where Ψ is the digamma function, $\ln k - \Psi(k)$ is a bias correction term, V_i^k is the volume of the hyper-sphere of radius $R_i = |x_i - x_i^{k\text{-NN}}|$, which is the Euclidean distance between x_i an its k-nearest neighbor $x_i^{k\text{-NN}}$, so that:

$$V_i^k = \frac{\big|x_i - x_i^{k\text{-NN}}\big|^p \cdot \boldsymbol{\pi}^{p/2}}{\Gamma(\frac{p}{2}+1)},$$

where Γ is the gamma function, and p the dimensions of $\boldsymbol{X}$. The estimator (1) is known to be asymptotically unbiased and consistent (Singh et al. 2003). When the target distribution f' differs from the sampling distribution f, we can provide an estimate of $H(f')$ by means of an Importance-Weighted (IW) k-NN estimator (Ajgl and Šimandl 2011):

$$\widehat{H}_k(f'|f) = -\sum_{i=1}^N \frac{W_i}{k} \ln \frac{W_i}{V_i^k} + \ln k - \Psi(k), \qquad (2)$$

where $W_i = \sum_{j\in\mathcal{N}_i^k} w_j$, such that $\mathcal{N}_i^k$ is the set of indices of the k-NN of x_i, and w_j are the normalized importance weights of samples x_j, which are defined as:

$$w_j = \frac{{}^{f'(x_j)}\!/_{f(x_j)}}{\sum_{n=1}^N {}^{f'(x_n)}\!/_{f(x_n)}}.$$

As a by-product, we have access to a non-parametric IW k-NN estimate of the Kullback-Leibler (KL) divergence, given by (Ajgl and Šimandl 2011):

$$\widehat{D}_{KL}\big(f||f'\big) = \frac{1}{N}\sum_{i=1}^N \ln \frac{{}^k\!/_N}{\sum_{j\in\mathcal{N}_i^k} w_j}. \qquad (3)$$

Note that, when $f' = f$, $w_j = {}^1\!/_N$, the estimator (2) is equivalent to (1), while $\widehat{D}_{KL}(f||f')$ is zero.

[1] A complete version of the paper, which includes the Appendix, is available at https://arxiv.org/abs/2007.04640

3 Analysis of the Importance-Weighted Entropy Estimator

In this section, we present a theoretical analysis over the quality of the estimation provided by (2). Especially, we provide a novel detailed proof of the bias, and a new characterization of its variance. Similarly as in (Singh et al. 2003, Theorem 8) for the estimator (1), we can prove the following.

Theorem 3.1. (Ajgl and Šimandl 2011, Sec. 4.1) *Let f be a sampling distribution, f' a target distribution. The estimator $\widehat{H}_k(f'|f)$ is asymptotically unbiased for any choice of k.*

Therefore, given a sufficiently large batch of samples from an unknown distribution f, we can get an unbiased estimate of the entropy of any distribution f', irrespective of the form of f and f'. However, if the distance between the two grows large, a high variance might negatively affect the estimation.

Theorem 3.2. *Let f be a sampling distribution, f' a target distribution. The asymptotic variance of the estimator $\widehat{H}_k(f'|f)$ is given by:*

$$\lim_{N\to\infty} \operatorname*{Var}_{x\sim f}\left[\widehat{H}_k(f'|f)\right] = \frac{1}{N}\Bigg(\operatorname*{Var}_{x\sim f}\left[\overline{w}\ln\overline{w}\right] + \operatorname*{Var}_{x\sim f}\left[\overline{w}\ln R^p\right] + \left(\ln C\right)^2 \operatorname*{Var}_{x\sim f}\left[\overline{w}\right]\Bigg),$$

where $\overline{w} = \frac{f'(x)}{f(x)}$, and $C = \frac{N\pi^{p/2}}{k\Gamma(p/2+1)}$ is a constant.

4 A Task-Agnostic Exploration Objective

In this section, we define a learning objective for task-agnostic exploration, which is a fully unsupervised phase that potentially precedes a set of diverse goal-based RL phases. First, we make a common regularity assumption on the class of the considered MDPs, which allows us to exclude the presence of unsafe behaviors or dangerous states.

Assumption 4.1. *For any policy $\pi \in \Pi$, the corresponding Markov chain P^π is ergodic.*

Then, following a common thread in maximum-entropy exploration (Hazan et al. 2019; Tarbouriech and Lazaric 2019; Mutti and Restelli 2020), and particularly (Lee et al. 2019), which focuses on a finite-horizon setting as we do, we define the *task-agnostic exploration* problem:

$$\underset{\pi\in\Pi}{\text{maximize}} \;\; \mathcal{F}_{\text{TAE}}(\pi) = H\left(\frac{1}{T}\sum_{t=1}^{T} d_t^\pi\right), \tag{4}$$

where $\bar{d}_T = \frac{1}{T}\sum_{t=1}^{T} d_t^\pi$ is the *average state distribution*. An optimal policy w.r.t. this objective favors a maximal coverage of the state space into the finite-horizon T, irrespective of the state-visitation order. Notably, the exploration horizon T has not to be intended as a given trajectory length, but rather as a parameter of the unsupervised exploration phase which allows to tradeoff exploration quality (i.e., state-space coverage) with exploration efficiency (i.e., mixing properties).

As the thoughtful reader might realize, optimizing Objective (4) is not an easy task. Known approaches would require either to estimate the transition model in order to obtain average state distributions (Tarbouriech and Lazaric 2019; Mutti and Restelli 2020), or to directly estimate these distributions through a density model (Hazan et al. 2019; Lee et al. 2019). In contrast to the literature, we turn to non-parametric entropy estimation without explicit state distributions modeling, deriving a more practical policy-search approach that we present in the following section.

Algorithm 1 MEPOL

Input: exploration horizon T, sample-size N, trust-region threshold δ, learning rate α, nearest neighbors k
initialize $\boldsymbol{\theta}$
for epoch = $1, 2, \ldots,$ until convergence **do**
  draw a batch of $\lceil \frac{N}{T} \rceil$ trajectories of length T with $\pi_{\boldsymbol{\theta}}$
  build a dataset of particles $\mathcal{D}_\tau = \{(\tau_i^t, s_i)\}_{i=1}^N$
  $\boldsymbol{\theta}' =$ *IS-Optimizer*$(\mathcal{D}_\tau, \boldsymbol{\theta})$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}'$
end for
Output: task-agnostic exploration policy $\pi_{\boldsymbol{\theta}}$

IS-Optimizer

Input: dataset of particles $\mathcal{D}_\tau$, sampling parameters $\boldsymbol{\theta}$
initialize $h = 0$ and $\boldsymbol{\theta}_h = \boldsymbol{\theta}$
while $D_{KL}(\bar{d}_T(\boldsymbol{\theta}_0)||\bar{d}_T(\boldsymbol{\theta}_h)) \leq \delta$ **do**
  compute a gradient step:
    $\boldsymbol{\theta}_{h+1} = \boldsymbol{\theta}_h + \alpha\nabla_{\boldsymbol{\theta}_h}\widehat{H}_k\big(\bar{d}_T(\boldsymbol{\theta}_h)|\bar{d}_T(\boldsymbol{\theta}_0)\big)$
  $h \leftarrow h + 1$
end while
Output: parameters $\boldsymbol{\theta}_h$

5 The Algorithm

In this section, we present a model-free policy-search algorithm, *Maximum Entropy POLicy optimization* (MEPOL), to deal with the task-agnostic exploration problem (4) in continuous, high-dimensional domains. MEPOL searches for a policy that maximizes the performance index $\widehat{H}_k(\bar{d}_T(\boldsymbol{\theta}))$ within a parametric space of stochastic differentiable policies $\Pi_\Theta = \{\pi_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^q\}$. The performance index is given by the non-parametric entropy estimator (1) where we replace f with the average state distribution $\bar{d}_T(\cdot|\pi_{\boldsymbol{\theta}}) = \bar{d}_T(\boldsymbol{\theta})$. The approach combines ideas from two successful policy-search algorithms, TRPO (Schulman et al. 2015) and POIS (Metelli et al. 2018), as it is reported in the following paragraphs. Algorithm 1 provides the pseudocode for MEPOL.

Trust-Region Entropy Maximization The algorithm is designed as a sequence of entropy index maximizations, called epochs, within a trust-region around the current policy $\pi_{\boldsymbol{\theta}}$ (Schulman et al. 2015). First, we select an exploration horizon T and an estimator parameter $k \in \mathbb{N}$. Then, at each epoch, a batch of trajectories of length T is sampled from the environment with $\pi_{\boldsymbol{\theta}}$, so as to take a total of N samples. By considering each state encountered in these trajectories as an unweighted particle, we have $\mathcal{D} = \{s_i\}_{i=1}^N$ where $s_i \sim \bar{d}_T(\boldsymbol{\theta})$. Then, given a trust-region threshold δ, we aim

to solve the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{\theta}' \in \Theta}{\text{maximize}} \quad & \widehat{H}_k\big(\bar{d}_T(\boldsymbol{\theta}')\big) \\ \text{subject to} \quad & D_{KL}\big(\bar{d}_T(\boldsymbol{\theta}) \big|\big| \bar{d}_T(\boldsymbol{\theta}')\big) \leq \delta. \end{aligned} \tag{5}$$

The idea is to optimize Problem (5) via Importance Sampling (IS) (Owen 2013), in a fully off-policy manner partially inspired by (Metelli et al. 2018), exploiting the IW entropy estimator (2) to calculate the objective and the KL estimator (3) to compute the trust-region constraint. We detail the off-policy optimization in the following paragraph.

Importance Sampling Optimization We first expand the set of particles $\mathcal{D}$ by introducing $\mathcal{D}_\tau = \{(\tau_i^t, s_i)\}_{i=1}^N$, where $\tau_i^t = (s_i^0, \dots, s_i^t = s_i)$ is the portion of the trajectory that leads to state s_i. In this way, for any policy $\pi_{\boldsymbol{\theta}'}$, we can associate to each particle its normalized importance weight:

$$\overline{w}_i = \frac{Pr(\tau_i^t|\pi_{\boldsymbol{\theta}'})}{Pr(\tau_i^t|\pi_{\boldsymbol{\theta}})} = \prod_{z=0}^{t} \frac{\pi_{\boldsymbol{\theta}'}(a_i^z|s_i^z)}{\pi_{\boldsymbol{\theta}}(a_i^z|s_i^z)}, \quad w_i = \frac{\overline{w}_i}{\sum_{n=0}^N \overline{w}_n}.$$

Then, having set a constant learning rate α and the initial parameters $\boldsymbol{\theta}_0 = \boldsymbol{\theta}$, we consider a gradient ascent optimization of the IW entropy estimator (2),

$$\boldsymbol{\theta}_{h+1} = \boldsymbol{\theta}_h + \alpha \nabla_{\boldsymbol{\theta}_h} \widehat{H}_k\big(\bar{d}_T(\boldsymbol{\theta}_h)|\bar{d}_T(\boldsymbol{\theta}_0)\big), \tag{6}$$

until the trust-region boundary is reached, i.e., when it holds:

$$\widehat{D}_{KL}\big(\bar{d}_T(\boldsymbol{\theta}_0)\big|\big|\bar{d}_T(\boldsymbol{\theta}_{h+1})\big) > \delta.$$

The following theorem provides the expression for the gradient of the IW entropy estimator in Equation (6).

Theorem 5.1. *Let $\pi_{\boldsymbol{\theta}}$ be the current policy and $\pi_{\boldsymbol{\theta}'}$ a target policy. The gradient of the IW estimator $\widehat{H}_k(\bar{d}_T(\boldsymbol{\theta}')|\bar{d}_T(\boldsymbol{\theta}))$ w.r.t. $\boldsymbol{\theta}'$ is given by:*

$$\nabla_{\boldsymbol{\theta}'} \widehat{H}_k(\bar{d}_T(\boldsymbol{\theta}')|\bar{d}_T(\boldsymbol{\theta})) = -\sum_{i=0}^{N} \frac{\nabla_{\boldsymbol{\theta}'} W_i}{k} \left(V_i^k + \ln \frac{W_i}{V_i^k} \right),$$

where:

$$\nabla_{\boldsymbol{\theta}'} W_i = \sum_{j \in \mathcal{N}_i^k} w_j \times \Bigg(\sum_{z=0}^{t} \nabla_{\boldsymbol{\theta}'} \ln \pi_{\boldsymbol{\theta}'}(a_j^z|s_j^z) - \frac{\sum_{n=1}^N \prod_{z=0}^t \frac{\pi_{\boldsymbol{\theta}'}(a_n^z|s_n^z)}{\pi_{\boldsymbol{\theta}}(a_n^z|s_n^z)} \sum_{z=0}^t \nabla_{\boldsymbol{\theta}'} \ln \pi_{\boldsymbol{\theta}'}(a_n^z|s_n^z)}{\sum_{n=1}^N \prod_{z=0}^t \frac{\pi_{\boldsymbol{\theta}'}(a_n^z|s_n^z)}{\pi_{\boldsymbol{\theta}}(a_n^z|s_n^z)}} \Bigg).$$

6 Empirical Analysis

In this section, we present a comprehensive empirical analysis, which is organized as follows:

6.1) We illustrate that MEPOL allows learning a maximum-entropy policy in a variety of continuous domains, outperforming the current state of the art (MaxEnt);

6.2) We illustrate how the exploration horizon T, over which the policy is optimized, maximally impacts the trade-off between state entropy and mixing time;

6.3) We reveal the significant benefit of initializing an RL algorithm (TRPO) with a MEPOL policy to solve numerous challenging continuous control tasks.

A thorough description of the experimental set-up, additional results, and visualizations are provided in Appendix C.

6.1 Task-Agnostic Exploration Learning

In this section, we consider the ability of MEPOL to learn a task-agnostic exploration policy according to the proposed objective (4). Such a policy is evaluated in terms of its induced entropy value $\widehat{H}_k(\bar{d}_T(\boldsymbol{\theta}))$, which we henceforth refer as *entropy index*. We chose k to optimize the performance of the estimator, albeit experiencing little to none sensitivity to this parameter (Appendix C.3). In any considered domain, we picked a specific T according to the time horizon we aimed to test in the subsequent goal-based setting (Section 6.3). This choice is not relevant in the policy optimization process, while we discuss how it affects the properties of the optimal policy in the next section. Note that, in all the experiments, we adopt a neural network to represent the parametric policy $\pi_{\boldsymbol{\theta}}$ (see Appendix C.2). We compare our algorithm with MaxEnt (Hazan et al. 2019). To this end, we considered their practical implementationof the algorithm to deal with continuous, non-discretized domains (see Appendix C.3 for further details). Note that MaxEnt learns a mixture of policies rather than a single policy. To measure its entropy index, we stick with the original implementation by generating a batch as follows: for each step of a trajectory, we sample a policy from the mixture and we take an action with it. This is not our design choice, while we found that using the mixture in the usual way leads to inferior performance anyway. We also investigated SMM (Lee et al. 2019) as a potential comparison. We do not report its results here for two reasons: we cannot achieve significant performance w.r.t. the random baseline, the difference with MaxEnt is merely in the implementation.

First, we evaluate task-agnostic exploration learning over two continuous illustrative domains: GridWorld (2D states, 2D actions) and MountainCar (2D, 1D). In these two domains, MEPOL successfully learns a policy that evenly covers the state space in a single batch of trajectories (state-visitation heatmaps are reported in Appendix C.3), while showcasing minimal variance across different runs (Figure 1a, 1b). Notably, it significantly outperforms MaxEnt in the MountainCar domain.[2] Additionally, In Figure 1c we show how a batch of samples drawn with a random policy (left) compares to one drawn with an optimal policy (right, the color fades with the time step). Then, we consider a set of continuous control, high-dimensional environments from the Mujoco suite (Todorov, Erez, and Tassa 2012): Ant (29D, 8D), Humanoid (47D, 20D), HandReach (63D, 20D). While we learn a policy that maps full state representations to actions, we maximize the entropy index over a subset of the state space dimensions: 7D for Ant (3D position and 4D torso orientation), 24D for Humanoid (3D position, 4D body orientation, and all the joint angles), 24D for HandReach (full set of joint angles). As we report in Figure 1d, 1e, 1f, MEPOL is able to learn policies with striking entropy values in all the environments. As a by-product, it unlocks several meaningful high-level skills during the process, such as jumping, rotating, navigation (Ant), crawling, standing up (Humanoid), and basic coordination (Humanoid, HandReach). Most importantly, the learning process is not negatively affected by the increas-

[2] We avoid the comparison in GridWorld, since the environment resulted particularly averse to MaxEnt.

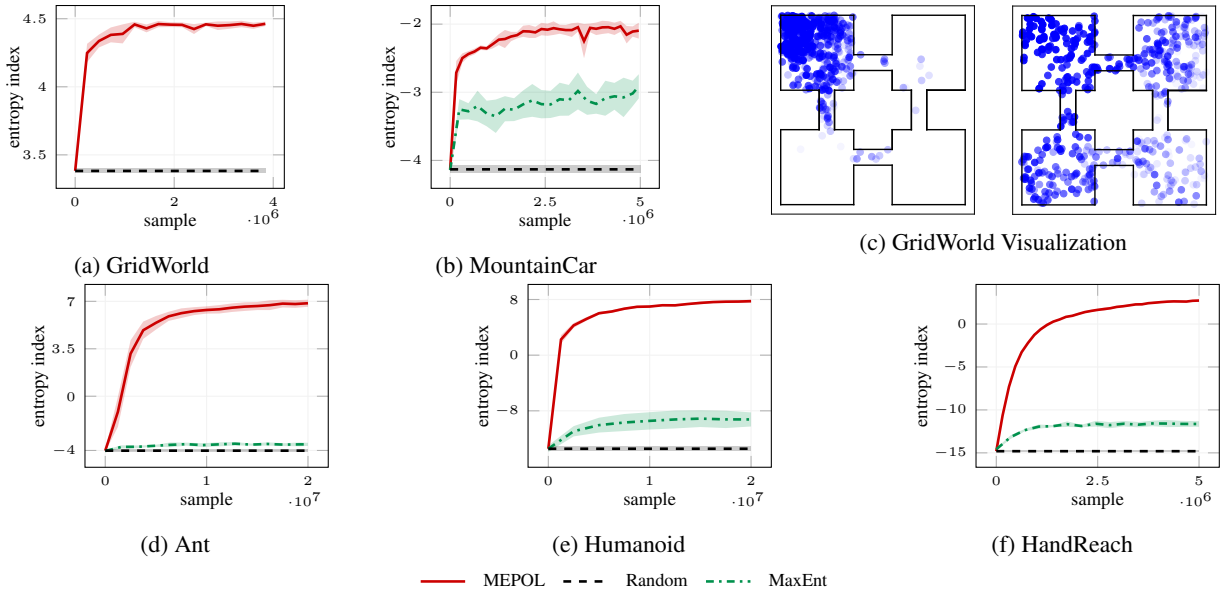


Figure 1: Comparison of the entropy index as a function of training samples achieved by MEPOL, MaxEnt, and a random policy. (95% c.i. over 8 runs. MEPOL: k: 4 (c, d, e, f), 50 (b); T: 400 (c), 500 (d, e, f), 1200 (b). MaxEnt epochs: 20 (c), 30 (d, e, f)).

ing number of dimensions, which is, instead, a well-known weakness of approaches based on explicit density estimation to compute the entropy (Beirlant et al. 1997). This issue is documented by the poor results of MaxEnt, which struggles to match the performance of MEPOL in the considered domains, as it prematurely converges to a low-entropy mixture.

Scalability As we detail above, in the experiments over continuous control domains we do not maximize the entropy over the full state representation. Note that this selection of features is not dictated by the inability of MEPOL to cope with even more dimensions, but to obtain reliable and visually interpretable behaviors (see Appendix C.3 for further details). To prove this point we conduct an additional experiment over a massively high-dimensional GirdWorld domain (200D, 200D), which is reported in Figure 2b.

On MaxEnt Results One might realize that the performance reported for MaxEnt appears to be much lower than the one presented in (Hazan et al. 2019). In this regard, some aspects need to be considered. First, their objective is different, as they focus on the entropy of discounted stationary distributions instead of $\bar{d}_T$. However, in the practical implementation, they consider undiscounted, finite-length trajectories as we do. Secondly, their results are computed over all samples collected during the learning process, while we measure the entropy over a single batch. Lastly, one could argue that an evaluation over the same measure (k-NN entropy estimate) that our method explicitly optimize is unfair. Nevertheless, even evaluating over the entropy of the 2D-discretized state space, which is the measure considered in (Hazan et al. 2019), leads to similar results (as reported in Figure 2a).

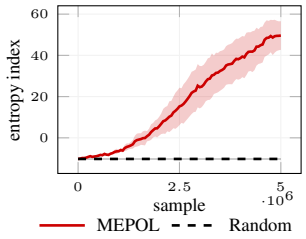
6.2 Impact of the Exploration Horizon Parameter

In this section, we discuss how choosing an exploration horizon T affects the properties of the learned policy. First, it is useful to distinguish between a *training horizon* T, which is an input parameter to MEPOL, and a *testing horizon* h on which the policy is evaluated. Especially, it is of particular interest to consider how an exploratory policy trained over T-steps fares in exploring the environment for a mismatching number of steps h. To this end, we carried out a set of experiments in the aforementioned GridWorld and Humanoid domains. We denote by π_T^* a policy obtained by executing MEPOL with a training horizon T and we consider the entropy of the h-step state distribution induced by π_T^*. Figure 2c (left), referring to the GridWorld experiment, shows that a policy trained over a shorter T might hit a peak in the entropy measure earlier (fast mixing), but other policies achieve higher entropy values at their optimum (highly exploring).[3] It is worth noting that the policy trained over 200-steps becomes overzealous when the testing horizon extends to higher values, while derailing towards a poor h-step entropy. In such a short horizon, the learned policy cannot evenly cover the four rooms and it overfits over easy-to-reach locations. Unsurprisingly, also the average state entropy over h-steps ($\bar{d}_h$), which is the actual objective we aim to maximize in task-agnostic exploration, is negatively affected, as we report in Figure 2c (right). This result points out the importance of properly choosing the training horizon in accordance with the downstream-task horizon the policy will eventually face. However, in other cases a policy learned over T-steps might gracefully generalize to longer horizons, as confirmed by the Humanoid experiment (Figure 2d). The environment is free

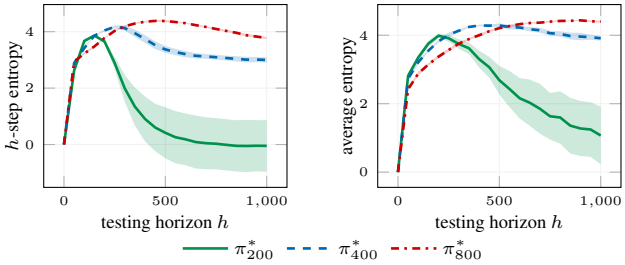
[3]The trade-off between entropy and mixing time has been substantiated for steady-state distributions in (Mutti and Restelli 2020).

	MountainCar	Ant	Humanoid
samples	$5 \cdot 10^6$	$2 \cdot 10^7$	$2 \cdot 10^7$
MEPOL	**4.31 ± 0.04**	**3.67 ± 0.05**	**1.92 ± 0.08**
MaxEnt	3.36 ± 0.4	1.92 ± 0.05	0.96 ± 0.06
Random	1.98 ± 0.05	1.86 ± 0.06	0.84 ± 0.04

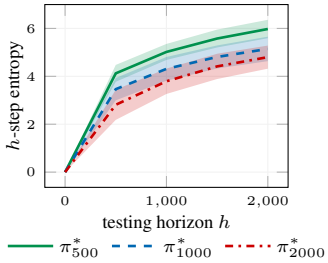
(a) Comparison of the entropy over the 2D-discretized states achieved by MEPOL, MaxEnt, and a random policy (95% c.i. over 8 runs).



(b) 200D-GridWorld



(c) GridWorld



(d) Humanoid

Figure 2: The entropy index over a 200D-GridWorld domain (b). The h-step entropy ($H(d_h^\pi)$) and average entropy ($H(\bar{d}_h)$) achieved by a set of policies trained over different horizons T as a function of the testing horizon h (c, d). (95% c.i. over 8 runs).

of obstacles that can limit the agent's motion, so there is no incentive to overfit an exploration behavior over a shorter T.

6.3 Goal-Based Reinforcement Learning

In this section, we illustrate how a learning agent can benefit from an exploration policy learned by MEPOL when dealing with a variety of goal-based RL tasks. Especially, we compare the performance achieved by TRPO (Schulman et al. 2015) initialized with a MEPOL policy (the one we learned in Section 6.1) w.r.t. a set of significant baselines that learn from scratch, i.e., starting from a randomly initialized policy. These baselines are: TRPO, SAC (Haarnoja et al. 2018), which promotes exploration over actions, SMM (Lee et al. 2019), which has an intrinsic reward related to the state-space entropy, ICM (Pathak et al. 2017), which favors exploration by fostering prediction errors, and Pseudocount (Bellemare et al. 2016), which assigns high rewards to rarely visited states. The algorithms are evaluated in terms of average return on a series of sparse-reward RL tasks defined over the environments we considered in the previous sections.

Note that we purposefully chose an algorithm without a smart exploration mechanism, i.e., TRPO, to employ the MEPOL initialization. In this way we can clearly show the merits of the initial policy in providing the necessary exploration. However, the MEPOL initialization can be combined with any other RL algorithm, potentially improving the reported performance. In view of previous results in task-agnostic exploration learning (Section 6.1), where MaxEnt is plainly dominated by our approach, we do not compare with TRPO initialized with a MaxEnt policy, as it would not be a challenging baseline in this setting.

In GridWorld, we test three navigation tasks with different goal locations (see Figure 3a). The reward is 1 in the states having Euclidean distance to the goal lower than 0.1. For the Ant environment, we define three, incrementally challenging, tasks: Escape, Jump, Navigate. In the first, the Ant starts from an upside-down position and it receives a reward of 1 whenever it rotates to a straight position (Figure 3b). In Jump, the agent gets a reward of 1 whenever it jumps higher than three units from the ground (Figure 3c). In Navigate, the reward is 1 in all the states further than 7 units from the initial location (Figure 3d). Finally, in Humanoid Up, the agent initially lies on the ground and it receives a reward of 1 when it is able to stand-up (Figure 3e). In all the considered tasks, the reward is zero anywhere except for the goal states, an episode ends when the goal is reached.

As we show in Figure 3, the MEPOL initialization leads to a striking performance across the board, while the tasks resulted extremely hard to learn from scratch. In some cases (Figure 3b), MEPOL allows for zero-shot policy optimization, as the optimal behavior has been already learned in the unsupervised exploration stage. In other tasks (e.g., Figure 3a), the MEPOL-initialized policy has lower return, but it permits for lighting fast adaptation w.r.t. random initialization. Note that, to match the tasks' higher-level of abstraction, in Ant Navigate and Humanoid Up we employed MEPOL initialization learned by maximizing the entropy over mere spatial coordinates (x-y in Ant, x-y-z in Humanoid). However, also the exact policies learned in Section 6.1 fares remarkably well in those scenarios (see Appendix C.4), albeit experiencing slower convergence.

7 Discussion and Conclusions

In this paper, we addressed task-agnostic exploration in environments with non-existent rewards by pursuing state entropy maximization. We presented a practical policy-search algorithm, MEPOL, to learn an optimal task-agnostic exploration

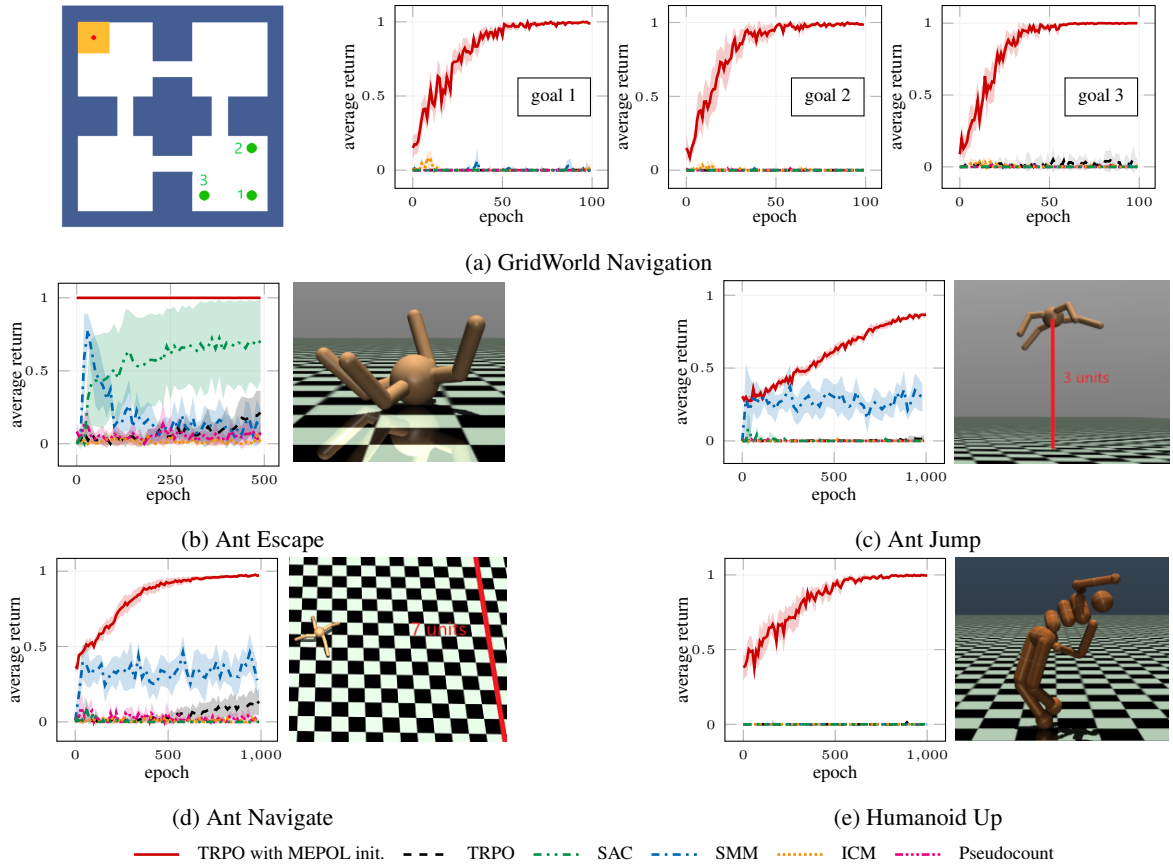


Figure 3: The average return achieved by TRPO with MEPOL initialization, TRPO, SAC, SMM, ICM, and Pseudocount over a set of sparse-reward RL tasks. For each task, we report a visual representation and learning curves. (95% c.i. over 8 runs).

policy in continuous, high-dimensional domains. We empirically showed that MEPOL performs outstandingly in terms of state entropy maximization, and that the learned policy paves the way for solving several reward-based tasks downstream.

Extensions and Future Directions First, the results reported for the goal-based setting (Section 6.3) can be easily extended, either considering a wider range of tasks or combining the MEPOL initialization with a variety of RL algorithm. In principle, any algorithm can benefit from task-agnostic exploration, especially when dealing with sparse-reward tasks. Secondly, while we solely focused on finite-horizon exploration, it is straightforward to adapt the presented approach to the discounted case: We could simply generate a batch of trajectories with a probability $1 - \gamma$ to end at any step instead of stopping at step T, and then keep everything else as in Algorithm 1. This could be beneficial when dealing with discounted tasks downstream. Future work might address an adaptive control over the exploration horizon T, so to induce a curriculum of exploration problems, from an easy problems (short T) to challenging (long T). Promising future directions also include learning task-agnostic exploration across a collection of environments, and contemplating the use of state entropy regularization in reward-based policy optimization.

Other Remarks It is worth mentioning that the choice of a proper metric for the k-NN computation might significantly impact the performance. In our experiments, we were able to get outstanding results with a simple Euclidean metric. However, different domains, such as learning from images, might require the definition of a more thoughtful metric in order to get reliable entropy estimates. In this regard, some recent works (e.g., Misra et al. 2020) provide a blueprint to learn state embeddings in reward-free rich-observation problems. Another theme that is worth exploring to get even better performance over future tasks is sample reuse. In MEPOL, the samples collected during task-agnostic training are discarded, while only the resulting policy is retained. An orthogonal line of research focuses on the problem of collecting a meaningful batch of samples in a reward-free setting (Jin et al. 2020), while discarding sampling policies. Surely a combination of the two objectives will be necessary to get truly efficient methods for task-agnostic exploration, but we believe that these two lines of work still require significant individual advances before being combined into a unique approach.

To conclude, we hope that this work can shed some light on the great potential of state entropy maximization approaches to perform task-agnostic exploration.

References

Achiam, J.; Edwards, H.; Amodei, D.; and Abbeel, P. 2018. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299* .

Ajgl, J.; and Šimandl, M. 2011. Differential entropy estimation by particles. *IFAC* .

Andrychowicz, O. M.; Baker, B.; Chociej, M.; Jozefowicz, R.; McGrew, B.; Pachocki, J.; Petron, A.; Plappert, M.; Powell, G.; Ray, A.; et al. 2020. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research* 39(1): 3–20.

Bechtle, S.; Molchanov, A.; Chebotar, Y.; Grefenstette, E.; Righetti, L.; Sukhatme, G.; and Meier, F. 2019. Meta-learning via learned loss. *arXiv preprint arXiv:1906.05374* .

Beirlant, J.; Dudewicz, E. J.; Györfi, L.; and Van der Meulen, E. C. 1997. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences* .

Bellemare, M.; Srinivasan, S.; Ostrovski, G.; Schaul, T.; Saxton, D.; and Munos, R. 2016. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, 1471–1479.

Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; Debiak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; et al. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680* .

Bonarini, A.; Lazaric, A.; and Restelli, M. 2006. Incremental skill acquisition for self-motivated learning animats. In *International Conference on Simulation of Adaptive Behavior*, 357–368.

Bonarini, A.; Lazaric, A.; Restelli, M.; and Vitali, P. 2006. Self-development framework for reinforcement learning agents. In *Proceedings of the International Conference on Development and Learning*, 355–362.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* .

Burda, Y.; Edwards, H.; Pathak, D.; Storkey, A.; Darrell, T.; and Efros, A. A. 2019a. Large-scale study of curiosity-driven learning. In *Proceedings of the International Conference on Learning Representations*.

Burda, Y.; Edwards, H.; Storkey, A.; and Klimov, O. 2019b. Exploration by random network distillation. In *Proceedings of the International Conference on Learning Representations*.

Chentanez, N.; Barto, A. G.; and Singh, S. P. 2005. Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems*, 1281–1288.

Cobbe, K.; Klimov, O.; Hesse, C.; Kim, T.; and Schulman, J. 2019. Quantifying generalization in reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 1282–1289.

Deisenroth, M. P.; Neumann, G.; Peters, J.; et al. 2013. A survey on policy search for robotics. *Foundations and Trends® in Robotics* 2(1–2): 1–142.

Duan, Y.; Chen, X.; Houthooft, R.; Schulman, J.; and Abbeel, P. 2016. Benchmarking deep reinforcement learning for continuous control. In *Proceedings of the International Conference on Machine Learning*, 1329–1338.

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

Eysenbach, B.; Gupta, A.; Ibarz, J.; and Levine, S. 2019. Diversity is all you need: Learning skills without a reward function. In *Proceedings of the International Conference on Learning Representations*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the International Conference on Machine Learning*, 1126–1135.

Gajane, P.; Ortner, R.; Auer, P.; and Szepesvari, C. 2019. Autonomous exploration for navigating in non-stationary CMPs. *arXiv preprint arXiv:1910.08446* .

Ghasemipour, S. K. S.; Zemel, R. S.; and Gu, S. 2019. A divergence minimization perspective on imitation learning methods. In *Proceedings of the Annual Conference on Robot Learning*, 1259–1277.

Gregor, K.; Rezende, D. J.; and Wierstra, D. 2017. Variational intrinsic control. In *Proceedings of the International Conference on Learning Representations*.

Gupta, A.; Eysenbach, B.; Finn, C.; and Levine, S. 2018. Unsupervised meta-learning for reinforcement learning. *arXiv preprint arXiv:1806.04640* .

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the International Conference on Machine Learning*, 1861–1870.

Hadfield-Menell, D.; Milli, S.; Abbeel, P.; Russell, S. J.; and Dragan, A. 2017. Inverse reward design. In *Advances in Neural Information Processing Systems*, 6765–6774.

Hazan, E.; Kakade, S.; Singh, K.; and Van Soest, A. 2019. Provably efficient maximum entropy exploration. In *Proceedings of the International Conference on Machine Learning*, 2681–2691.

Hodges, J. L.; and Le Cam, L. 1960. The Poisson approximation to the Poisson binomial distribution. *The Annals of Mathematical Statistics* .

Houthooft, R.; Chen, X.; Duan, Y.; Schulman, J.; De Turck, F.; and Abbeel, P. 2016. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, 1109–1117.

Jin, C.; Krishnamurthy, A.; Simchowitz, M.; and Yu, T. 2020. Reward-free exploration for reinforcement learning. In *Proceedings of the International Conference on Machine Learning*.

Kaufmann, E.; Ménard, P.; Domingues, O. D.; Jonsson, A.; Leurent, E.; and Valko, M. 2020. Adaptive reward-free exploration. *arXiv preprint arXiv:2006.06294* .

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

Lim, S. H.; and Auer, P. 2012. Autonomous exploration for navigating in mdps. In *Proceedings of the Conference on Learning Theory*, 40–1.

Lopes, M.; Lang, T.; Toussaint, M.; and Oudeyer, P.-Y. 2012. Exploration in model-based reinforcement learning by empirically estimating learning progress. In *Advances in Neural Information Processing Systems*, 206–214.

Metelli, A. M.; Papini, M.; Faccio, F.; and Restelli, M. 2018. Policy optimization via importance sampling. In *Advances in Neural Information Processing Systems*, 5442–5454.

Misra, D.; Henaff, M.; Krishnamurthy, A.; and Langford, J. 2020. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. In *Proceedings of the International Conference on Machine Learning*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540): 529.

Mohamed, S.; and Rezende, D. J. 2015. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems*, 2125–2133.

Mutti, M.; and Restelli, M. 2020. An intrinsically-motivated approach for learning highly exploring and fast mixing policies. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Oudeyer, P.-Y.; Kaplan, F.; and Hafner, V. V. 2007. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation* 11(2): 265–286.

Owen, A. B. 2013. Monte Carlo theory, methods and examples. *Monte Carlo Theory, Methods and Examples* .

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 16–17.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12: 2825–2830.

Pong, V. H.; Dalal, M.; Lin, S.; Nair, A.; Bahl, S.; and Levine, S. 2020. Skew-fit: State-covering self-supervised reinforcement learning. In *Proceedings of the International Conference on Machine Learning*.

Puterman, M. L. 2014. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons.

Salge, C.; Glackin, C.; and Polani, D. 2014. Empowerment–an introduction. In *Guided Self-Organization: Inception*, 67–114.

Schmidhuber, J. 1987. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. Ph.D. thesis, Technische Universität München.

Schmidhuber, J. 1991. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proceedings of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, 222–227.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *Proceedings of the International Conference on Machine Learning*, 1889–1897.

Shannon, C. E. 1948. A mathematical theory of communication. *Bell System Technical Journal* 379–423.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529(7587): 484.

Singh, H.; Misra, N.; Hnizdo, V.; Fedorowicz, A.; and Demchuk, E. 2003. Nearest neighbor estimates of entropy. *American Journal of Mathematical and Management Sciences* .

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

Tang, H.; Houthooft, R.; Foote, D.; Stooke, A.; Chen, O. X.; Duan, Y.; Schulman, J.; DeTurck, F.; and Abbeel, P. 2017. # Exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 2753–2762.

Tarbouriech, J.; and Lazaric, A. 2019. Active Exploration in Markov Decision Processes. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 974–982.

Tarbouriech, J.; Shekhar, S.; Pirotta, M.; Ghavamzadeh, M.; and Lazaric, A. 2020. Active Model Estimation in Markov Decision Processes. *arXiv preprint arXiv:2003.03297* .

Todorov, E.; Erez, T.; and Tassa, Y. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Zhang, X.; Singla, A.; et al. 2020. Task-agnostic exploration in reinforcement learning. *arXiv preprint arXiv:2006.09497* .

Zheng, Z.; Oh, J.; Hessel, M.; Xu, Z.; Kroiss, M.; van Hasselt, H.; Silver, D.; and Singh, S. 2020. What can learned intrinsic rewards capture? In *Proceedings of the International Conference on Machine Learning*.