Policy Information Capacity: Information-Theoretic Measure for Task Complexity in Deep Reinforcement Learning

Hiroki Furuta [1]   **Tatsuya Matsushima** [1]   **Tadashi Kozuno** [2]   **Yutaka Matsuo** [1]
Sergey Levine [3]   **Ofir Nachum** [3]   **Shixiang Shane Gu** [3]

Abstract

Progress in deep reinforcement learning (RL) research is largely enabled by benchmark task environments. However, analyzing the nature of those environments is often overlooked. In particular, we still do not have agreeable ways to measure the difficulty or solvability of a task, given that each has fundamentally different actions, observations, dynamics, rewards, and can be tackled with diverse RL algorithms. In this work, we propose *policy information capacity (PIC)* – the mutual information between policy parameters and episodic return – and *policy-optimal information capacity (POIC)* – between policy parameters and episodic optimality – as two environment-agnostic, algorithm-agnostic quantitative metrics for task difficulty. Evaluating our metrics across toy environments as well as continuous control benchmark tasks from OpenAI Gym and DeepMind Control Suite, we empirically demonstrate that these information-theoretic metrics have higher correlations with normalized task solvability scores than a variety of alternatives. Lastly, we show that these metrics can also be used for fast and compute-efficient optimizations of key design parameters such as reward shaping, policy architectures, and MDP properties for better solvability by RL algorithms without ever running full RL experiments[1].

1. Introduction

The myriad recent successes of reinforcement learning (RL) have arguably been enabled by the proliferation of deep neural network function approximators applied to rich observational inputs (Mnih et al., 2015; Hessel et al., 2018; Kalashnikov et al., 2018), enabling an agent to adeptly manage long-term sequential decision making in complex environments. While in the past much of the empirical RL research has focused on tabular or linear function approximation case (Dieterich, 1998; McGovern & Barto, 2001; Konidaris & Barto, 2009), the impressive successes of recent years (and anticipation of domains ripe for subsequent successes) has spurred the creation of *non-tabular* benchmarks – i.e., continuous control and/or continuous observation – in which neural network function approximators are effectively a prerequisite (Bellemare et al., 2013; Brockman et al., 2016; Tassa et al., 2018). Accordingly, empirical RL research is presently heavily focused on the use of neural network function approximators, spurring new algorithmic developments in both model-free (Mnih et al., 2015; Schulman et al., 2015; Lillicrap et al., 2016; Gu et al., 2016b; 2017; Haarnoja et al., 2018) and model-based (Chua et al., 2018; Janner et al., 2019; Hafner et al., 2020a) RL.

Despite the impressive progress of RL algorithms, the analysis of the RL *environments* has been difficult and stagnant, precisely due to the complexity of modern benchmarks and neural network architectures required to solve them. Most prior works analyzing sample complexity (a common measure of task complexity) focus on tabular MDPs with finite state and action dimensionalities (Strehl et al., 2006; 2009) or MDPs with simple dynamics (Recht, 2018), and are not applicable or measurable for typical deep RL benchmark tasks. Besides the fact that the components of the environments – observations, actions, dynamics, and rewards – are drastically different in typical benchmarks, the investigation into task solvability is further complicated by the diversity of deep RL algorithms used in practice (Schulman et al., 2015; Lillicrap et al., 2016; Gu et al., 2016b;a; 2017; Haarnoja et al., 2018; Chua et al., 2018; Salimans et al., 2017), where each algorithm has distinct convergence

---

[1]The University of Tokyo, Tokyo, Japan [2]University of Alberta, Edmonton, Canada [3]Google Research, Mountain View, USA. Correspondence to: Hiroki Furuta <furuta@weblab.t.u-tokyo.ac.jp>.

[1]The code is available at https://github.com/frt03/pic.

behaviors. Coming up with a universal, scalable, and measurable definition for task complexity of an RL environment appears an impossible task.

In this work, we propose *policy information capacity (PIC)* and *policy-optimal information capacity (POIC)* as practical metrics for task or environment complexity, taking inspiration from information-theoretic RL research – particularly mutual information maximization or *empowerment* (Klyubin et al., 2005; Tishby & Polani, 2011; Jung et al., 2011; Mohamed & Rezende, 2015; Eysenbach et al., 2019; Warde-Farley et al., 2019; Sharma et al., 2020b; Hafner et al., 2020b)[2]. Policy information capacity measures mutual information between policy parameters and cumulative episodic rewards. As with standard decomposition in mutual information, maximizing policy information capacity corresponds to maximizing reward marginal entropy through policies (diversity maximization) while minimizing reward conditional entropy conditioned on any given policy parameter (predictability maximization), and effectively corresponds to maximizing reward *controllability*. Intuitively, if an agent can easily control rewards or relevant states that affect the cumulative rewards, then most RL algorithms can generally[3] maximize the rewards easily and the environment should be classified as "easy". Alternatively, policy-optimal information capacity (POIC), a variant of PIC drawing on the control as inference literature (Todorov et al., 2006; Toussaint & Storkey, 2006; Rawlik et al., 2012; Fox et al., 2016; Jaques et al., 2017; Haarnoja et al., 2017; Levine, 2018), measures mutual information between policy parameters and whether an episode is optimal or not, and even more closely relates to the optimizability of an RL environment.

We compute policy and policy-optimal information capacities across a range of benchmark environments (Brockman et al., 2016; Tassa et al., 2018), and show that, in practice, especially POIC has a higher correlation with normalized task scores (computed as a brute-force proxy for task complexity by executing many runs of RL algorithms) than other sensible alternatives. Considering the simplicity of our metrics and the drastically varied nature of the benchmarks, our result shows that PIC and POIC can serve as practical and measurable metrics for task complexity.

In summary, our work makes the following contributions:

- We define and propose *policy information capacity (PIC)* and its variant, *policy-optimal information capacity (POIC)* as algorithm-agnostic quantitative metrics for measuring task complexity, and show that, *POIC* in particular corresponds well with empirical task solv-

ability computed across diverse benchmark environments (Brockman et al., 2016; Tassa et al., 2018).

- We set up the first quantitative experimental protocol to evaluate the correctness of a task complexity metric.

- We show that both PIC and POIC can be used as fast proxies for tuning experimental parameters to improve learning progress, such as reward shaping, or policy architectures and initialization parameters, without running any full RL experiment.

2. Related Work

We provide a brief overview of related works, first of previously proposed proxy metrics for assessing the properties of RL algorithms or environments, and then of instances of mutual information (MI) in the context of RL.

Analysis of RL Environments A large body of prior work has sought to theoretically analyze RL algorithms, as opposed to RL environments. For example, Kearns & Singh (2002); Strehl et al. (2009); Dann & Brunskill (2015) derive sample complexity bounds under a finite MDP setting, while Jaksch et al. (2010); Azar et al. (2017); Jin et al. (2018) and Jin et al. (2020) prove regret bounds under finite MDP and linear function approximation settings respectively. Some recent works extend these previous results to non-linear function approximation (Du et al., 2019; Wang et al., 2020; Yang et al., 2020), but they require strong assumptions on function approximators, such as a low Eluder dimension or infinitely-wide neural networks. All these works, however, are algorithm-specific and do not study the properties of RL environments or MDPs.

Asides from algorithms, there are theoretical works that directly study the properties of MDPs. Jaksch et al. (2010) consider the diameter of an MDP, which is the maximum over distinct state pairs (s, s') of expected steps to reach s' from s. Jiang et al. (2017) propose *Bellman rank* and show that an MDP with a low Bellman rank can be provably-efficiently solved. Maillard et al. (2014) propose the environmental norm, the one-step variance of an optimal state-value function. However, those metrics are often intractable to estimate in practical problems, where state or action dimensions are high dimensional (Jaksch et al., 2010; Pong et al., 2018), neural network function approximations are used (Jiang et al., 2017; Dann et al., 2018), or oracle Q-functions are not computable (Jaksch et al., 2010; Jiang et al., 2017; Maillard et al., 2014). Orthogonally to all these works, we propose tractable metrics that can be approximated numerically for complex RL environments with high-dimensional states and actions and, crucially, complex function approximators such as neural networks.

The recent work of Oller et al. (2020) is the closest to ours,

---

[2]"Empowerment" classically measures MI between actions and states, but since rewards can be thought as an additional state dimension, we might regard PIC and POIC as a type of empowerment.

[3]See Section 5 for a thorough discussion.

where they qualitatively visualize marginal reward distributions and show how their variances are intuitively related to environment-specific task difficulty scores estimated from a random search algorithm. While they present very promising early results for tackling this ambitious problem, ours has a few critical differences from their work, which we detail in Section 3.2.

Mutual Information Mutual information has been widely used in RL algorithm development, as a mechanism to encourage emergence of diverse behaviors, sometimes known as *empowerment* (Klyubin et al., 2005; Tishby & Polani, 2011; Jung et al., 2011; Mohamed & Rezende, 2015). Gregor et al. (2016) employ such diverse behaviors as intrinsic-motivation-based exploration methods, which intend to reach diverse states per option, maximizing the lower bound of mutual information between option and trajectory. Related to exploration (Leibfried et al., 2019; Pong et al., 2020), recently MI-based skill discovery has become a popular topic (Florensa et al., 2017; Eysenbach et al., 2019; Warde-Farley et al., 2019; Nachum et al., 2019; Sharma et al., 2020b;a; Campos et al., 2020; Hansen et al., 2020), and these previous works are sources of inspiration for our own metrics, PIC and POIC. For instance, Eysenbach et al. (2019) and Warde-Farley et al. (2019) learn diverse behaviors through maximizing a lower bound on mutual information between skills and future states, which encourages the agent to learn many distinct skills. In other words, the agents learn how to control the environments (future states) via maximization of mutual information. This intuition – that mutual information is related to the controllability of the environments – motivates our own MI-based task solvability metrics, where our metrics PIC and POIC can be seen as *reward* and *optimality empowerments* respectively.

3. Preliminaries

We consider standard RL settings with a Markov Decision Process (MDP) $\mathcal{M}$ defined by state space $\mathcal{S}$, action space $\mathcal{A}$, transition probability $p(s_{t+1}|s_t, a_t)$, initial state distribution $p(s_1)$, and reward function $r(s_t, a_t)$. A policy[4] $\pi(a|s) \in \Pi$ maps from states to probability distributions over actions. With function approximation, this policy is parameterized by $\theta \in \mathbb{R}^d$, initialized by sampling from a prior distribution of the parameter $p(\theta)$[5]. We use the upper case Θ to represent this random variable. We also denote a trajectory as $\tau := (s_1, a_1, s_2, a_2, ..., s_T)$, and a cumulative reward as $r(\tau) :=$

---

[4]While we denote Markovian policies in our derivations, our metrics are also valid for non-Markovian policies.

[5]For the familiarity of notations, we introduce θ as parameters of a parametric function. However, in general, θ can represent the function itself. Since our methods do not require estimations of $\mathcal{H}(\Theta|\cdot)$, any distribution over functions is applicable, e.g. $p(\theta)$ can represent a distribution over different network architectures.

$\sum_{(s,a)\in\tau} r(s, a)$; when clear from the context, we slightly abuse notation and simply use r for $r(\tau)$. We use the upper case R to represent the random variable taking on value $r(\tau)$. Since we focus on evaluation of the environments, we omit a discount factor $\gamma \in [0, 1)$.

The distributions $p(r)$ and $p(r|\theta)$ may be factored as,

$$p(r) = \mathbb{E}_{p(\tau|\theta)p(\theta)}\left[p(r|\tau)\right], \ p(r|\theta) = \mathbb{E}_{p(\tau|\theta)}\left[p(r|\tau)\right],$$

where $p(r|\tau)$ is the reward distribution over trajectory, which, for simplicity, we assume is a deterministic delta distribution, and the marginal distribution of the trajectory conditioned on θ is $p(\tau|\theta) = p(s_1)\prod_{t=1}^{T} p(s_{t+1}|s_t, a_t)\pi(a_t|s_t, \theta)$.

3.1. Optimality Variable

RL concerns with not only characterization of reward distribution, but also its maximization. Information-theoretic perspective on RL, or control as inference (Todorov et al., 2006; Toussaint & Storkey, 2006; Fox et al., 2016; Jaques et al., 2017; Levine, 2018), connects such maximization with probabilistic inference through the notion of "optimality" variable, a binary random variable $\mathcal{O}_t \in \{0, 1\}$ in MDP, where $\mathcal{O}_t = 1$ means the agent behaves "optimally" at timestep t, and $\mathcal{O}_t = 0$ means not optimal. For simplicity, we denote $\mathcal{O}_{1:T}$ as $\mathcal{O}$, which is also a binary random variable, representing whether the agent behaves optimally during the entire episode. We define the distribution of this variable as: $p(\mathcal{O} = 1|\tau) = \exp\left((r - r_{\max})/\eta\right)$, where $\eta > 0$ is a temperature and $r_{\max}$ is the maximum return on the MDP. Note that we subtract $r_{\max}$ to ensure $p(\mathcal{O}|\tau)$ is an appropriate probability distribution.

3.2. Random Weight Guessing

Oller et al. (2020) recently proposed a qualitative analysis protocol of environment complexity with function approximation via random weight guessing. It obtains N particles of θ from prior $p(\theta)$ and runs the deterministic policy $\pi(a_t|s_t, \theta)$ with M episodes per parameter *without any training*. They qualitatively observe that the mean, percentiles, and variance of episodic returns have certain relations with an approximate difficulty of finding a good policy through random search.

However, our work has a number of key differences from their work: (1) we propose a detailed *quantitative* evaluation protocol for verifying task difficulty metrics while they focus on qualitative discussions; (2) we derive our main task difficulty metric based on a mixture of SoTA RL algorithms, instead of random search, to better reflect the diversity of algorithm choices in practice; (3) we estimate reward entropies non-parametrically with many particles to reduce approximation errors, while their variance metric assumes

Gaussianity of reward distributions and poorly approximates in the case of multi-modality; and (4) we verify the metrics on more diverse set of benchmark environments including OpenAI MuJoCo (Brockman et al., 2016) and DeepMind Control Suite (Tassa et al., 2018) while they evaluate on classic control problems only.

4. Policy and Policy-Optimal Information Capacity

We now introduce our own proposed task complexity metrics. We begin with formal definitions for both metrics, and then provide details on how to estimate them.

4.1. Formal Definitions

Policy Information Capacity (PIC)   We define PIC as the mutual information $\mathcal{I}(\cdot;\cdot)$ between cumulative reward R and policy parameter Θ:

$$\mathcal{I}(R;\Theta) = \mathcal{H}(R) - \mathbb{E}_{p(\theta)}\left[\mathcal{H}(R|\Theta=\theta)\right], \qquad (1)$$

where $\mathcal{H}(\cdot)$ is Shannon entropy. The intuitive interpretation is that when the environment gives a more diverse reward signal (first term in Equation 1) and a more consistent reward signal per parameter (second term), it enables the agent to learn better behaviors.

Policy-Optimal Information Capacity (POIC)   We introduce the variant of PIC, termed *Policy-Optimal Information Capacity (POIC)*, defined as the mutual information between the optimality variable and the policy parameter:

$$\mathcal{I}(\mathcal{O};\Theta) = \mathcal{H}(\mathcal{O}) - \mathbb{E}_{p(\theta)}\left[\mathcal{H}(\mathcal{O}|\Theta=\theta)\right]. \qquad (2)$$

4.2. Estimating Policy Information Capacity

In this section, we describe a practical procedure for measuring PIC. In general, it is intractable to compute Equation 1 directly. The typical approach to estimate mutual information is to consider the lower bound (Barber & Agakov, 2004; Belghazi et al., 2018; Poole et al., 2019); however, if we estimate entropies in the one-dimensional reward space, we can use simpler techniques based on discretization (Bellemare et al., 2017).

We employ random policy sampling to measure mutual information between cumulative reward and parameter. Given an environment, a policy network π_θ, and a prior distribution of the policy parameter $p(\theta)$, we generate N particles of θ_i ($i = 1, \ldots, N$) $\sim p(\theta)$ randomly and run the policy π_{θ_i} for M episodes per particle (without any training). In total, we collect NM trajectories and their corresponding cumulative rewards. We use r_{ij} to denote the cumulative rewards of the j-th trajectory using θ_i.

Algorithm 1 PIC/POIC Estimation

Input: MDP $\mathcal{M}$, Policy π, Prior distribution of the parameter $p(\theta)$, Number of parameter N, Number of episodes M. Number of bins B.

1: **for** $i = 1$ **to** N **do**
2:     Generate parameter $\theta_i \sim p(\theta)$ and set it to π.
3:     **for** $j = 1$ **to** M **do**
4:         Initialize MDP $\mathcal{M}$.
5:         Run π_{θ_i} and Collect cumulative reward $r_{ij} \sim p(r|\theta_i)$.
6:     **end for**
7: **end for**
8: (for PIC) Approximate $p(r)$ and all $p(r|\theta_i)$ by the same discretization using B bins.
9: Estimate PIC via Equation 3, and/or POIC via Equation 4.

We then empirically estimate Equation 1 via discretization of the empirical cumulative reward distribution for $p(r)$ and each $p(r|\theta_i)$ using the same B bins. We set min and max values observed in sampling as the limit, and divide it into B ($> M$) equal parts:

$$\hat{\mathcal{I}}(R;\Theta) = -\sum_{b=1}^{B} \hat{p}(r_b) \log \hat{p}(r_b)$$
$$+ \frac{1}{N}\sum_{i=1}^{N}\sum_{b=1}^{B} \hat{p}(r_b|\theta_i) \log \hat{p}(r_b|\theta_i). \qquad (3)$$

While there is an unavoidable approximation error when applying this estimator to continuous probability distributions, this approximation error can be reduced with sufficiently large N, M, B. The sketch of this procedure is described in Algorithm 1.

4.3. Estimating Policy-Optimal Information Capacity

Equation 2 can be approximated by using the same samples from Algorithm 1:

$$p(\mathcal{O} = 1|\theta_i) \approx \frac{1}{M}\sum_{j=1}^{M} \exp\left(\frac{r_{ij} - r_{\max}}{\eta}\right) := \hat{p}_{1i};$$
$$p(\mathcal{O} = 1) \approx \frac{1}{N}\sum_{i=1}^{N}\hat{p}_{1i} := \hat{p}_1,$$

then,

$$\hat{\mathcal{I}}(\mathcal{O};\Theta) = -\hat{p}_1 \log \hat{p}_1 - (1-\hat{p}_1)\log(1-\hat{p}_1)$$
$$+ \frac{1}{N}\left(\sum_{i=1}^{N} \hat{p}_{1i} \log \hat{p}_{1i} + (1-\hat{p}_{1i})\log(1-\hat{p}_{1i})\right). \qquad (4)$$

Compared to PIC, the exponential reward transform in POIC is more likely to favor reward maximization rather than minimization, which is preferred in a task solvability metric. Additionally, since Equation 4 is reduced to the entropies

of discrete Bernoulli distributions, we can avoid reward discretization that is necessary for PIC (Equation 3). However, since $\log p(\mathcal{O} = 1|\theta) = \log \int_r p(\mathcal{O} = 1|r)p(r|\theta)$, the sample-based estimators in Equation 4 are biased (but asymptotically consistent).

Tuning temperature   One disadvantage of POIC is that the choice of temperature η can be arbitrary. To circumvent this, we choose the temperature which maximizes the mutual information: $\eta^* := \arg\max_\eta \mathcal{I}(\mathcal{O}; \Theta)$. In practice, we employ a black-box optimizer to numerically find it.

4.4. Policy Selection and Policy Information Capacity

Before proceeding to experiments, we theoretically explain why a high PIC might imply the ease of solving an MDP. Concretely, we consider how the PIC is related to the ease of choosing a better policy among two policies. Such a situation naturally occurs when random search or evolutionary algorithms are used.

We have the following proposition about relation between the PIC and the ease of policy selection. We say that a policy parameter θ_1 is better than another policy parameter θ_2 if its expected return $E_{p(r|\theta_1)}[r] (:= \mu_{\theta_1})$ is larger than or equal to $E_{p(r|\theta_2)}[r] (:= \mu_{\theta_2})$.

Proposition 1. *Consider a situation where which policy parameter, θ_1 or θ_2, is better based on the order of N-sample-average estimates of expected returns, $\hat{\mu}_1 \approx E_{p(r|\theta_1)}[r]$ and $\hat{\mu}_2 \approx E_{p(r|\theta_2)}[r]$. Assume that $p(r|\theta) = \mathcal{N}(\mu_\theta, \sigma_\theta^2)$ for any $\theta \in \mathbb{R}^d$. Then the probability that we wrongly determine the order of θ_1 and θ_2 is at most*

$$\mathbb{E}\left[\exp\left(-\pi e N \left(\frac{\mu_{\theta_1} - \mu_{\theta_2}}{\exp(\mathcal{H}_1) + \exp(\mathcal{H}_2)}\right)^2\right)\right],$$

where the expectation is with respect to θ_1, θ_2, and $\mathcal{H}_i := \mathcal{H}(R|\Theta = \theta_i)$.

Proof. See Appendix E.   □

Regarding the ease of policy selection, this proposition tells us that policy selection becomes easier when each $\mathcal{H}_i$ is small, and μ_{θ_1} and μ_{θ_2} are distant. A high PIC $\mathcal{I}(R; \Theta)$ suggests small $\mathcal{H}_i$ and a large distance between μ_{θ_1} and μ_{θ_2}. Indeed, to keep $\mathcal{I}(R; \Theta)$ high, the unconditional distribution of r must be broad (high $\mathcal{H}(R)$), and a distribution of r conditioned by each parameter must be narrow (low $\mathcal{H}_i$). To simultaneously achieve these two requirements, what can be done is narrowing each conditional distribution (low $\mathcal{H}_i$), and evenly scattering the conditional distributions over $\mathbb{R}$, resulting in a large distance between μ_{θ_1} and μ_{θ_2}.

5. Synthetic Experiments

As a way of motivating PIC and POIC, we introduce a simple setting in which both metrics correlate with task difficulty.

We aim to investigate the following two questions using simple MDPs in Figure 1 to build intuitions: (1) Do PIC and POIC decrease as the conceptual difficulty of the MDP increases? (2) How much do PIC and POIC change as the parameters of $p(\theta)$ are optimized during training (e.g. via evolutionary strategy)? Additionally, we present comparisons between our information capacity measures and marginal entropies in Appendix B.

We assume the following simple MDP:

- The set of states is given by $\mathcal{S} = \{s_1, s_2, s_3, s_4, s_5\}$. The initial state is $s_1 = [1, 0, 0]$ while the other state vectors are $s_2 = [0, 1, 0], s_3 = [0, 0, 1], s_4 = [1, 1, 1], s_5 = [0, 0, 0]$.

- The action space is $\mathcal{A} = \{a_1, a_2\}$, and the parameterized policy $\pi_\theta(a|s)$ for $\theta \in \mathbb{R}^3$ is given by

$$\pi_\theta(a|s) = \begin{cases} \text{sigmoid}(\theta^{\mathrm{T}} s) & (a = a_1) \\ 1 - \text{sigmoid}(\theta^{\mathrm{T}} s) & (a = a_2). \end{cases}$$

- The transitions are deterministic as illustrated in Figure 1.

- We consider three possible reward functions: r_A, r_B, r_C. For r_A, we have $r_A(s_2) = 1$ and $r_A(s) = 0$ otherwise. For r_B, we have $r_B(s_4) = 1$, and $r_B(s) = 0$ otherwise. For r_C, we have $r_C(s_5) = 1$, and $r_C(s) = 0$ otherwise.

We consider variants of this MDP according to horizon $T \in \{1, 2, 3\}$ and we pair each choice of horizon with a reward function; i.e. horizon $T = 1$ is associated with r_A, horizon $T = 2$ is associated with r_B, and horizon $T = 3$ is associated with r_C. We can describe this MDP as $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, r(s), T\}$. We take the policy parameter prior to be a Gaussian distribution $p(\theta) = \mathcal{N}(\mu, \sigma^2 I)$, where μ, σ are hyper-parameters.
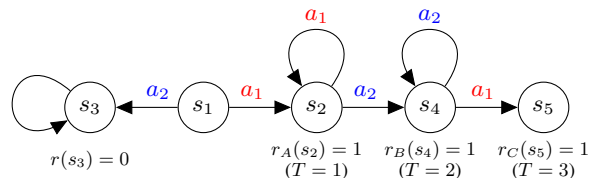

Figure 1. Multi-step discrete MDP. The transition is deterministic and s_3 is an absorbing state. The reward function is determined by horizon T. Intuitively, this MDP becomes difficult when the horizon is longer.

Answer to (1)   We measure both information capacities and normalized score in multi-step MDPs with different horizon $T = 1, 2, 3$ via Algorithm 1. The normalized score

is defined as: $\frac{r_{\text{ave}} - r_{\text{min}}}{r_{\text{max}} - r_{\text{min}}}$, where $r_{\text{ave}}, r_{\text{min}}, r_{\text{max}}$ are the average, maximum, and minimum return over parameters. Intuitively, the MDP in Figure 1 becomes more difficult when the horizon gets longer. We set prior parameter as $\mu = 0$ and $\sigma = 1$. We sample 1,000 parameters from the prior, and evaluate each of them with 1,000 episode per parameter (i.e. $N = 1000$ and $M = 1000$). The results appear in Table 1. We observe that PIC and POIC get lower when the horizon gets longer. The longer horizon MDP leads to a lower normalized score, which means that PIC and POIC properly reflect the task solvability of the MDP.

Horizon	Normalized Score	PIC	POIC
$T = 1$	0.451	0.087	0.087
$T = 2$	0.253	0.064	0.062
$T = 3$	0.112	0.050	0.049

Table 1. The relations between normalized score by random behavior and PIC $\hat{\mathcal{I}}(R; \Theta)$ or POIC $\hat{\mathcal{I}}(\mathcal{O}; \Theta)$ on simple MDP with different horizon (Figure 1). All of them use Gaussian prior $\mathcal{N}(0, I)$. The lower normalized score, the lower information capacity metrics are.

Answer to (2) Our metrics are intrinsically *local*, in that it assumes some $p(\theta)$ for estimation. A natural question is, what happens to these metrics throughout a realistic learning process? To answer this, we optimize μ in $p(\theta)$ to solve the MDP by evolution strategy (Salimans et al., 2017), and observe PIC, POIC, and the agent performance during training. We assume the Gaussian prior $\mathcal{N}(\mu, I)$, and vary μ initializations using $[-10, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 10]$. We set $N = 1000$ and $M = 1000$, and horizon is $T = 3$. Figure 2 presents the results of $\mu = 0, -3.0, -4.0, -5.0$ (for the visibility); the rest of results and the case of PIC are shown in Appendix B. These results confirm that the initial prior with high POIC ($\mu = 0$) actually solves the environment faster than those with low POIC ($\mu = -3.0, -4.0, -5.0$). Interestingly, Figure 2 also shows that high POIC effectively corresponds to regions of fastest learning. This allows us to build an intuition about what happens during learning: in parameter regions with high POIC learning accelerates, and in those with low POIC learning slows down, if $p(\theta)$ corresponds approximately to each local search region per update step. As for PIC, the same trends can be observed in Appendix B. We additionally provide the further examples on more complex environments like classic control or MuJoCo tasks in Appendix D.

6. Deep RL Experiments

In this section, we begin by elaborating on how we derive a brute-force task complexity metric to serve as a ground-truth metric to compare with. Then, we study the following questions: (1) Do PIC and POIC metrics correlate well with task complexity across standard deep RL benchmarks such
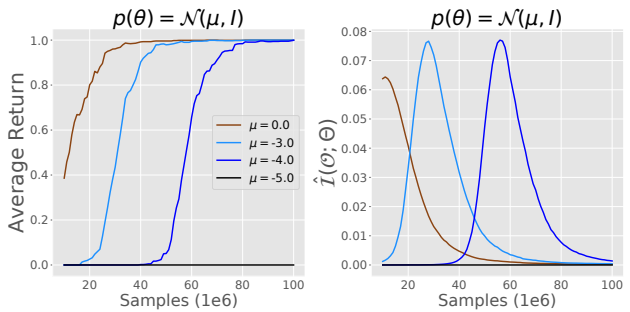

Figure 2. Average return (left) and POIC (right) during the training of evolution strategies. We vary the parameter of initial prior distribution μ and optimize it. High POIC correctly corresponds to regions of fastest learning.

as OpenAI Gym (Brockman et al., 2016; Todorov et al., 2012) and DM Control (Tassa et al., 2018)? (2) Are PIC and POIC more correlated with task complexity than other possible metrics (entropy or variance of returns (Oller et al., 2020))? (3) Can both PIC and POIC be used to evaluate and tune goodness of reward shaping, network architectures, and parameter initialization without requiring running full RL training?

6.1. Defining and Estimating Brute-Force Task Complexity Measure

While an oracle metric for task solvability on complex RL environments seems intractable, one possible (but costly) alternative is to run a large set of diverse RL algorithms and evaluate their normalized average performances. On any given environment, some of these algorithms may completely and efficiently solve the task while others may struggle to learn; an appropriate averaging of the performances of the algorithms can serve as a "ground-truth" task complexity score. As the preparation for the following experiments, we will pre-compute these normalized average performances.

Collecting Raw Algorithm Performances First, we prepare a bag of algorithms (and hyper-parameters) for learning and execute them all. We treat three types of environments separately; classic control, MuJoCo (Brockman et al., 2016), and DM Control (Tassa et al., 2018). For classic control, we run 23 algorithms, based on PPO (Schulman et al., 2017), DQN (Mnih et al., 2015) and Evolution Strategy with different hyper-parameters for discrete-action, and PPO, DDPG (Lillicrap et al., 2016), SAC (Haarnoja et al., 2018), and Evolution Strategy with different hyper-parameters for continuous-action space environments. For MuJoCo and DM Control, we run SAC, MPO (Abdolmaleki et al., 2018) and AWR (Peng et al., 2019), and to simulate more diverse set of algorithms, we additionally incorporated the leaderboard scores reported in previous SoTA works (Fu-

jimoto et al., 2018; Peng et al., 2019; Laskin et al., 2020).
See Appendix C.1 for further details.

Computing Normalized Scores (Algorithm) After collecting raw performances, we compute the average return over the all algorithms $r_{\text{ave}}^{\text{algo}}$. For the comparison, we need to align the range of reward that is different from each environment. To normalize average return over the environments, we take the maximum between this algorithm-based and random-sampling-based maximum scores (explained below), and use the minimum return obtained by random policy sampling:

$$\text{Normalized Score} := \frac{r_{\text{ave}}^{\text{algo}} - r_{\text{min}}^{\text{rand}}}{\max(r_{\text{max}}^{\text{rand}}, r_{\text{max}}^{\text{algo}}) - r_{\text{min}}^{\text{rand}}}.$$

As a sanity check, we checked that the task scores do not trivially correlate with obvious properties of MDP or policy networks, such as state, action dimensionalities; episodic horizon in Appendix C.3.

Computing Normalized Scores (Random Sampling)
In addition to "bag-of-algorithms" task scores, we also compute random-sampling-based task scores that are considered in Oller et al. (2020) for characterizing task difficulties; see Appendix C.2 for more details. While this metric is easier to compute, this only measures the task difficulty of an environment with respect to random search algorithm, and ignores the availability of more advanced RL algorithms.

6.2. Evaluating PIC and POIC as Task Complexity Measures

To verify that our MI-based metrics perform favorably as task solvability metrics in practical settings, we measure both PIC and POIC, along with other alternative metrics, following the random-sampling protocol in Algorithm 1 on the standard RL benchmarks: CartPole, Pendulum, MountainCar, MountainCarContinuous, and Acrobot from classic control in Open AI Gym; HalfCheetah, Walker2d, Humanoid, and Hopper from MuJoCo tasks; cheetah run, reacher easy, and ball_in_cup catch in DM Control (see Appendix A).

We prepare a "bag-of-policy-architectures" to model a realistic prior over policy functions practitioners would use:
([0] layers + [1, 2] layers × [4, 32, 64] hidden units) × [Gaussian prior $\mathcal{N}(0, I)$, Uniform prior $Unif(-1, 1)$, Xavier Normal, Xavier Uniform] × [w/ bias, w/o bias]; totally 56 variants of architectures[6]. We sample 1,000 parameters from the prior for each architecture, and evaluate each of them

with 1,000 episode per parameter ($N = 56 \times 1000 = 56000$ and $M = 1000$) for random policy sampling. The number of bins for discretization is set to $B = 10^5$ for surely maximizing PIC. To compare the suitability of our information capacity metrics and Shannon entropy or variance as task solvability metrics, we compute Pearson correlation coefficients between these measures and the normalized scores for the quantitative evaluation.

Figure 3 visualizes the relation between metrics computed via random sampling (Algorithm 1) and normalized scores (see Table 11 in Appendix F for the detailed scores). Note that variance of returns is scaled by $r_{\text{max}}^{\text{rand}} - r_{\text{min}}^{\text{rand}}$ for a normalized comparison among different environments. The results suggest that POIC seems to positively correlate better with algorithm-based normalized score ($R = 0.807$; statistically significant with $p < 0.01$) compared to any other alternatives, such as reward marginal entropy ($R = -0.349$) or variance of returns ($R = 0.372$) (Oller et al., 2020). Here, we can see that POIC seems to work as a task solvability metric in standard RL benchmarks. In contrast, PIC is correlated with random-sampling-based normalized score ($R = 0.401$) and superior to variance of returns ($R = 0.314$). However, it is less correlated with algorithm-based task scores, which seems closer to actual task difficulty. These differences are possibly due to maximization bias in optimality variable from exponential transform and estimation in Bernoulli space[7].

6.3. Evaluating the Goodness of Reward Shaping and Other "Hyper-Parameters" without RL

We additionally test whether both PIC and POIC can directly evaluate the goodness of reward shaping properly. We investigate two goal-oriented tasks, Reacher (Todorov et al., 2012) and Pointmaze (Fu et al., 2020), where both tasks are defined based on distance functions. We prepare the following four families of distance reward functions: L1 norm, L2 norm, Fraction, and Sparse, each with 1 or 2 hyper-parameters. We select 4 hyper-parameter values for each, totaling 16 different reward functions. To get the normalized scores, we run PPO with 500k steps and 5 seeds for each (see Appendix G for the details).

Figure 4 reveals that Pearson correlation coefficients between the normalized score and information capacity metrics have positive correlations (statistically significant with $p < 0.05$). Coincidentally, clustering of L1 and L2 results reveals that these two reward families are much more robust to ill-specified reward hyper-parameters (i.e. require much less hyper-parameter tuning) than Fraction and Sparse, which is expected given the smoothness and low curvatures of L1 and L2. The results show that *both* PIC and POIC can

---

[6]Oller et al. (2020) by contrast only studied smaller networks, such as 2 layers with 4 units, which were sufficient for their classic control experiments, but certainly would not be for our MuJoCo environments.

[7]The actual behavior of PIC and POIC during RL training might be also related to it (see Appendix D).
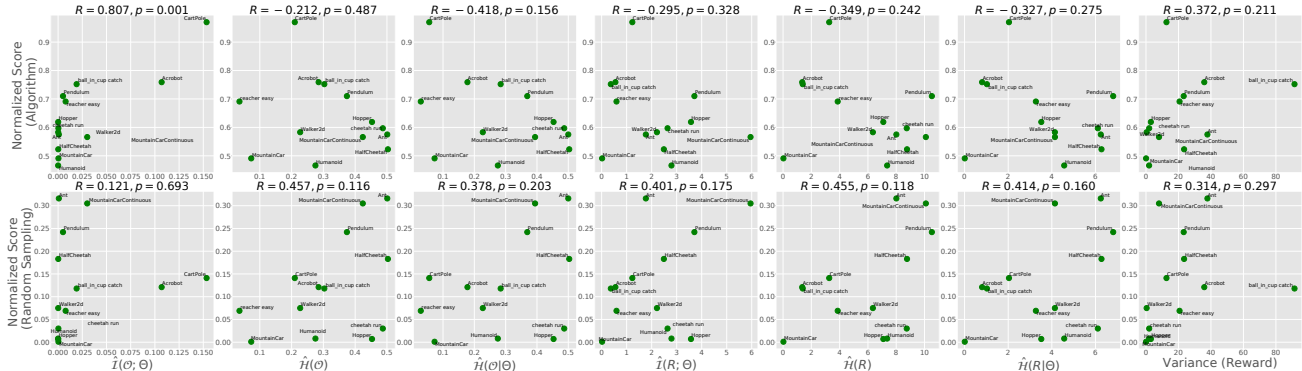
Figure 3. 2D-Scatter plots between each metric (x-axis) and normalized scores (algorithm-based (top) and random-sampling-based (bottom) ; y-axis); see Table 11 for labeled values with environment names. Variance (last column) approximately corresponds to the metric proposed by Oller et al. (2020). **POIC ($\hat{\mathcal{I}}(\mathcal{O}; \Theta)$) positively correlates with algorithm-based normalized score (R = 0.807**; statistically significant with $p < 0.01$), the more realistic of the two task difficulty scores, more than all other alternatives including variance of returns (Oller et al., 2020). Note that the two normalized task scores have a low correlation of 0.139 (see Figure 9), and therefore a high correlation in one means a low correlation in the other.

evaluate the goodness of reward shaping for optimizability[8]. We run additional experiments to evaluate the goodness of architectures, initializations, and dynamics noises. See Appendix H and J for the details.

7. Limitation and Future Work

We tackle a seemingly intractable problem: *to quantify the difficulty of an RL environment irrespective of learning algorithms*. While our empirical evaluations present many encouraging positive results, our metrics have obvious limitations. The biggest is the dependence on $p(\theta)$. As discussed in Section 5, $p(\theta)$ intuitively defines an effective search area (in function space or function parameter space) and our two information capacity metrics approximately measure the easiness of searchability and maximizability respectively within this area weighted by a prior search distribution. Since these metrics in our experiments are only measured at initialization (except results in Figure 2 and Appendix B), they may not correspond well with overall task solvability: (1) if the optimization landscape drastically changes in the later stages of learning, particularly likely for neural networks (Li et al., 2017) or MDPs with discontinuous rewards or dynamics; or (2) if $p(\theta)$ poorly approximates actual search directions given by SGD, natural gradients, Q-learning, etc., during learning. Exploring our metrics throughout the dynamics of optimization and learning to adapt $p(\theta)$ are some of the exciting future directions, along with making connections to works in supervised learning

that relate convergence to signal-to-noise ratios in gradient estimators (Saxe et al., 2019; Smith et al., 2017; Smith & Le, 2017). Another important direction is to scale the evaluations to problems requiring larger neural networks, like ALE with image observations (Bellemare et al., 2013).

8. Conclusion

We defined *policy information capacity (PIC)* and *policy-optimal information capacity (POIC)* as information-theoretic metrics for numerically analyzing the *generic* task complexities of RL environments. These metrics are simple and practical: estimating these metrics only requires a prior distribution over policy functions $p(\theta)$ and trajectory sampling. We formalized a quantitative evaluation protocol for verifying the correctness of task difficulty metrics, that properly accounts for both the richness of available RL algorithms and the complexities of high-dimensional benchmark environments. Through careful experimentation, we successfully identified POIC as the only metric that exhibited high correlations with a brute-force measurement of environment complexity, and demonstrated that these PIC and POIC metrics can be used for tuning task parameters such as reward shaping, MDP dynamics, network architecture and initialization for best learnability without running RL experiments. We hope our work can inspire future research to further explore these long overdue questions of analyzing, measuring, and categorizing the properties of RL *environments*, which can guide us to developing even better learning algorithms.

---

[8]Typically, separately from reward shaping, there is a true task reward (success metric). However, in our definition of task difficulty, we only measure how easy to optimize the given reward (shaped reward). In practice, one should choose a shaped reward that is both easy to optimize (e.g. based on our metric) and accurately reflecting the true task success.
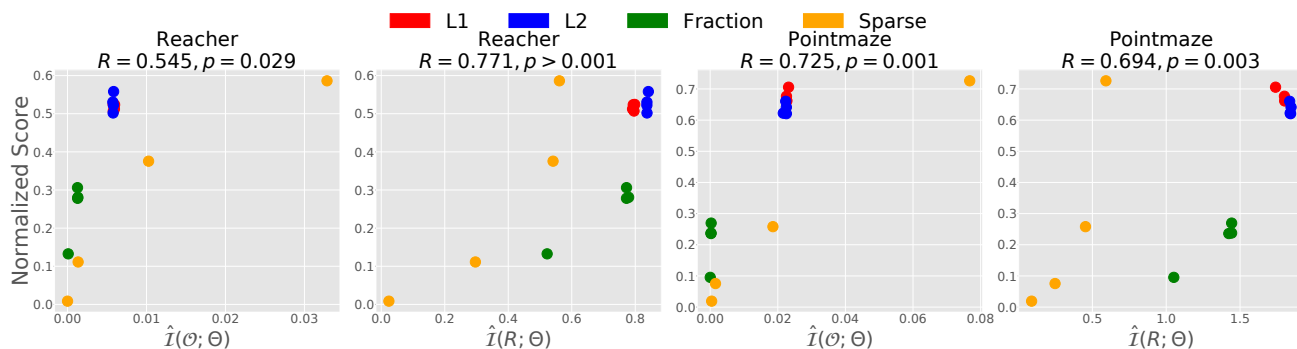
Figure 4. 2D-Scatter plots between PIC or POIC (x-axis) and normalized score (y-axis), after 500k training steps and averaged with 5 seeds. We test 4 family of reward functions (L1 norm, L2 norm, Fraction, Sparse) and 4 variants each (see Appendix G for the details). The results indicate that *both* PIC and POIC can be used to evaluate the goodness of reward shaping proposals. The clusterings of L1 and L2 rewards indicate that learning difficulty with those reward functions does not vary much with hyper-parameters, and suggest that they do not require much hyper-parameter tuning in practice for these tasks.

Acknowledgements

References

Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*, 2018.

Andrychowicz, M., Raichuk, A., Stańczyk, P., Orsini, M., Girgin, S., Marinier, R., Hussenot, L., Geist, M., Pietquin, O., Michalski, M., Gelly, S., and Bachem, O. What matters for on-policy deep actor-critic methods? a large-scale study. In *International Conference on Learning Representations*, 2021.

Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, 2017.

Barber, D. and Agakov, F. The im algorithm : A variational approach to information maximization. In *Advances in neural information processing systems*, 2004.

Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. Mutual information neural estimation. In *International Conference on Machine Learning*, 2018.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 2013.

Bellemare, M. G., Dabney, W., and Munos, R. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, 2017.

Boucheron, S., Lugosi, G., and Massart, P. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, Oxford, UK, 2013.

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

Campos, V. A., Trott, A., Xiong, C., Socher, R., i Nieto, X. G., and Torres, J. Explore, discover and learn: Unsupervised discovery of state-covering skills. In *International Conference on Machine Learning*, 2020.

Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, 2018.

Dann, C. and Brunskill, E. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, 2015.

Dann, C., Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., and Schapire, R. E. On oracle-efficient pac rl with rich observations. In *Advances in Neural Information Processing Systems*, 2018.

Dietterich, T. G. The maxq method for hierarchical reinforcement learning. In *International Conference on Machine Learning*, 1998.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019.

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., and Madry, A. Implementation matters in deep rl: A case study on ppo and trpo. In *International Conference on Learning Representations*, 2019.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning diverse skills without a reward function. In *international conference on learning representations*, 2019.

Florensa, C., Duan, Y., and Abbeel, P. Stochastic neural networks for hierarchical reinforcement learning. In *International conference on learning representations*, 2017.

Fox, R., Pakman, A., and Tishby, N. Taming the noise in reinforcement learning via soft updates. In *Conference on Uncertainty in Artificial Intelligence*, 2016.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Fujimoto, S., van Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, 2018.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2010.

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

Gu, S., Lillicrap, T., Ghahramani, Z., Turner, R. E., and Levine, S. Q-prop: Sample-efficient policy gradient with an off-policy critic. *arXiv preprint arXiv:1611.02247*, 2016a.

Gu, S., Lillicrap, T., Sutskever, I., and Levine, S. Continuous deep q-learning with model-based acceleration. In *International Conference on Machine Learning*, 2016b.

Gu, S. S., Lillicrap, T., Turner, R. E., Ghahramani, Z., Schölkopf, B., and Levine, S. Interpolated policy gradient: Merging on-policy and off-policy gradient estimation for deep reinforcement learning. In *Advances in neural information processing systems*, 2017.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, 2017.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., and Levine, S. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020a.

Hafner, D., Ortega, P. A., Ba, J., Parr, T., Friston, K., and Heess, N. Action and perception as divergence minimization. *arXiv preprint arXiv:2009.01791*, 2020b.

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

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2018.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 2010.

Janner, M., Fu, J., Zhang, M., and Levine, S. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, 2019.

Jaques, N., Gu, S., Bahdanau, D., Hernández-Lobato, J. M., Turner, R. E., and Eck, D. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *International Conference on Machine Learning*, 2017.

Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., and Schapire, R. E. Contextual decision processes with low Bellman rank are PAC-learnable. In *International Conference on Machine Learning*, 2017.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*, 2018.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, 2020.

Jung, T., Polani, D., and Stone, P. Empowerment for continuous agent—environment systems. *Adaptive Behavior*, 2011.

Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., Vanhoucke, V., and Levine, S. QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, 2018.

Kearns, M. and Singh, S. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 2002.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. All else being equal be empowered. In *Advances in Artificial Life*, 2005.

Konidaris, G. and Barto, A. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in Neural Information Processing Systems*, 2009.

Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., and Srinivas, A. Reinforcement learming with augmented data. In *Advances in Neural Information Processing Systems*, 2020.

Leibfried, F., Pascual-Díaz, S., and Grau-Moya, J. A unified bellman optimality principle combining reward maximization and empowerment. In *Advances in Neural Information Processing Systems*, 2019.

Levine, S. Reinforcement Learning and Control as Probabilistic Inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. *arXiv preprint arXiv:1712.09913*, 2017.

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. In *International Conference on Learning Representations*, 2016.

Maillard, O.-A., Mann, T. A., and Mannor, S. How hard is my mdp?" the distribution-norm to the rescue". In *Advances in Neural Information Processing Systems*, 2014.

McGovern, A. and Barto, A. G. Automatic discovery of subgoals in reinforcement learning using diverse density. In *International Conference on Machine Learning*, 2001.

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. Human-level control through deep reinforcement learning. *Nature*, 2015.

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

Nachum, O., Gu, S., Lee, H., and Levine, S. Near-optimal representation learning for hierarchical reinforcement learning. In *international conference on learning representations*, 2019.

Oller, D., Glasmachers, T., and Cuccu, G. Analyzing reinforcement learning benchmarks with random weight guessing. In *International Conference on Autonomous Agents and Multi-Agent Systems*, 2020.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-Weighted Regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Pong, V., Gu, S., Dalal, M., and Levine, S. Temporal difference models: Model-free deep rl for model-based control. *arXiv preprint arXiv:1802.09081*, 2018.

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

Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and Tucker, G. On variational bounds of mutual information. In *International Conference on Machine Learning*, 2019.

Rajeswaran, A., Lowrey, K., Todorov, E. V., and Kakade, S. M. Towards generalization and simplicity in continuous control. In *Advances in Neural Information Processing Systems*, 2017.

Rawlik, K., Toussaint, M., and Vijayakumar, S. On stochastic optimal control and reinforcement learning by approximate inference. In *International Joint Conference on Artificial Intelligence*, 2012.

Recht, B. A tour of reinforcement learning: The view from continuous control. *arXiv preprint arXiv:1806.09460*, 2018.

Salimans, T., Ho, J., Chen, X., Sidor, S., and Sutskever, I. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., and Cox, D. D. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019.

Schulman, J., Levine, S., Moritz, P., Jordan, M., and Abbeel, P. Trust region policy optimization. In *International Conference on Machine Learning*, 2015.

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

Sharma, A., Ahn, M., Levine, S., Kumar, V., Hausman, K., and Gu, S. Emergent real-world robotic skills via unsupervised off-policy reinforcement learning. In *Robotics: Science and Systems*, 2020a.

Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. Dynamics-aware unsupervised discovery of skills. In *International conference on learning representations*, 2020b.

Smith, S. L. and Le, Q. V. A bayesian perspective on generalization and stochastic gradient descent. *arXiv preprint arXiv:1710.06451*, 2017.

Smith, S. L., Kindermans, P.-J., Ying, C., and Le, Q. V. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

Strehl, A. L., Li, L., Wiewiora, E., Langford, J., and Littman, M. L. Pac model-free reinforcement learning. In *International Conference on Machine Learning*, 2006.

Strehl, A. L., Li, L., and Littman, M. L. Reinforcement learning in finite mdps: Pac analysis. *Journal of Machine Learning Research*, 2009.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., de Las Casas, D., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., Lillicrap, T., and Riedmiller, M. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Tishby, N. and Polani, D. Information theory of decisions and actions. In *Perception-action cycle*. Springer, 2011.

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

Todorov, E. et al. Linearly-solvable markov decision problems. In *Advances in Neural Information Processing Systems*, 2006.

Toussaint, M. and Storkey, A. Probabilistic inference for solving discrete and continuous state markov decision processes. In *Proceedings of the 23rd international conference on Machine learning*, pp. 945–952, 2006.

Wang, R., Salakhutdinov, R. R., and Yang, L. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. In *Advances in Neural Information Processing Systems*, 2020.

Warde-Farley, D., de Wiele, T. V., Kulkarni, T. D., Ionescu, C., Hansen, S., and Mnih, V. Unsupervised control through non-parametric discriminative rewards. In *international conference on learning representations*, 2019.

Yang, Z., Jin, C., Wang, Z., Wang, M., and Jordan, M. Provably efficient reinforcement learning with kernel and neural function approximations. In *Advances in Neural Information Processing Systems*, 2020.