When Do Skills Help Reinforcement Learning? A Theoretical Analysis of Temporal Abstractions

Zhening Li [1] **Gabriel Poesia** [2] **Armando Solar-Lezama** [1]

Abstract

Skills are temporal abstractions that are intended to improve reinforcement learning (RL) performance through hierarchical RL. Despite our intuition about the properties of an environment that make skills useful, a precise characterization has been absent. We provide the first such characterization, focusing on the utility of deterministic skills in deterministic sparse-reward environments with finite action spaces. We show theoretically and empirically that RL performance gain from skills is worse in environments where solutions to states are less compressible. Additional theoretical results suggest that skills benefit exploration more than they benefit learning from existing experience, and that using unexpressive skills such as macroactions may worsen RL performance. We hope our findings can guide research on automatic skill discovery and help RL practitioners better decide when and how to use skills.

1. Introduction

In most real-world sequential decision making problems, agents are only given *sparse rewards* for their actions. This makes reinforcement learning (RL) challenging, as agents can only recognize good behavior after long sequences of good decisions. This issue can be mitigated by leveraging *temporal abstractions* (Sutton et al., 1999), also known as *skills*. A skill is a high-level action — such as a fixed sequence of actions (*macroaction*) or a sub-policy with a termination condition (*option*) — that is expected to be useful in a large number of states. Skills can be hand-engineered to perform subtasks (Pedersen et al., 2016; He et al., 2011) or learned from experience (Machado et al., 2017; Bacon et al., 2017; Barreto et al., 2019; Kipf et al., 2019; Jiang et al., 2022; Li et al., 2022). Incorporating skills into the agent's action space (*hierarchical RL*) allows it to act at a higher level and reach goals in fewer steps, which may improve exploration and thus RL performance.

Despite their appeal, skills have not seen widespread use. In fact, they were not involved in most major breakthroughs and applications of RL, such as surpassing human-level performance in all Atari games (Badia et al., 2020), RLHF for aligning LLMs with human preferences (Ouyang et al., 2022), AlphaTensor for faster matrix multiplication (Fawzi et al., 2022), and AlphaDev for faster sorting (Mankowitz et al., 2023). A reason skills have not been widely adopted is that they sometimes do not improve RL performance and it is unclear how to determine beforehand whether they would. While several methods have been developed to automatically discover skills, most of them require the practitioner to decide whether to use skills at all. To our knowledge, LEMMA (Li et al., 2022) is the only algorithm that automatically decides whether skills are useful by learning the optimal number of skills — zero would mean that skills do not help. However, this is accomplished by optimizing a heuristic objective that does not necessarily reflect the benefits to RL. Other skill discovery algorithms such as Option-Critic (Bacon et al., 2017), eigenoptions (Machado et al., 2017), deep skill chaining (Bagaria & Konidaris, 2019), LOVE (Jiang et al., 2022) and COPlanLearn (Nayyar et al., 2023) determine the number of skills using a hyperparameter. A better understanding of how exactly skills benefit RL may guide research in automatically determining whether skills would be useful in an environment and the optimal number to learn if they are. Such an understanding can also provide insight into why skills do not work in certain environments as well as help practitioners better decide whether to use skills for a given RL task.

Our work provides a theoretical analysis of when and how skills and hierarchical RL benefit RL performance in deterministic sparse-reward environments. We hope our insights will serve to guide research in automatic skill discovery including the automatic determination of whether to use skills, and allow practitioners to better understand the kinds of environments where skills are helpful. In summary, we make the following contributions:

[1]MIT CSAIL, Cambridge, MA, USA [2]Stanford University, Stanford, CA, USA. Correspondence to: Zhening Li <zli11010@csail.mit.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

- We define two metrics — p-exploration difficulty and p-learning difficulty — that quantify the hardness of exploration and learning from experience in a deterministic sparse-reward environment with a finite action space. We show empirically that these metrics correlate strongly with the sample complexity of several RL algorithms (Section 3).
- We define two closely related metrics that measure the incompressibility of solutions to states generated by the environment. Under mild assumptions, we prove lower bounds on the change in p-learning difficulty and p-exploration difficulty due to deterministic skills in terms of the incompressibility measures. We show that skills are better suited to decreasing p-exploration difficulty rather than p-learning difficulty, and less expressive skills are less apt at decreasing the difficulty metrics. In particular, for each difficulty metric, we demonstrate the existence of environments where incorporating macroactions provably increases it (Sections 4 and 5).
- We show empirically that macroactions and deep neural options are less beneficial in environments with higher incompressibility (Section 6).
- We describe how to derive skill learning objectives from our incompressibility metrics (Section 7).

All proofs are found in Appendix E. Code for experiments are publicly available at `https://github.com/uranium11010/rl-skill-theory`.

2. Preliminary Definitions

We first introduce basic definitions related to deterministic sparse-reward Markov decision processes (MDPs), which are the focus of this paper. We choose to focus on sparse-reward environments since skills are purported to alleviate the sparse-reward problem. Despite our focus on deterministic environments, a large number of environments both in the standard RL literature (e.g., the original Atari game environments (Bellemare et al., 2013) and MuJoCo (Todorov et al., 2012)) and in applications of RL (e.g., program synthesis (Ellis et al., 2019; Mankowitz et al., 2023) and mathematical reasoning (Kaliszyk et al., 2018; Poesia et al., 2021; Wu et al., 2021)) are deterministic. Furthermore, by focusing on a special case of MDPs, our hardness results — lower bounds on the change in difficulty due to skills — suggest that improving RL using skills in the general case of stochastic environments can be at least as hard. Finally, Appendix F.1 provides preliminary results on generalizing to stochastic environments, suggesting that many insights obtained from studying deterministic environments apply to stochastic ones as well.

Definition 2.1. A *deterministic sparse-reward MDP (DSMDP)* is defined by a 4-tuple $\mathcal{M} = (S, A, T, g)$ where S is the state space, A is the action space, $T : (S \setminus \{g\}) \times A \to S$ is the deterministic transition function and $g \in S$ is the goal state.

Note that environments that have multiple goal states can also be formulated as DSMDPs by merging these goal states into a single goal state. The `CompILE2` environment introduced in Section 3.3 is one such example — see Appendix B for more details.

Borrowing terminology commonly used in symbolic reasoning domains, we say "solve a state" as a shorthand for "finding a sequence of actions that lead to the goal state," and we call such a sequence of actions a *solution*. This is formalized below.

Definition 2.2. A *solution* to a state $s \in S \setminus \{g\}$ of a DSMDP $\mathcal{M} = (S, A, T, g)$ is a sequence of actions $(a_1, \dots, a_l) \in A^l$ ($l \geq 1$) such that applying the sequence of actions starting in s results in the goal state g:

$$T(s, (a_1, \dots, a_l)) = g, \tag{1}$$

where $T(s, (a_1, \dots, a_l)) := T(\cdots(T(s, a_1), a_2)\cdots, a_l)$ denotes the result of applying action sequence $(a_1, \dots, a_l)$ to state s. Here, $l > 0$ is called the *length* of the solution. We will denote by $\mathrm{Sol}_{\mathcal{M}}(s)$ the set of solutions to s and $d_{\mathcal{M}}(s) = \min_{\sigma \in \mathrm{Sol}_{\mathcal{M}}(s)} |\sigma|$ the length of a shortest solution to s.

Note that a state can have no solutions. For example, in domains where we'd like to formalize the notion of "death," one could transition to a "dead state" that goes to itself for all actions taken, and that dead state has no solutions. In contrast, states that have at least one solution are called *solvable* states.

Some results in this paper assume that no two states share a solution, a property we call *solution separability*.

Definition 2.3. A DSMDP is *solution-separable* if no sequence of actions is a solution to more than one state.

Any DSMDP with invertible transitions is solution-separable. Here, we say a DSMDP (S, A, T, g) has invertible transitions if $s = s'$ whenever $T(s, a) = T(s', a)$ and $T(s, a)$ is either solvable or the goal. Examples include (a) all twisty puzzles such as the Rubik's cube; (b) grid world domains where taking a vacuous action (e.g., walking into a wall or picking up a non-existent object) leads to instant death; (c) sliding puzzles where taking a vacuous action leads to instant death.

The following definition formalizes RL in the episodic setting as applied to a DSMDP.

Definition 2.4. In *reinforcement learning (RL) in the episodic setting*, an agent interacts with an environment

(MDP) in *episodes* to learn a policy $\pi(a \mid s)$ that optimizes the expected cumulative reward from one episode. For a DSMDP, the optimal policy is

$$\arg\max_{\pi} \mathop{\mathbb{E}}_{\substack{s_0 \sim p_0 \\ (s_0,a_1,\ldots,a_l,s_l)\sim \mathrm{Rollout}_\pi(s_0)}} \left[\gamma^{l-1}\mathbf{1}[s_l = g]\right]. \quad (2)$$

Here, p_0 is the initial state distribution and $0 < \gamma \leq 1$ is the discount factor. $\mathrm{Rollout}_\pi(s_0)$ is the result of rolling out policy π starting in state s_0, stopping when either the goal state is reached or H actions have been taken, where H is called the *horizon* and sometimes considered part of the definition of an MDP. Note that when $\gamma = 1$, then Equation (2) becomes maximizing the probability that the policy solves $s_0 \sim p_0$.

Now, we introduce *skills*. Whereas skills need not be deterministic in general, we are studying deterministic environments and will thus focus on deterministic skills.

Definition 2.5. A *deterministic skill* in a DSMDP is a function from states to finite action sequences. In other words, for each state, we specify the sequence of actions to be taken if the agent initiates the skill in that state. Note that this sequence is allowed to be empty.

We will refer to deterministic skills as simply "skills."

The prototypical example of an unexpressive class of skills is *macroactions*.

Definition 2.6. A *macroaction* is a skill that produces the same sequence of actions of length greater than 1 regardless of the state in which the skill is initiated.

Incorporating skills into a DSMDP is called a *skill augmentation*, which is more precisely defined below.

Definition 2.7. A DSMDP $\mathcal{M}_0 = (S, A_0, T_0, g)$ augmented with a finite set of skills Z is the DSMDP $\mathcal{M}_+ = (S, A_+, T_+, g)$ where $A_+ = A_0 \cup Z$, $T_+(s,a) = T_0(s,a)$ for $a \in A_0$, and $T_+(s,a) = T_0(s,a(s))$ for $a \in Z$.[1] We say $\mathcal{M}_+$ is the A_+-*skill augmentation* of $\mathcal{M}_0$. We call A_0 the *base action space* and A_+ the *skill-augmented action space*. Furthermore, if $Z \neq \emptyset$ so that A_0 is a proper subset of A_+, then we say the skill augmentation is *strict*.

For simplicity, when discussing a base environment $\mathcal{M}_0$ and its skill augmentation $\mathcal{M}_+$, we will abuse notation by writing subscripts "+" or "0" in places where they should really be "$\mathcal{M}_+$" or "$\mathcal{M}_0$", such as $d_0(s)$ and $\mathrm{Sol}_0(s)$ for $d_{\mathcal{M}_0}(s)$ and $\mathrm{Sol}_{\mathcal{M}_0}(s)$. We allow repetition of skills and skills are also allowed to overlap with base actions. In such cases, Z and A_+ should be interpreted as multisets.

[1] Technically, T_+ is a partial function as $T_+(s,z)$ is undefined if unrolling the skill z reaches the goal state before the unrolling finishes. Thus, in this case, the agent is considered *not* to have reached the goal state. (However, our HRL implementation in our experiments follows the more common convention that the agent is considered successful in this situation.)

3. Quantifying RL Difficulty in a Deterministic Sparse-Reward Environment

To study how much skills can reduce the difficulty of applying RL to a DSMDP, we need to first quantify this difficulty. Unfortunately, existing MDP difficulty metrics fail to capture RL difficulty in DSMDPs since they were not designed to directly estimate sample efficiency or regret, but instead appear in loose asymptotic performance bounds of RL algorithms (see Appendix A for a brief survey). As a result, they correlate poorly with actual performance measures like total regret (Conserva & Rauber, 2022). We therefore aim to develop difficulty metrics for DSMDPs by directly estimating an RL performance measure — in our case, sample efficiency — and to verify them empirically.

Below, we introduce two metrics quantifying the difficulty of applying RL to a deterministic sparse-reward environment, assuming that the environments compared have the same state space (e.g., they are different skill augmentations of the same base environment). We motivate these metrics using heuristic arguments that estimate the sample efficiency of an RL agent in the episodic setting without assuming any particular RL algorithm. We then experimentally test how well the metrics correlate with the sample efficiency of 4 popular RL algorithms in 32 macroaction augmentations of each of 4 base environments.

3.1. Quantifying Difficulty in Learning from Experience

To quantify the complexity of learning a DSMDP from existing experience, suppose that the agent has gathered enough experience to effectively reduce the remaining learning problem to a planning problem. Then Lemma 3.1 shows that the number of iterations through the entire state space needed to learn the value of a state is linear in the minimum length of a solution to that state.

Lemma 3.1. *Suppose we apply value iteration with discount rate $\gamma = 1$ and learning rate α to a DSMDP $\mathcal{M} = (S, A, T, g)$ with a finite action space. In particular, we initialize $V(s) \leftarrow 0$ for $s \neq g$ and $V(g) \leftarrow 1$, and at time t, we update the entire table using*

$$V(s) \leftarrow (1-\alpha)V(s) + \alpha \max_a V(T(s,a)) \quad \text{for all } s \neq g. \quad (3)$$

If $\alpha = 1$, then the number of time steps until the value of a solvable state s becomes its true value (i.e., 1) is $d_{\mathcal{M}}(s)$. If $\alpha < 1$, then the number of time steps until the value of a solvable state s is within ε of its true value (i.e., $1 - V(s) < \varepsilon$) is

$$\Theta\left(\frac{d_{\mathcal{M}}(s) + \log(1/\varepsilon)}{\alpha}\right).$$

Since each iteration has a complexity of $\Theta(|S||A|)$, the total complexity for learning the value of a state s is

$\Theta(|S||A|d_{\mathcal{M}}(s))$ for constant α, ε. If we apply the same intuition to the RL setting, then we would expect that learning the optimal policy at a state s requires $\Theta(d_{\mathcal{M}}(s))$ "iterations," where one "iteration" involves the agent sampling experiences that effectively cover the entire space of state-action pairs. Thus, as a rough estimation, approximately $\Theta(|S_{\text{eff}}||A|d_{\mathcal{M}}(s))$ samples are needed to learn the policy at state s. Here, $|S_{\text{eff}}|$ is some effective size of the state space, counting only those states that we "care about," i.e., those with positive $p_0(s)$ or that are part of (short) solutions to states with positive $p_0(s)$. For constant $|S_{\text{eff}}|$, this estimation of the sample complexity motivates using a weighted average of $|A|d_{\mathcal{M}}(s)$ over states s to measure the complexity of learning from experience.

Definition 3.2. Let $\mathcal{M} = (S, A, T, g)$ be a DSMDP with finite action space A. For a probability distribution p on solvable states, the *p-learning difficulty* of $\mathcal{M}$ is defined as

$$J_{\text{learn}}(\mathcal{M}; p) = |A|\mathbb{E}_{s\sim p}[d_{\mathcal{M}}(s)] \tag{4}$$

where $d_{\mathcal{M}}(s)$ is the length of a shortest solution to s.

The distribution p assigns higher importance to states that we care more about learning to solve. If p_0 denotes the initial state distribution of the MDP, then p should be higher for states with higher p_0. For simplicity, we can just take p to be p_0.

The p-learning difficulty can be viewed as a generalization of diameter (Auer et al., 2008). While the diameter of an MDP is originally defined for the continuous learning setting, a natural extension to the episodic setting for a DSMDP is the maximum length of a solution to a state, $\max_{s\neq g} d_{\mathcal{M}}(s)$. Ignoring the $|A|$ factor, this is the p-learning difficulty when p is zero for all but the state(s) with the largest $d_{\mathcal{M}}(s)$.

3.2. Quantifying Difficulty in Exploration

p-learning difficulty does not take into account the complexity of gathering the needed experience: learning a state s starts to take place only after the agent has seen state-action pairs that form a chain leading from s to the goal state. Thus, as a simplification, an agent's learning process in the episodic setting can be roughly divided into two stages: the first stage is dominated by exploration, where the agent tries to find reward signal and gather experience; the second stage is dominated by learning, where the agent learns from the experience. The sample efficiency of the learning stage is captured by the p-learning difficulty. Let us now motivate the definition of *p-exploration difficulty* by estimating the sample efficiency of the exploration stage.

Suppose that the initial exploration policy is a uniformly random policy, and let $q(s)$ denote the probability that such a policy solves s in one episode. Assuming that the policy remains roughly uniform until the agent finally solves s for the first time, the expected number of episodes until this happens is $1/q(s)$, and the number of environment steps taken is $H/q(s)$ where H is the horizon. To obtain an upper bound on the expected total number of steps taken to find a solution to every state, we simply sum this expression over all states to arrive at $N_{\text{sum}} = H\sum_s \frac{1}{q(s)}$. Note that this can be a significant overestimate of the true sample complexity: solving a state s often updates the agent in a way that helps it solve states whose solutions contain s. We will address this issue later.

For a constant horizon H and state space size, $N_{\text{sum}} \propto \mathbb{E}_{s\sim p}[1/q(s)]$ where p is a uniform distribution over all states. As with the p-learning difficulty, we generalize this to allow different weights $p(s)$ to be assigned to different states. For example, if a state has small $q(s)$ but the MDP's initial state distribution p_0 assigns almost zero probability to s, then we can afford not to learn to solve s and this can be reflected by having $p(s) \approx 0$. For simplicity, we can simply set p to p_0, as with the p-learning difficulty.

We now address the issue of overestimating the sample complexity. In practice, this overestimation is more significant when $q(s)$ for different s are more disparate. In DSMDPs where states vary in difficulty (vary in $q(s)$), solving easy states (states with large $q(s)$) generally updates the agent in a way that helps it find solutions to harder states (states with small $q(s)$). For this reason, we find empirically (Appendix D.2) that the arithmetic mean $N_{\text{AM}} = \mathbb{E}_{s\sim p}[1/q(s)]$ is outperformed by the geometric mean $N_{\text{GM}} = \exp(\mathbb{E}_{s\sim p}[\log(1/q(s))])$, which is lower than N_{AM} when there's variety in $1/q(s)$. Although this estimation of exploration sample complexity is quite rough, it is difficult to make better estimates without knowing details of the MDP structure and RL algorithm. Also, the resultant definition of p-exploration difficulty already performs well empirically on several environments for several RL algorithms (Section 3.3).

Finally, we take the logarithm of N_{GM} as that simplifies notation in our theoretical results. We also replace the fixed horizon with a random horizon sampled from a geometric distribution to simplify theoretical analysis.

Definition 3.3. Let $\mathcal{M} = (S, A, T, g)$ be a DSMDP with finite action space A. For a probability distribution p on solvable states and $0 \leq \delta < 1$, the *δ-discounted p-exploration difficulty* of $\mathcal{M}$ is defined as

$$J_{\text{explore}}(\mathcal{M}; p, \delta) = \mathbb{E}_{s\sim p}[-\log q_{\mathcal{M},\delta}(s)] \tag{5}$$

where

$$q_{\mathcal{M},\delta}(s) := \sum_{\sigma\in \text{Sol}(s)} \left(\frac{1-\delta}{|A|}\right)^{|\sigma|} \tag{6}$$

is the probability that the following policy solves s: at every time step, terminate with probability δ and choose an action

uniformly at random with probability $1-\delta$. $q_{\mathcal{M},\delta}(s)$ is also the probability that the uniformly random policy solves s within a horizon of length H, where $H+1$ is sampled from the geometric distribution with parameter δ.

3.3. Experiments

In motivating p-learning difficulty and p-exploration difficulty, we made significant approximations to estimate the sample complexity without assuming a particular environment or RL algorithm. Despite this, we show empirically that a combination of the two difficult metrics predicts sample complexity well across a variety of environments and RL algorithms.

We study four deterministic sparse-reward environments: (a) `CliffWalking`, a simple grid world (Sutton & Barto, 2018); (b) `CompILE2`, the CompILE grid world with visit length 2 (Kipf et al., 2019); (c) `8Puzzle`, the 8-puzzle; (d) `RubiksCube222`, the 2x2 Rubik's cube. For the computation of p-learning difficulty and p-exploration difficulty to be feasible, p needs to have finite support over a sufficiently small number of states ($\sim 10^7$ or less). To mitigate this limitation, we chose environments for which there exist larger versions with a similar MDP structure. For example, the 2x2 Rubik's cube should behave similarly to the 3x3 cube, 4x4 cube, etc., and the 8-puzzle should behave similarly to the 15-puzzle, 24-puzzle, etc.

Each environment has 32 action space variants, with one being the base environment (the trivial skill augmentation) and 31 with different sets of macroactions. One macroaction augmentation is calculated using LEMMA (Li et al., 2022) on offline data derived from breadth-first search; 5 are variations of that macroaction augmentation; and 25 are generated randomly. More details are given in Appendix B.

We evaluate how well a combination of p-learning difficulty and p-exploration difficulty captures the sample complexity of 4 RL algorithms on the different variants of each environment. The algorithms are: (a) Q-learning (Watkins, 1989); (b) Value iteration (Bellman, 1957), modified to the RL setting, similar to (Agostinelli et al., 2019); (c) REINFORCE (Williams, 1992), made tabular by parameterizing the policy directly with the logits of the actions; (d) Deep Q-networks (DQN) (Mnih et al., 2015).

According to Sections 3.1 and 3.2, we expect J_{learn} to scale roughly linearly with the sample complexity of learning from experience and $\exp(J_{\text{explore}})$ to scale roughly linearly with the sample complexity of exploration. We thus choose a weighted average $J = \lambda J_{\text{learn}} + (1-\lambda)\exp(J_{\text{explore}})$ ($0 \le \lambda \le 1$) to represent the combined difficulty. The discount δ used in the p-exploration difficulty is set to $1/H$, where H is the environment's horizon. The sample complexity N and the combined difficulty J spanned several orders of magnitude in `CliffWalking` and `CompILE2`, so we took the logarithm of both before computing their Pearson correlation coefficient. The value of λ was chosen to maximize this correlation. The results are summarized in Table 1. Most correlation values are at least around 0.7, demonstrating that combining p-learning difficulty and p-exploration difficulty allows us to capture a significant portion of the variation in RL sample efficiency on different action space variants of the same environment.

We also conducted experiments to directly test Lemma 3.1 by computing the correlation between the number of iterations it takes value iteration to converge and the p-weighted average solution length (Appendix D.1). In addition to state value iteration, we also considered Q-value iteration to simulate Q-learning. With two exceptions, all correlations are above 0.9, thus empirically corroborating Lemma 3.1.

4. Effect of Skills on Learning from Experience

Part of our goal is to understand what makes a particular set of skills helpful for an RL agent. One intuition articulated in prior work (Jiang et al., 2022; Kipf et al., 2019) is that skills help *compress* optimal trajectories, making them shorter and thus more likely to be found during exploration. But, conversely, data distributions can be provably *incompressible* when their entropy is too high (Cover, 1994). As a result, we expect that skills are less likely to be helpful when the distribution of optimal trajectories in the environment is incompressible. This intuition is made precise by Theorem 4.2, which states that the ratio between the new and old p-learning difficulties after an A_+-skill augmentation is lower-bounded by the product of an incompressibility measure and a factor penalizing large $|A_+|$. Before stating the theorem, let's first define this incompressibility measure.

Definition 4.1. Let $\mathcal{M}_0 = (S, A_0, T_0, g)$ be a DSMDP with finite $|A_0| > 1$ and $\mathcal{M}_+ = (S, A_+, T_+, g)$ its A_+-skill augmentation. Let p be a distribution over solvable states. The *A_+-merged p-incompressibility* is defined as

$$\mathrm{IC}_{A_+}(\mathcal{M}_0; p) = \sup_{0<\varepsilon<1} \frac{\mathrm{H}[P_+] - \log\left(\frac{1-\varepsilon}{\varepsilon}\right)}{\mathbb{E}_{s\sim p}[d_0(s)] \log\left(\frac{|A_0|}{1-\varepsilon}\right)}. \quad (7)$$

Here, P_+ is the distribution of canonical shortest solutions in $\mathcal{M}_+$ to states sampled from p, where the canonical shortest solutions are chosen such that $\mathrm{H}[P_+]$ is maximized. Note that $\mathrm{H}[P_+]$ is the entropy of the state distribution after states with the same canonical solution in $\mathcal{M}_+$ have been merged into one state. Thus, it has the property $\mathrm{H}[P_+] \le \mathrm{H}[p]$, where equality holds iff all states in the support of p have different canonical solutions.

A_+-merged p-incompressibility can be understood as the coding efficiency of using base actions to write solutions to

Table 1. Correlations between $\log N$ and $\log J$ where N is the number of environment steps the agent takes to learn the environment and $J = \lambda J_{\text{learn}} + (1-\lambda)\exp(J_{\text{explore}})$ is a weighted average of the p-learning difficulty and the exponential of the p-exploration difficulty. Convergence criteria include reaching a certain reward threshold r^* (0.5 for `RubiksCube222` and 0.9 for the other environments) or reaching a certain threshold ΔQ^* or ΔV^* in the p-weighted average error in action or state values (0.2 for `RubiksCube222` and 0.05 for the other environments). The value of $\lambda \in [0,1]$ was chosen so that the correlation was maximized. Data points where the algorithm never converges before the experiment run ends (100M environment steps) were excluded from the calculation of the correlation. The reported errors are standard errors of the mean over 5 random seeds.

		$\log J_{\texttt{CliffWalking}}$	$\log J_{\texttt{CompILE2}}$	$\log J_{\texttt{8Puzzle}}$	$\log J_{\texttt{RubiksCube222}}$
Q-Learning	$\log N_{r \geq r^*}$	0.947 ± 0.006	0.792 ± 0.025	0.403 ± 0.036	0.857 ± 0.023
	$\log N_{\overline{\Delta Q} \leq \Delta Q^*}$	0.953 ± 0.008	0.786 ± 0.023	0.671 ± 0.056	0.937 ± 0.003
Value iteration	$\log N_{r \geq r^*}$	0.933 ± 0.009	0.825 ± 0.018	0.693 ± 0.051	0.785 ± 0.031
	$\log N_{\overline{\Delta V} \leq \Delta V^*}$	0.951 ± 0.015	0.849 ± 0.013	0.885 ± 0.011	0.748 ± 0.029
REINFORCE	$\log N_{r \geq r^*}$	0.949 ± 0.006	0.869 ± 0.013	0.678 ± 0.020	0.892 ± 0.029
DQN	$\log N_{r \geq r^*}$	0.789 ± 0.028	0.758 ± 0.076	0.583 ± 0.039	0.753 ± 0.019

states sampled from p as opposed to using a code optimized for the distribution of shortest solutions with skills. More precisely, we can write

$$\mathrm{IC}_{A_+}(\mathcal{M}_0; p) = \sup_{0<\varepsilon<1} \frac{\mathrm{H}[P_+] - \log\left(\frac{1-\varepsilon}{\varepsilon}\right)}{\mathrm{H}[P_0, P_{0,\text{unif},\varepsilon}] - \log\left(\frac{1-\varepsilon}{\varepsilon}\right)}, \quad (8)$$

where $\mathrm{H}[P_+]$ is the optimal expected number of bits needed to encode a (canonical) shortest solution in $\mathcal{M}_+$ to a state $s \sim p$, and $\mathrm{H}[P_0, P_{0,\text{unif},\varepsilon}]$ denotes the cross entropy between P_0 and $P_{0,\text{unif},\varepsilon}$. P_0 is the distribution of shortest solutions to states sampled from p containing only base actions. $P_{0,\text{unif},\varepsilon}(\sigma) = \varepsilon(1-\varepsilon)^{|\sigma|-1}|A_0|^{-|\sigma|}$ is a uniform prior over base action sequences. $\mathrm{H}[P_0, P_{0,\text{unif},\varepsilon}]$ is thus the expected number of bits required to encode a shortest solution using a fixed-length code over base actions A_0, optimized for a termination symbol that appears at the end of each time step with probability ε.

We now introduce the theorem, which shows how A_+-merged p-incompressibility can be used to bound how much skills in A_+ can improve p-learning difficulty.

Theorem 4.2. *Let $\mathcal{M}_+ = (S, A_+, T_+, g)$ be the A_+-skill augmentation of the DSMDP $\mathcal{M}_0 = (S, A_0, T_0, g)$ with finite $|A_0| > 1$, and p a probability distribution over solvable states. Then*

$$\frac{J_{\text{learn}}(\mathcal{M}_+; p)}{J_{\text{learn}}(\mathcal{M}_0; p)} \geq \frac{|A_+| \log |A_0|}{|A_0| \log |A_+|} \mathrm{IC}_{A_+}(\mathcal{M}_0; p). \quad (9)$$

We can use Theorem 4.2 to understand the effect that the expressivity of skills has on their ability to improve p-learning difficulty.[2] More expressive skills can encode more diverse behavior and thus allow a larger number of action sequences to be encoded as the same skill. This allows states to share solutions more often, which decreases $\mathrm{H}[P_+]$ and hence $\mathrm{IC}_{A_+}(\mathcal{M}_0; p)$. As a result, the lower bound on the p-learning difficulty ratio decreases. As concrete examples, if we place no restriction on what kinds of skills are allowed, then we can simply include a single skill that solves all solvable states, resulting in $\mathrm{IC}_{A_+}(\mathcal{M}_0; p) = 0$ and $J_{\text{learn}}(\mathcal{M}_+; p) = |A_0| + 1$. This is less than $J_{\text{learn}}(\mathcal{M}_0; p)$ whenever $\mathbb{E}_{s\sim p}[d_0(s)] > 1 + 1/|A_0|$, which is true for all RL environments of practical interest. If a skill is allowed to be a concrete sequence of actions and loops of actions, then states whose solutions involve different numbers of repetitions of the same component will have the same solution containing a skill with a loop whose body is that component. Thus, $\mathrm{H}[P_+] < \mathrm{H}[p]$ but is larger than the value of zero obtained when no restriction is placed on skills. Finally, if skills are restricted to macroactions, then distinct solutions remain distinct after rewriting with macroactions, and so the A_+-merged p-incompressibility achieves its maximum value. In solution-separable environments, this maximum value is equal to the *unmerged p-incompressibility* (Definition 4.3), in which case Theorem 4.2 can be restated in terms of it (Corollary 4.4).

Definition 4.3. Let $\mathcal{M} = (S, A, T, g)$ be a DSMDP with finite $|A| > 1$ and p a distribution over solvable states. The *unmerged p-incompressibility* is defined as

$$\mathrm{IC}(\mathcal{M}; p) = \sup_{0<\varepsilon<1} \mathrm{IC}(\mathcal{M}; p, \varepsilon), \quad (10)$$

where the *ε-discounted unmerged p-incompressibility*

$$\mathrm{IC}(\mathcal{M}; p, \varepsilon) = \frac{\mathrm{H}[p] - \log\left(\frac{1-\varepsilon}{\varepsilon}\right)}{\mathbb{E}_{s\sim p}[d_{\mathcal{M}}(s)] \log\left(\frac{|A|}{1-\varepsilon}\right)}. \quad (11)$$

It measures incompressibility on a scale from 0 to 1 if $\mathcal{M}$ is solution-separable. Furthermore, unlike the A_+-merged

[2] See Appendix F.2 for a more formal treatment where the incompressibility measure in Theorem 4.2 is replaced with one defined explicitly in terms of a quantitative measure of expressivity.

p-incompressibility, it is a function of only $\mathcal{M}$ and p and is thus a general measure of the incompressibility of $\mathcal{M}$.

Corollary 4.4 (Corollary to Theorem 4.2)**.** *In the setup to Theorem 4.2, suppose $\mathcal{M}_0$ is solution-separable*[3] *and A_+ is a macroaction augmentation. Then*

$$\frac{J_{\text{learn}}(\mathcal{M}_+; p)}{J_{\text{learn}}(\mathcal{M}_0; p)} \geq \frac{|A_+| \log |A_0|}{|A_0| \log |A_+|} \text{IC}(\mathcal{M}_0; p). \tag{12}$$

A direct consequence of the above corollary is that there exist environments where incorporating macroactions will always worsen p-learning difficulty, no matter how many there are or what they are.

Corollary 4.5 (Corollary to Corollary 4.4)**.** *In the setup to Theorem 4.2, suppose $\mathcal{M}_0$ is solution-separable and A_+ is a strict macroaction augmentation. If*

$$1 - \text{IC}(\mathcal{M}_0; p) \leq \frac{1}{|A_0| + 1} \left(1 - \frac{1}{\ln |A_0|}\right),$$

then $J_{\text{learn}}(\mathcal{M}_+; p) > J_{\text{learn}}(\mathcal{M}_0; p)$.

5. Effect of Skills on Exploration

To study the properties of a DSMDP that make exploration difficult, we have derived a tight lower bound on the p-exploration difficulty of a DSMDP in terms of the entropy of p and a term representing how dense solutions to states are in the space of all solutions (Theorem 5.2).

Definition 5.1. Let $\mathcal{M} = (S, A, T, g)$ be a DSMDP with finite action space A. For $0 \leq \delta < 1$, the *δ-discounted solution density* of $\mathcal{M}$ is defined as

$$D(\mathcal{M}; \delta) = \sum_s \rho_{\mathcal{M},\delta}(s), \tag{13}$$

where

$$\begin{aligned} \rho_{\mathcal{M},\delta}(s) &= \frac{\delta}{1-\delta} q_{\mathcal{M},\delta}(s) \\ &= \sum_{\sigma \in \text{Sol}_{\mathcal{M}}(s)} \delta(1-\delta)^{|\sigma|-1} |A|^{-|\sigma|} \end{aligned} \tag{14}$$

is the probability that a uniformly random action sequence with length sampled from $\text{Geometric}(\delta)$ solves s.

Theorem 5.2. *Let $\mathcal{M}_+ = (S, A_+, T_+, g)$ be the A_+-skill augmentation of the DSMDP $\mathcal{M}_0 = (S, A_0, T_0, g)$ with a finite action space, and p a probability distribution over solvable states. Then for $0 < \delta < 1$,*

$$J_{\text{explore}}(\mathcal{M}_+; p, \delta) \geq \text{H}[p] - \log\left(\frac{1-\delta}{\delta} D(\mathcal{M}_+; \delta)\right). \tag{15}$$

[3] See Appendix F.3 for the version of this corollary that does not assume solution-separability.

Furthermore, if the state space is finite and $\delta > \max_s p(s)$, then for any $\varepsilon > 0$, there exists an A_+-skill augmentation $\mathcal{M}_+$ of $\mathcal{M}_0$ such that

$$J_{\text{explore}}(\mathcal{M}_+; p, \delta) < \text{H}[p] - \log\left(\frac{1-\delta}{\delta} D(\mathcal{M}_+; \delta)\right) + \varepsilon, \tag{16}$$

thus showing that the lower bound given above is tight for all finite DSMDPs and a large range of δ.

The fact that the lower bound grows with $\text{H}[p]$ is intuitive: when there are many states that we care about learning to solve ($\text{H}[p]$ is large), it is hard for the agent to gather the experience needed to learn to solve all these states (J_{explore} is large). However, incorporating skills only changes the action space and cannot affect $\text{H}[p]$. Skills thus improve exploration by increasing the δ-discounted solution density, which is interpreted as the density of solutions to states within the space of all action sequences. Action sequences of length l equally divide a total density of $\delta(1-\delta)^{l-1}$, so that the combined density of all possible action sequences is 1. If $\mathcal{M}_+$ is solution-separable, then $\sum_s \rho_{+,\delta}(s) \leq 1$, whereas if every action sequence solves some state, then $\sum_s \rho_{+,\delta}(s) \geq 1$. Skills improve exploration by increasing this density, similar to how skills reduce A_+-merged p-incompressibility by allowing more states to share solutions. More expressive skills are more apt at increasing solution density. For example, introducing macroactions in a solution-separable environment results in a solution-separable environment, so the density remains at most 1. If we introduce the logic of loops, then states whose solutions involve different repetitions of the same component can be solved by the same action sequence containing a loop skill, hence increasing the density. In the extreme case where no restriction is placed on the kind of skills allowed, we can introduce many skills, each of which automatically solves all solvable states. The resultant density is approximately $\delta |S_{\text{solvable}}|$, which is usually much larger than 1.

As a corollary to Theorem 5.2, increase in p-exploration difficulty due to macroactions is lower-bounded by the δ-discounted unmerged p-incompressibility (Equation (11)) in solution-separable environments, thus providing the p-exploration difficulty counterpart to Corollary 4.4.

Corollary 5.3 (Corollary to Theorem 5.2)**.** *In the setup to Theorem 5.2, suppose $\mathcal{M}_0$ is solution-separable, $|A_0| > 1$, and A_+ is a macroaction augmentation. Then*

$$\frac{J_{\text{explore}}(\mathcal{M}_+; p, \delta)}{J_{\text{explore}}(\mathcal{M}_0; p, \delta)} \geq \text{IC}(\mathcal{M}_0; p, \delta). \tag{17}$$

Compared to Corollary 4.4, the factor $\frac{|A_+| \log |A_0|}{|A_0| \log |A_+|}$ penalizing large A_+ is absent, and the sup in $\text{IC}(\mathcal{M}_0; p) = \sup_{0<\delta<1} \text{IC}(\mathcal{M}_0; p, \delta)$ has been removed. The resultant

weaker bound suggests that skills are better suited to improving exploration than learning from experience. This is made more precise in Theorem 5.4 and Corollary 5.5 below, but before stating these results, we shall first give an intuitive explanation for why this is the case.

In discussing the effects of skills on learning from existing experience, there was a tradeoff between action space size and reducing solution lengths. Intuitively, while skills allow reward information to propagate to states faster, a large action space means a larger number of experiences to iterate through to efficiently cover the space of all state-action pairs (s, a). Such a tradeoff is not so clear in the effects of skills on exploration. To improve exploration, skills are chosen so that a uniformly random policy in the augmented action space is more likely to reach the goal. If skills are expressive enough, this should always be possible, unless the base action space is already close to optimal. Of course, the most general skills trivially improve p-exploration difficulty by simply mapping every solvable state to the goal, which gives $J_{\text{explore}} \approx 0$. But there can be skills that achieve the maximum possible A_+-merged p-incompressibility (which appears in the lower bound for p-learning difficulty increase in Theorem 4.2) but still decrease p-exploration difficulty. This is made precise by the following theorem.

Theorem 5.4. *Let $\mathcal{M}_0 = (S, A_0, T_0, g)$ be a solution-separable DSMDP with finite $|A_0| > 1$ as well as finite $|S|$. Let p be a probability distribution over solvable states. For all $\delta > \max_s p(s)$ for which $p \not\equiv \rho_{0,\delta}$, there exists an A_+-skill augmentation $\mathcal{M}_+$ of $\mathcal{M}_0$ such that:*

- *There exist distinct shortest solutions in A_+ to all states in the support of p (namely, $\mathrm{H}[P_+]$ achieves its maximum possible value $\mathrm{H}[p]$ and thus $\mathrm{IC}_{A_+}(\mathcal{M}_0; p)$ achieves its maximum possible value $\mathrm{IC}(\mathcal{M}_0; p)$);*
- $J_{\text{explore}}(\mathcal{M}_+; p, \delta) < J_{\text{explore}}(\mathcal{M}_0; p, \delta)$.

Corollary 5.5 (Corollary to Theorem 5.4)**.** *Assume the setup to Theorem 5.4. If*

$$1 - \mathrm{IC}(\mathcal{M}_0; p) \leq \frac{1}{|A_0| + 1}\left(1 - \frac{1}{\ln |A_0|}\right),$$

then there exists a skill augmentation $\mathcal{M}_+$ of $\mathcal{M}_0$ such that $J_{\text{learn}}(\mathcal{M}_+; p) > J_{\text{learn}}(\mathcal{M}_0; p)$ but $J_{\text{explore}}(\mathcal{M}_+; p, \delta) < J_{\text{explore}}(\mathcal{M}_0; p, \delta)$.

Corollary 5.5 shows that there are environments where skills can benefit exploration but harm learning from experience. This again suggests that skills are more apt at improving exploration than learning.

As a final discussion on the effect that skills have on exploration, we answer the question: are there environments where unexpressive skills like macroactions always harm exploration? Unlike Corollary 4.4, there is no penalty factor in the lower bound given in Corollary 5.3. As a result, there is no environment where the lower bound is above 1, which would have implied that all macroaction augmentations increase p-exploration difficulty. Nevertheless, the answer to the question is still affirmative. The following two theorems construct environments where incorporating macroactions always increases p-exploration difficulty, no matter how many there are or what they are.

Theorem 5.6. *Let $\mathcal{M}_0 = (S, A_0, T_0, g)$ be a solution-separable DSMDP with a finite action space such that any state that has a length-1 solution only has length-1 solutions. Let p be a probability distribution over solvable states. Suppose that $\delta > 0$ and*

$$D_{\mathrm{KL}}(p \parallel \rho_{0,\delta}) \leq \frac{\delta^2 \log e}{8(|A_0| + 1)^2}.$$ [4]

Then $J_{\text{explore}}(\mathcal{M}_+; p, \delta) > J_{\text{explore}}(\mathcal{M}_0; p, \delta)$ for any strict macroaction augmentation $\mathcal{M}_+$ of $\mathcal{M}_0$.

Theorem 5.7. *Let $\mathcal{M}_0 = (S, A_0, T_0, g)$ be a solution-separable DSMDP with a finite action space such that: 1) every action sequence is the solution to some state; 2) for every solvable state, all solutions to that state have the same length. Let p be a probability distribution over solvable states such that $p(s)/p(s') = |\operatorname{Sol}_0(s)|/|\operatorname{Sol}_0(s')|$ for any s, s' whose solutions have the same length. Then*

$$J_{\text{explore}}(\mathcal{M}_+; p, 0) - J_{\text{explore}}(\mathcal{M}_0; p, 0) \geq \frac{|A_0|}{|A_+|}\left(1 - \frac{|A_0|}{|A_+|}\right) \quad (18)$$

for any strict A_+-macroaction augmentation $\mathcal{M}_+$ of $\mathcal{M}_0$.

A stronger version of this theorem (Appendix F.4) relaxes the conditions on $\mathcal{M}_0$ and p and the modified bound involves subtracting a corresponding KL-divergence term.

Stated in words, Theorem 5.6 says that macroactions harm exploration when most action sequences are solutions to some state and that a state's assigned importance $p(s)$ is close to the probability that a uniformly random action sequence solves it. Theorem 5.7 suggests that it suffices for $p(s)$ to be roughly proportional to this probability across states whose solutions have the same length. These results make more precise our intuition that it is more difficult to use skills to improve exploration in environments where solutions to states look uniformly randomly distributed.

[4] Technically, $\sum_s \rho_{0,\delta}(s) \leq 1$ but may not equal 1, so the KL-divergence is really between p' and $\rho'_{0,\delta}$, defined such that $p' \equiv p$ and $\rho'_{0,\delta} \equiv \rho_{0,\delta}$ on all solvable states and a dummy state s_d is introduced that bears the remaining probability (i.e., $p'(s_d) = 0, \rho'_{0,\delta}(s_d) = 1 - \sum_{s \neq s_d} \rho_{0,\delta}(s)$).

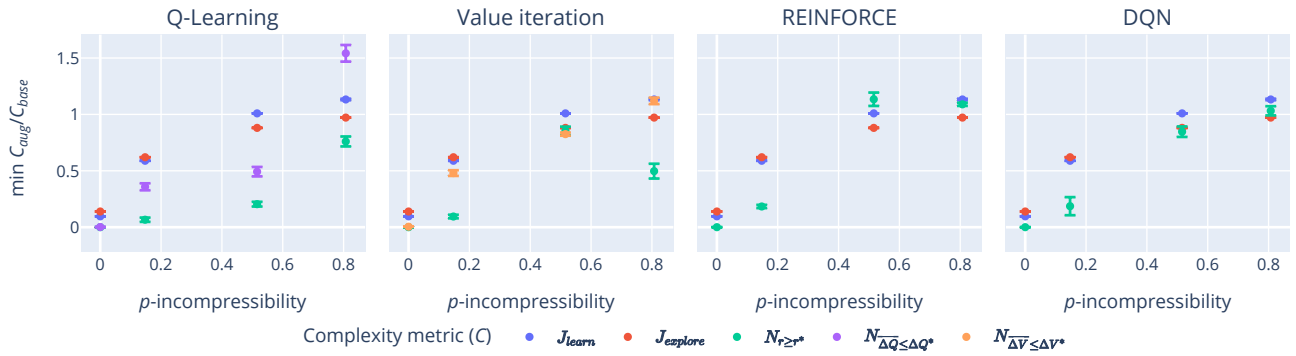

Figure 1. For each of the 4 environments studied, we plot the point (x, y) where x is the unmerged p-incompressibility of the base environment and y is the best complexity improvement ratio $\min C_+/C_0$ over the 31 macroaction augmentations of the base environment. Different colors represent different measures C of complexity, and different panels correspond to sample complexities N of different RL algorithms. The plots corresponding to p-learning difficulty (J_{learn}) and p-exploration difficulty (J_{explore}) have been repeated across panels for clearer comparison with the plots corresponding to the sample complexities (N) of the RL algorithms.

6. Experiments

Corollaries 4.4 and 5.3 suggest that solution-separable DSMDPs with lower unmerged p-incompressibility can benefit more from macroactions. We test this prediction on the four environments studied in Section 3.3, which include both solution-separable (`RubiksCube222`) and non-solution-separable (`CliffWalking`, `CompILE2`, `8Puzzle`) DSMDPs. For different complexity measures C (p-learning difficulty, p-exploration difficulty, and sample complexity N of four RL algorithms), Figure 1 shows the best complexity improvement ratio $\min C_+/C_0$ across the 31 (strict) macroaction augmentations of each base environment against the unmerged p-incompressibility of the base environment. We observe a positive correlation regardless of the choice of C and RL algorithm, thus corroborating our theoretical predictions: macroactions are more helpful in environments with lower unmerged p-incompressibility.

While the definition of unmerged p-incompressibility is motivated in the context of macroactions (Corollaries 4.4 and 5.3), experiments with general stochastic options discovered by LOVE (Jiang et al., 2022) show that it successfully captures the difficulty of applying HRL with general options in an environment. Table 2 shows the unmerged p-incompressibility values of our four environments, along with the sample complexity improvement ratio N_+/N_0 from optionally applying HRL with options discovered by LOVE. The improvement from HRL decreases as the unmerged p-incompressibility increases.

7. p-Incompressibility for Skill Learning

Appendix G demonstrates two ways to use our incompressibility measures to derive objectives for skill learning. We show that, under mild approximations, these two objectives are equivalent to two minimum description length (MDL) objectives previously used in the skill learning literature.

Table 2. Unmerged p-incompressibility $\text{IC}(\mathcal{M}_0; p)$ vs. the improvement ratio N_+/N_0 of sample complexity $N_{r\geq r^*}$ from applying HRL with LOVE options. Results are averaged over 5 seeds. Because HRL can fail to learn an environment on some seeds, we set the improvement ratio to 1 if HRL does not improve the sample complexity.

Environment	N_+/N_0	$\text{IC}(\mathcal{M}_0; p)$
`CliffWalking`	0.000007 ± 0.000007	0.0000
`CompILE2`	0.00023 ± 0.00011	0.1475
`8Puzzle`	0.64 ± 0.19	0.5157
`RubiksCube222`	0.73 ± 0.17	0.8072

In particular, finding the A_+ that minimizes A_+-merged p-incompressibility corresponds to the objective used by LOVE (Jiang et al., 2022), and finding the skills such that the resultant skill-augmented environment has the highest unmerged p-incompressibility corresponds to the objective used by LEMMA (Li et al., 2022).

8. Conclusion

We introduce the first theoretical analysis of the utility of RL skills, focusing on deterministic sparse-reward MDPs. With both theoretical motivation and empirical verification, we introduce metrics that quantify two aspects of RL complexity: exploration and learning from experience. We show both theoretically and experimentally that these metrics can be improved more in environments where solutions to states are more compressible. Further theoretical results suggest that skills benefit exploration more than learning from experience, and that less expressive skills are less beneficial to improving RL sample efficiency. Our work is a first step towards characterizing the properties of an environment that make skills helpful for RL, and we expect future theoretical work to generalize beyond deterministic sparse-reward MDPs with finite action spaces.

Acknowledgements

We thank anonymous referees for useful suggestions and discussions, as well as instructors of the MIT Advanced Undergraduate Research Opportunities Program (SuperUROP) for suggestions on presentation.

This work was funded by U.S. National Science Foundation (NSF) awards #1918771 and #1918839. In addition, ZL was supported by the MIT Advanced Undergraduate Research Opportunities Program (SuperUROP) and GP was supported by the Stanford Interdisciplinary Graduate Fellowship (SIGF).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Agostinelli, F., McAleer, S., Shmakov, A., and Baldi, P. Solving the Rubik's cube with deep reinforcement learning and search. *Nature Machine Intelligence*, 1(8):356–363, 2019.

Auer, P., Jaksch, T., and Ortner, R. Near-optimal regret bounds for reinforcement learning. *Advances in Neural Information Processing Systems*, 21, 2008.

Bacon, P.-L., Harb, J., and Precup, D. The option-critic architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Badia, A. P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, Z. D., and Blundell, C. Agent57: Outperforming the atari human benchmark. In *International Conference on Machine Learning*, pp. 507–517. PMLR, 2020.

Bagaria, A. and Konidaris, G. Option discovery using deep skill chaining. In *International Conference on Learning Representations*, 2019.

Barreto, A., Borsa, D., Hou, S., Comanici, G., Aygün, E., Hamel, P., Toyama, D., Mourad, S., Silver, D., Precup, D., et al. The option keyboard: Combining skills in reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The Aarcade Learning Environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Bellman, R. A Markovian decision process. *Journal of Mathematics and Mechanics*, pp. 679–684, 1957.

Choi, K. P. On the medians of gamma distributions and an equation of Ramanujan. *Proceedings of the American Mathematical Society*, 121(1):245–251, 1994.

Conserva, M. and Rauber, P. Hardness in markov decision processes: Theory and practice. *Advances in Neural Information Processing Systems*, 35:14824–14838, 2022.

Cover, T. Information theory and statistics. In *Proceedings of 1994 Workshop on Information Theory and Statistics*, pp. 2. IEEE, 1994.

Ellis, K., Nye, M., Pu, Y., Sosa, F., Tenenbaum, J., and Solar-Lezama, A. Write, execute, assess: Program synthesis with a repl. *Advances in Neural Information Processing Systems*, 32, 2019.

Fawzi, A., Balog, M., Huang, A., Hubert, T., Romera-Paredes, B., Barekatain, M., Novikov, A., R Ruiz, F. J., Schrittwieser, J., Swirszcz, G., et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022.

He, R., Brunskill, E., and Roy, N. Efficient planning under uncertainty with macro-actions. *Journal of Artificial Intelligence Research*, 40:523–570, 2011.

Hukmani, K., Kolekar, S., and Vobugari, S. Solving twisty puzzles using parallel Q-learning. *Engineering Letters*, 29(4), 2021.

Jiang, Y., Liu, E., Eysenbach, B., Kolter, J. Z., and Finn, C. Learning options via compression. *Advances in Neural Information Processing Systems*, 35:21184–21199, 2022.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 267–274, 2002.

Kaliszyk, C., Urban, J., Michalewski, H., and Olšák, M. Reinforcement learning of theorem proving. *Advances in Neural Information Processing Systems*, 31, 2018.

Kipf, T., Li, Y., Dai, H., Zambaldi, V., Sanchez-Gonzalez, A., Grefenstette, E., Kohli, P., and Battaglia, P. Compile: Compositional imitation learning and execution. In *International Conference on Machine Learning*, pp. 3418–3428. PMLR, 2019.

Li, Z., Poesia, G., Costilla-Reyes, O., Goodman, N., and Solar-Lezama, A. Lemma: Bootstrapping high-level mathematical reasoning with learned symbolic abstractions. *NeurIPS'22 MATH-AI Workshop*, 2022.

Machado, M. C., Bellemare, M. G., and Bowling, M. A laplacian framework for option discovery in reinforcement learning. In *International Conference on Machine Learning*, pp. 2295–2304. PMLR, 2017.

Maillard, O.-A., Mann, T. A., and Mannor, S. "How hard is my MDP?" The distribution-norm to the rescue. *Advances in Neural Information Processing Systems*, 27, 2014.

Mankowitz, D. J., Michi, A., Zhernov, A., Gelmi, M., Selvi, M., Paduraru, C., Leurent, E., Iqbal, S., Lespiau, J.-B., Ahern, A., et al. Faster sorting algorithms discovered using deep reinforcement learning. *Nature*, 618(7964): 257–263, 2023.

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*, 518(7540): 529–533, 2015.

Nayyar, R. K., Verma, S., and Srivastava, S. Learning generalizable symbolic options for transfer in reinforcement learning. In *NeurIPS 2023 Workshop on Generalization in Planning*, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Pedersen, M. R., Nalpantidis, L., Andersen, R. S., Schou, C., Bøgh, S., Krüger, V., and Madsen, O. Robot skills for manufacturing: From concept to industrial deployment. *Robotics and Computer-Integrated Manufacturing*, 37: 282–291, 2016.

Poesia, G., Dong, W., and Goodman, N. Contrastive reinforcement learning of symbolic reasoning domains. *Advances in Neural Information Processing Systems*, 34: 15946–15956, 2021.

Simchowitz, M. and Jamieson, K. G. Non-asymptotic gap-dependent regret bounds for tabular MDPs. *Advances in Neural Information Processing Systems*, 32, 2019.

Sutton, R. S. and Barto, A. G. Temporal difference learning. In *Reinforcement Learning: An Introduction*, chapter 6. MIT Press, 2018.

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

Todorov, E., Erez, T., and Tassa, Y. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Watkins, C. J. C. H. Learning from delayed rewards. 1989.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Wu, M., Norrish, M., Walder, C., and Dezfouli, A. Tacticzero: Learning to prove theorems from scratch with deep reinforcement learning. *Advances in Neural Information Processing Systems*, 34:9330–9342, 2021.

A. Survey on Existing RL Difficulty Metrics

Here, we provide a brief survey on existing RL difficulty metrics and explain why they are inadequate for our purposes. See Conserva & Rauber (2022) for a more detailed survey and benchmark. We will be using the notation $\mathcal{M} = (S, A, P, R)$ for an MDP with state space S, action space A, transition kernel P, and reward kernel R.

- The *environmental value norm of the optimal policy* (Maillard et al., 2014) is given by

$$\sup_{(s,a)\in S\times A} \sqrt{\mathrm{Var}_{s'\sim P(s,a)}\, V_\gamma^*(s')}, \tag{19}$$

  where $P(s,a)$ is the transition kernel of the MDP and V_γ^* is the value function of the optimal policy with discount factor γ. The variation in the values of next states quantifies the difficulty in obtaining accurate sample estimates of action values. However, in deterministic MDPs, which are our focus, the environmental value norm of the optimal policy is always zero and is therefore not applicable.

- The *distribution mismatch coefficient* (Kakade & Langford, 2002) is given by

$$\sup_\pi \sum_{s\in S} \frac{\mu_s^*}{\mu_s^\pi}, \tag{20}$$

  where μ_s^π is the stationary distribution of the Markov chain induced by policy π and μ_s^* is the stationary distribution of the Markov chain induced by the optimal policy. It measures how much the stationary distribution of states visited by the agent can differ from the optimal distribution. It is defined only for ergodic MDPs (otherwise the stationary distribution may not be uniquely defined) in the continuous setting, whereas we focus on deterministic MDPs (which are not ergodic when $|S| > 1$) in the episodic setting.

- The *sum of reciprocals of suboptimality gaps* (Simchowitz & Jamieson, 2019) is given by

$$\sum_{(s,a)\in S\times A:\Delta(s,a)\neq 0} \frac{1}{\Delta(s,a)}, \quad \Delta(s,a) = V^*(s) - Q^*(s,a), \tag{21}$$

  where $V^*(s)$ and $Q^*(s,a)$ are the state and action value functions of the optimal policy. Larger $\Delta(s,a)$ allows the agent to more easily distinguish suboptimal actions from the optimal action and can thus reduce average total regret in the long run. However, as Conserva & Rauber (2022) points out, smaller $\Delta(s,a)$ makes it easier to find a near-optimal policy, which contributes to decreasing the sample complexity.

- The *diameter* (Auer et al., 2008) is defined to be

$$\sup_{s_1\neq s_2} \inf_\pi T^\pi_{s_1\to s_2}, \tag{22}$$

  where $T^\pi_{s_1\to s_2}$ denotes the expected time to reach s_2 starting in s_1 following policy π. While this is defined for the continuous setting, a natural definition for the diameter of a DSMDP $\mathcal{M}$ in the episodic setting would be

$$\sup_{s\neq g:\ \mathrm{Sol}_\mathcal{M}(s)\neq\emptyset} d_\mathcal{M}(s), \tag{23}$$

  where $d_\mathcal{M}(s)$ denotes the length of a shortest solution to s. However, taking the supremum is overly pessimistic, and in many cases, there may be states that are far from the goal but that we do not care about solving. Our p-learning difficulty takes this into account by using a weighted average of $d_\mathcal{M}(s)$, multiplied by $|A|$ to take into account the additional sample complexity due to a large action space.

B. Environments

Experiments were conducted on 4 base environments of varying complexity:

- `CliffWalking` (Sutton & Barto, 2018), a toy grid world environment of size 4×12 where the agent always begins in the bottom left corner and has to travel to the bottom right corner. The available actions are moving one step in each of the 4 cardinal directions. The agent returns to its original position whenever it touches a square in the bottom row other than the leftmost and rightmost squares.

- `CompILE2` is one of the CompILE grid world environments (Kipf et al., 2019). The agent navigates in an 10×10 grid world with walls both lining the edges and within the grid. The world also has several objects of different kinds, possibly with several of each kind. The agent's goal is to pick up several specified (kinds of) objects in order. In `CompILE2`, the agent has to pick up 2 objects. The available actions are moving one step in each of the 4 cardinal directions in addition to attempting to pick up the object in the current cell. The positions and types of the objects are fixed but the agent's position is randomized at every reset, following Jiang et al. (2022). We did not choose 3 or more objects for the agent to pick up because we found that the agent could not find the positive reward signal without suitable skills in these cases, consistent with previous findings on the same environment (Kipf et al., 2019; Jiang et al., 2022). Since whether the goal is reached depends on the sequence of objects the agent has picked up, the state includes both the grid and the sequence of objects that the agent has picked up thus far. Since Kipf et al. (2019) did not publish the source code for the environment, we use the implementation by Jiang et al. (2022).

  Because there can be several of the same kind of object on the grid, there are different sequences of objects the agent can pick up that amount to the same sequence of kinds of objects. There are thus multiple goal states, which are merged into one to comply with the definition of a DSMDP.

- `8Puzzle` is the 8-puzzle, the 3×3 version of the more well-known 15-puzzle. There are 8 tiles numbered 1 to 8 on a 3×3 board so that there is one tile missing. The available actions are moving the position of the missing tile in each of the four cardinal directions. The solved state has the numbers 1 to 8 in order from left-to-right, top-to-bottom. The puzzle is scrambled from the solved state by applying a random legal action K times where K is uniform between 1 and 31. Here, 31 is the maximum distance from any state to the goal state. The puzzle is re-scrambled if the scramble solves the cube.

- `RubiksCube222` is the 2x2 Rubik's cube, also called the pocket cube. The available actions are turning the front, right, or top faces clockwise by $90°$. The cube is scrambled by applying a random sequence of moves of length K where K is uniform between 1 and 11 and where each move is turning the front, right, or top face $90°$ clockwise, $180°$, or $90°$ counterclockwise and no two consecutive moves turn the same face. (Note that the action space used for scrambling is larger than the action space of the agent.) Here, 11 is the maximum number distance from a state to the solved state. We use the implementation provided by Hukmani et al. (2021).

For `8Puzzle` and `RubiksCube222`, our choice of sampling the scramble length uniformly from 1 to some maximum K follows Agostinelli et al. (2019).

Basic information about the 4 base environments is summarized in Table 3.

For each base environment, one of the 32 action space variants is just the base environment itself. The remaining 31 are (strict) macroaction augmentations generated as follows:

- For `CliffWalking`, the LEMMA abstraction algorithm (Li et al., 2022) found one single macroaction from the offline trajectory data generated using breadth-first search (BFS). That single macroaction is just the shortest sequence of actions that solves the only possible starting state of the environment: (U = up, R = right, D = down, L = left)

Table 3. Basic information about the base environments studied by our experiments. $|A_0|$: size of base action space; $|S|$: size of state space; $|S_{p>0}|$: size of support of p.

Environment	$\lvert A_0 \rvert$	$\lvert S \rvert$	$\lvert S_{p>0} \rvert$
`CliffWalking`	4	32	1
`CompILE2`	5	115,462	59
`8Puzzle`	4	362,880	181,439
`RubiksCube222`	3	3,674,160	3,674,159

  - URRRRRRRRRRD

  5 other sets of macroactions were derived from subsequences of near-optimal solutions to the starting state:

  - RR
  - RR, RRRR, RRRRRRRR
  - RRRRRRRRRRR
  - UUURRRR, RRR, DRDRD
  - URRRRRRRRRRR, RRRRRRRRRRD

  Furthermore, for each $k = 1, 2, 3, 4, 5$, we randomly generated 5 sets of k distinct macroactions. A random macroaction with length $L + 1$ ($L \sim \text{Geometric}(1/3)$) was generated as follows:

  - With probability 0.4, randomly choose between U and R with probabilities 0.3 and 0.7;
  - With probability 0.3, randomly choose between R and D with probabilities 0.7 and 0.3;
  - With probability 0.1, randomly choose between D and L with probabilities 0.7 and 0.3;
  - With probability 0.2, randomly choose between L and U with probabilities 0.3 and 0.7.

  We didn't choose probabilities uniform across all directions because this results in several sets of macroactions that cause the agent to drift leftward or downward during random exploration, and the agent almost never receives any positive reward signal. However, it was also the presence of drift that helped us generate variety in the learnability of the macroaction-augmented environments. Variation in the direction of the drift across different sets of macroactions resulted in sample efficiencies that varied across 7 orders of magnitude.

- For `CompILE2`, LEMMA discovered the following set of macroactions: (L = left, U = up, R = right, D = down, P = pick up)

  - PUURRRP, LL, UU, DD

  5 other sets of macroactions were derived from subsequences of subsets of these macroactions:

  - LL, UU, DD
  - LL, UU, RRR, DD
  - PUU, RRRP
  - PUURRRP
  - PUURRRP, LL, UU, RRR, DD

  Furthermore, for each $k = 1, 2, 3, 4, 5$, we randomly generated 5 sets of k distinct macroactions. A random macroaction with length $L + 1$ ($L \sim \text{Geometric}(1/3)$) was generated as follows:

  - With probability 1/4, randomly choose among L, U and P with probabilities 0.4, 0.4 and 0.2;
  - With probability 1/4, randomly choose among U, R and P with probabilities 0.4, 0.4 and 0.2;
  - With probability 1/4, randomly choose among R, D and P with probabilities 0.4, 0.4 and 0.2;
  - With probability 1/4, randomly choose among D, L and P with probabilities 0.4, 0.4 and 0.2.

- For `8Puzzle`, LEMMA discovered the following set of macroactions: (U = up, R = right, D = down, L = left)

  - RD, LDR

  5 other sets of macroactions were derived from subsets of these macroactions, possibly with reflection across the diagonal (a symmetry of the puzzle):

  - RD
  - LDR
  - RD, DR
  - LDR, URD
  - RD, DR, LDR, URD

Furthermore, for each $k = 1, 2, 3, 4, 5$, we randomly generated 5 sets of k distinct macroactions. A random macroaction with length $L + 1$ ($L \sim \text{Geometric}(1/2)$) was generated by sampling from U, R, D, L with probabilities 0.2, 0.3, 0.3, 0.2. The higher probabilities for R and D are intended to encourage moving the position of the missing tile towards the bottom-right corner.

- For `RubiksCube222`, LEMMA generated the empty set. However, the 3 top-scoring macroactions were: (F = front face $90°$, R = right face $90°$, U = top face $90°$)
  - FF, RR, UU

  5 other sets of macroactions were derived from subsets of these macroactions, possibly with more repetition of some base action:
  - FF
  - FF, FFF
  - FF, RR
  - FF, FFF, RR, RRR
  - FF, FFF, RR, RRR, UUU

  Note that FF, RR, UU are half-turns of faces (denoted F2, R2, U2 in standard cube notation) and FFF, RRR, UUU are counter-clockwise $90°$ turns (usually denoted F′, R′, U′).

  Furthermore, for each $k = 1, 2, 3, 4, 5$, we randomly generated 5 sets of k distinct macroactions. A random macroaction with length $L + 1$ ($L \sim \text{Geometric}(1/2)$) was generated by sampling from F, R, U each with probability 1/3.

C. Experimental Details

C.1. Hyperparameters

- The learning rate is $\alpha = 0.1$ for Q-learning, value iteration and REINFORCE, and $\alpha = 0.0005$ for DQN.
- For the off-policy RL algorithms (Q-learning, value iteration, and DQN), the optimal epsilon schedule for epsilon greedy can vary by orders of magnitude across different action space variants of the same base environment. We therefore adopt an adaptive epsilon-greedy exploration policy where the probability ε of choosing a random action starts at 1 and is decreased by 0.002 every time the agent beats its highest test reward so far by 0.002, until $\varepsilon = 0.1$.
- Testing was performed with 200 episodes (1 episode for `CliffWalking`, which only has one starting state) using the greedy policy (Q-learning, value iteration, DQN) or the current policy (REINFORCE). For the purposes of computing sample complexity, the N at which a reward or value error threshold is reached is computed by averaging over all values of N where the reward/value error crosses above/below the threshold.
- Experiments were run with a maximum of 100M environment steps. We applied early stopping with a test reward threshold of 0.95 (0.75 for `RubiksCube222`) and average value error threshold of 0.025 (0.1 for `RubiksCube222`).
- The horizon is 50 for all environments, including skill-augmented environments. In addition, to simulate a cost of applying too many base actions, we terminate an episode whenever the number of base actions reaches 100.
- For Q-learning, value iteration, and DQN, the replay buffer size is 1000 and updates are performed once every 4 episodes with a batch size of 32.
- Details on the model architecture of DQN are given in Appendix C.3.

No extensive hyperparameter tuning was done as the purpose of our experiments was not to compare RL algorithms, but to compare the performance of one algorithm on different action space variants of the same base environment.

C.2. Computational Resources

Experiments were run on 28 NVIDIA GPUs ($8\times$Quadro RTX 5000, $8\times$GeForce GTX 1080 Ti, $8\times$Tesla V100 SXM2 32GB, $4\times$RTX 6000 Ada Generation). One experiment, which usually consisted of 32 runs of some RL algorithm on different macroaction augmentations of the same base environment, took between under a minute to about a week to finish. In total, all experiments were completed within one month.

C.3. Algorithm-Specific Details

- Value iteration is modified to the RL setting in a way similar to Deep Approximate Value Iteration (DAVI) (Agostinelli et al., 2019). In DAVI, a state is chosen from some initial distribution and the value network is updated by minimizing the quadratic loss between the current state value and the Bellman update, thus requiring a forward pass that computes the values of all next states. In our version of value iteration, a state is chosen from the initial distribution and we apply a rollout of the epsilon-greedy policy. For each state s in the rollout, we also compute all possible next states. Similar to Q-learning, these next states are stored along with s in a replay buffer. When we sample a state (along with its next states) from the replay buffer, its value is updated in the direction of the Bellman update. Note that the fact that all possible next states are computed from each state in a rollout multiplies the number of environment steps taken by $|A|$.

- The policy $\pi_\theta(a \mid s)$ in REINFORCE is parameterized directly by the logits. In other words, the weights are an $|S| \times |A|$ matrix and $\pi_\theta(\cdot \mid s) = \text{Softmax}(\theta_{s,\cdot})$.

- The implementation of the deep neural net in DQN depends on the environment. A state embedding is first constructed from the input before passed into a linear projection head that outputs the action values $Q(s, \cdot)$ of a state s.

  - In `CliffWalking`, the input is a length-3 multihot vector at every location of the 4-by-12 grid (hence a $4 \times 12 \times 3$ binary tensor). In each multihot vector, the 3 indices represent the player, goal, and cliff. The state embedding is constructed by passing the input through a 2-layer CNN with ReLU activation followed by a 2-layer MLP with ReLU activation. The CNN has a kernel size of 3 and padding of 1. The hidden dimension is 32 and the output embedding has dimension 16.
  - In `CompILE2`, the input has two components. The grid is represented as a length-12 multihot vector at every location of the 10-by-10 grid (hence a $10 \times 10 \times 12$ binary tensor). The 12 indices of each multihot vector represent the 10 types of objects, wall, and agent. The next object the agent has to pick up is represented as a length-10 one-hot vector. The grid is passed through a 2-layer CNN with ReLU activation followed by a 2-layer MLP with ReLU activation. The result is concatenated with an embedding of the next object the agent has to pick up and passed through a linear projection to form the final embedding of the observation. The CNN has a a kernel size of 3 and padding of 0. The hidden dimension is 32 in the CNN layers and 128 in the MLP layers; the object embedding has dimension 16; the output embedding has dimension 128.
  - In `8Puzzle`, the input is a length-9 onehot vector at every location of the 3-by-3 grid (hence a $3 \times 3 \times 9$ binary tensor) denoting the tile present at each location (or the absence thereof). The state embedding is constructed by passing the input through a 2-layer CNN with ReLU activation followed by a 2-layer MLP with ReLU activation. The CNN has a kernel size of 3 and padding of 1. The hidden dimension is 32 and the output dimension is 32.
  - In `RubiksCube222`, the input is a length-6 multihot vector for each of $6 \times 4 = 24$ tiles of the cube (hence a 24×6 binary tensor) denoting the color of each tile. The state embedding is constructed by flattening the input and passing it through a 4-layer MLP with ReLU activation. The hidden dimension is 64 and the output dimension is 32.

D. Additional Empirical Tests of p-Learning and p-Exploration Difficulty

D.1. Empirically Verifying Lemma 3.1 for Motivating p-Learning Difficulty

To test how well p-learning difficulty captures learning from experience, we study the value iteration algorithm for planning with known transitions and rewards in a DSMDP. We consider two variants of value iteration: state value iteration for learning the values of states (Bellman, 1957), and action value iteration for learning the values of state-action pairs. The latter is like Q-learning (Watkins, 1989) but modified to update the values of all state-action pairs at once. Instead of the original Bellman update, each update uses a linear interpolation between the old value and the new value given by the Bellman update with a learning rate of $\alpha = 0.1$ (see Equation (3)).

For each base environment, we test the correlation between average solution length and sample complexity N on 32 macroaction augmentations of that environment. The results are summarized in Table 4. We find that the correlation between convergence time and average solution length is almost always greater than 0.9, with it occasionally being near-perfect (above 0.99).

Table 4. Across 32 macroaction augmentations of each of 4 base environments, we report the correlations between: the number of iterations until convergence (N) for two variants of value iteration (state values and Q-values) and two convergence criteria ($r \geq 0.95$; $\overline{\Delta V}$ or $\overline{\Delta Q} \leq 0.01$); and the p-weighted mean solution length of a state ($\overline{d} := \mathbb{E}_{s\sim p}[d_+(s)]$). The reported errors are standard errors of the mean over 5 seeds.

		$\overline{d}_{\texttt{CliffWalking}}$	$\overline{d}_{\texttt{CompILE2}}$	$\overline{d}_{\texttt{8Puzzle}}$	$\overline{d}_{\texttt{RubiksCube222}}$
Q-value iteration	$N_{r\geq 0.95}$	0.980 ± 0.001	0.934 ± 0.012	0.901 ± 0.013	0.942 ± 0.007
	$N_{\overline{\Delta Q}\leq 0.01}$	0.998 ± 0.000	0.977 ± 0.003	0.968 ± 0.006	0.989 ± 0.001
Value iteration	$N_{r\geq 0.95}$	0.977 ± 0.001	0.942 ± 0.005	0.902 ± 0.015	0.942 ± 0.007
	$N_{\overline{\Delta V}\leq 0.01}$	0.998 ± 0.000	0.984 ± 0.002	0.969 ± 0.005	0.985 ± 0.001

Table 5. Version of Table 1 where the geometric mean is replaced with the arithmetic mean in the definition of J_{explore}. With 3 exceptions, all correlation values are no higher than those when the geometric mean are used (Table 1).

		$\log J_{\texttt{CliffWalking}}$	$\log J_{\texttt{CompILE2}}$	$\log J_{\texttt{8Puzzle}}$	$\log J_{\texttt{RubiksCube222}}$
Q-Learning	$\log N_{r\geq r^*}$	0.947 ± 0.006	0.661 ± 0.049	0.301 ± 0.047	0.366 ± 0.081
	$\log N_{\overline{\Delta Q}\leq \Delta Q^*}$	0.953 ± 0.008	0.631 ± 0.061	0.442 ± 0.043	0.763 ± 0.019
Value iteration	$\log N_{r\geq r^*}$	0.933 ± 0.009	0.724 ± 0.043	0.788 ± 0.042	0.247 ± 0.058
	$\log N_{\overline{\Delta V}\leq \Delta V^*}$	0.951 ± 0.015	0.732 ± 0.035	0.877 ± 0.011	0.694 ± 0.021
REINFORCE	$\log N_{r\geq r^*}$	0.949 ± 0.006	0.732 ± 0.039	0.715 ± 0.020	0.537 ± 0.139
DQN	$\log N_{r\geq r^*}$	0.789 ± 0.028	0.752 ± 0.075	0.621 ± 0.025	0.576 ± 0.023

D.2. Arithmetic Mean Variant of p-Exploration Difficulty Performs Worse Than the Geometric Mean

Table 5 shows the version of Table 1 where $J_{\text{explore}} = \log N_{\text{GM}}$ is redefined to be $\log N_{\text{AM}}$. (Up to a constant factor, $N_{\text{AM}} = \mathbb{E}_{s\sim p}[1/q(s)]$ estimates an upper bound on the sample complexity of the exploration stage of RL.) Comparing the results with Table 1, we find that with 3 exceptions (in `8Puzzle`), all correlation values are no higher than those when the geometric mean is used.[5] This provides empirical validation for using the geometric mean as opposed to the arithmetic mean in our definition of p-exploration difficulty.

E. Proofs

Proof of Lemma 3.1. (*Note:* This proof assumes $\log$ refers to the natural logarithm.)

For $\alpha = 1$, simple induction on t shows that, at time t, the states with value 1 are exactly those states that can be solved with t actions or less, and all other states have value 0.

For the $\alpha < 1$ case, let's first consider the case where the DSMDP is a chain of states $0, 1, \ldots, n$ where state 0 is the only goal state and $T(s, a) = s - 1$ for any action a and non-goal state $s \neq 0$. Then the value iteration formula becomes $V(s) \leftarrow (1 - \alpha)V(s) + \alpha V(s - 1)$ for $s > 0$ and $V(0) = 1$. For $\alpha \ll 1$, we can write this as a differential equation

$$\frac{dV_s}{dt} = -\alpha(V_s - V_{s-1})$$

for $s > 0$, and $\frac{dV_0}{dt} = 0$. (We have switched to subscript notation to make it clearer that this is a linear system of ODEs in time.) Solving the system with the initial conditions $V_0(0) = 1$ and $V_s(0) = 0$ for $s > 0$ yields

$$V_s(t) = 1 - e^{-\alpha t} \sum_{k=0}^{s-1} \frac{(\alpha t)^k}{k!}.$$

Note that $V_s(t)$ decreases in s, i.e., at any time t, states closer to the goal have higher value.

[5] The correlation values of `CliffWalking` are exactly equal across the two tables because this environment has only one possible starting state, as a result of which the arithmetic and geometric means are exactly equal.

If $\alpha t = as + b\log(1/\varepsilon)$ where $a > 1$ and $b = \frac{a}{a-1}$, then

$$\begin{aligned}
\log(1 - V_s(t)) &\leq -\alpha t + \log\left(s\frac{(\alpha t)^s}{s!}\right) \\
&\leq -as - b\log(1/\varepsilon) + s\log(as + b\log(1/\varepsilon)) - (s-1)(\log(s-1) - 1) \\
&\leq -as - b\log(1/\varepsilon) + s\log s + s\log a + \frac{b}{a}\log(1/\varepsilon) - (s-1)(\log(s-1) - 1) \\
&= -s(a - \log a - 1) - \left(b - \frac{b}{a}\right)\log(1/\varepsilon) + s(\log s - \log(s-1)) + \log(s-1) - 1 \\
&= \log\varepsilon - s\left(a - \log a - 1 + \log\left(\frac{s-1}{s}\right) - \frac{\log(s-1) - 1}{s}\right),
\end{aligned}$$

which is less than $\log\varepsilon$ for sufficiently large s since $a - \log a - 1 > 0$.

Let $\alpha t = s + \log(1/\varepsilon) - 1 - \log 2$. Then for $s \geq 2$ and $\varepsilon \leq 1/2$, we have

$$\begin{aligned}
\log(1 - V_s(t)) &\geq -s - \log(1/\varepsilon) + 1 + \log 2 + \log\left(\sum_{k=0}^{s-1}\frac{(s-1)^k}{k!}\right) \\
&\overset{(*)}{\geq} -s - \log(1/\varepsilon) + 1 + \log 2 + \log\left(\frac{1}{2}e^{s-1}\right) \\
&= \log\varepsilon,
\end{aligned}$$

where the inequality marked (*) made use of the fact that the the median of a Poisson distribution with positive integer rate $s-1$ is exactly $s-1$ (Choi, 1994).

We have thus shown that we need $\alpha t = \Theta(s + \log(1/\varepsilon))$ to obtain $1 - V_s(t) = \varepsilon$. In other words, the time until the value estimate $V_s(t)$ is within ε of its true value of 1 is

$$t = \Theta\left(\frac{s + \log(1/\varepsilon)}{\alpha}\right). \tag{24}$$

Now let's return to the general graph setting. In this situation, the invariants are as follows:

- $\max_a V(T(s,a),t) = V(n(s),t)$ where $n(s)$ is the next state on the shortest path from s to any goal.
- $V(s,t) = V_{d(s)}(t)$ where $V_d(t)$ is the solution to the value function in the case of a simple chain, as we just derived.

This invariants are preserved by the fact that $V_d(t)$ is non-increasing in d. Thus, replacing s with $d(s)$ in the formula for the chain DSMDP (Equation (24)) yields the result for the general DSMDP case. □

Proof of Theorem 4.2. (*Note:* We use the version of the geometric distribution with support excluding 0.)

For $\sigma \in (A_+)^+$, let $P_{+,\text{unif},\varepsilon}(\sigma) = \varepsilon(1-\varepsilon)^{|\sigma|-1}|A_+|^{-|\sigma|}$, the probability that a random sequence of length $\sim \text{Geometric}(\varepsilon)$ with actions chosen uniformly from A_+ is exactly σ. Then $P_{+,\text{unif},\varepsilon}$ is a probability distribution over $(A_+)^+$, so

$$\mathbb{E}_{\sigma\sim P_+}[-\log P_{+,\text{unif},\varepsilon}(\sigma)] = \mathrm{H}[P_+, P_{+,\text{unif},\varepsilon}] \geq \mathrm{H}[P_+],$$

where $\mathrm{H}[p,q]$ denotes the cross entropy between p and q.

Now, fix any $0 < \varepsilon < 1$. Then

$$\begin{aligned}
\mathbb{E}_{s\sim p}[d_+(s)]\log\left(\frac{|A_+|}{1-\varepsilon}\right) + \log\left(\frac{1-\varepsilon}{\varepsilon}\right) &= \mathbb{E}_{\sigma\sim P_+}[|\sigma|]\log\left(\frac{|A_+|}{1-\varepsilon}\right) + \log\left(\frac{1-\varepsilon}{\varepsilon}\right) \\
&= \mathbb{E}_{\sigma\sim P_+}[-\log P_{+,\text{unif},\varepsilon}(\sigma)] \\
&\geq \mathrm{H}[P_+].
\end{aligned}$$

Thus,

$$\frac{\mathbb{E}_{s\sim p}[d_+(s)]\log\left(\frac{|A_+|}{1-\varepsilon}\right)}{\mathbb{E}_{s\sim p}[d_0(s)]\log\left(\frac{|A_0|}{1-\varepsilon}\right)} \geq \frac{\mathrm{H}[P_+]-\log\left(\frac{1-\varepsilon}{\varepsilon}\right)}{\mathbb{E}_{s\sim p}[d_0(s)]\log\left(\frac{|A_0|}{1-\varepsilon}\right)}$$

$$\begin{aligned}\frac{J_{\text{learn}}(\mathcal{M}_+;p)}{J_{\text{learn}}(\mathcal{M}_0;p)} = \frac{\mathbb{E}_{s\sim p}[d_+(s)]|A_+|}{\mathbb{E}_{s\sim p}[d_0(s)]|A_0|} &\geq \frac{|A_+|\log\left(\frac{|A_0|}{1-\varepsilon}\right)}{|A_0|\log\left(\frac{|A_+|}{1-\varepsilon}\right)}\frac{\mathrm{H}[P_+]-\log\left(\frac{1-\varepsilon}{\varepsilon}\right)}{\mathbb{E}_{s\sim p}[d_0(s)]\log\left(\frac{|A_0|}{1-\varepsilon}\right)} \\ &\geq \frac{|A_+|\log|A_0|}{|A_0|\log|A_+|}\frac{\mathrm{H}[P_+]-\log\left(\frac{1-\varepsilon}{\varepsilon}\right)}{\mathbb{E}_{s\sim p}[d_0(s)]\log\left(\frac{|A_0|}{1-\varepsilon}\right)}.\end{aligned}$$

The last inequality used the fact that $|A_+| \geq |A_0|$ gives

$$\frac{\log\left(\frac{|A_0|}{1-\varepsilon}\right)}{\log\left(\frac{|A_+|}{1-\varepsilon}\right)} = 1 - \frac{\log\left(\frac{|A_+|}{|A_0|}\right)}{\log\left(\frac{|A_+|}{1-\varepsilon}\right)} \geq 1 - \frac{\log\left(\frac{|A_+|}{|A_0|}\right)}{\log|A_+|} = \frac{\log|A_0|}{\log|A_+|}.$$

Now, we have

$$\frac{J_{\text{learn}}(\mathcal{M}_+;p)}{J_{\text{learn}}(\mathcal{M}_0;p)} \geq \sup_{0<\varepsilon<1}\frac{|A_+|\log|A_0|}{|A_0|\log|A_+|}\frac{\mathrm{H}[P_+]-\log\left(\frac{1-\varepsilon}{\varepsilon}\right)}{\mathbb{E}_{s\sim p}[d_0(s)]\log\left(\frac{|A_0|}{1-\varepsilon}\right)} = \frac{|A_+|\log|A_0|}{|A_0|\log|A_+|}\mathrm{IC}_{A_+}(\mathcal{M}_0;p),$$

which completes the proof. □

Proof of Corollary 4.5. Since $|A_0| \geq 2$, we have $|A_+| \geq |A_0|+1 \geq 3$. The function $f(x) = \ln x/x$ is decreasing for $x \geq e$, so

$$\begin{aligned}\frac{|A_0|\ln|A_+|}{|A_+|\ln|A_0|} = \frac{f(|A_+|)}{f(|A_0|)} &\leq \frac{f(|A_0|+1)}{f(|A_0|)} = \frac{|A_0|\ln(|A_0|+1)}{(|A_0|+1)\ln|A_0|} \\ &= \frac{|A_0|}{|A_0|+1}\left(1+\frac{\ln\left(1+\frac{1}{|A_0|}\right)}{\ln|A_0|}\right) < \frac{|A_0|}{|A_0|+1}\left(1+\frac{1}{|A_0|\ln|A_0|}\right) = \frac{1}{|A_0|+1}\left(|A_0|+\frac{1}{\ln|A_0|}\right).\end{aligned}$$

Then

$$\frac{J_{\text{learn}}(\mathcal{M}_+;p)}{J_{\text{learn}}(\mathcal{M}_0;p)} = \frac{|A_+|\ln|A_0|}{|A_0|\ln|A_+|}\mathrm{IC}(\mathcal{M}_0;p) > \frac{1-\frac{1}{|A_0|+1}\left(1-\frac{1}{\ln|A_0|}\right)}{\frac{1}{|A_0|+1}\left(|A_0|+\frac{1}{\ln|A_0|}\right)} = 1,$$

as desired. □

Proof of Theorem 5.2.

$$\begin{aligned}J_{\text{explore}}(\mathcal{M}_+;p,\delta) &= \mathbb{E}_{s\sim p}[-\log q_{+,\delta}(s)] \\ &= \mathbb{E}_{s\sim p}[-\log\rho_{+,\delta}(s)] - \log\left(\frac{1-\delta}{\delta}\right) \\ &= \mathbb{E}_{s\sim p}\left[-\log\left(\frac{\rho_{+,\delta}(s)}{D(\mathcal{M}_+;\delta)}\right)\right] - \log\left(\frac{1-\delta}{\delta}D(\mathcal{M}_+;\delta)\right) \\ &= \mathrm{H}[p] + D_{\mathrm{KL}}\left(p \parallel \frac{\rho_{+,\delta}(\cdot)}{D(\mathcal{M}_+;\delta)}\right) - \log\left(\frac{1-\delta}{\delta}D(\mathcal{M}_+;\delta)\right) \\ &\geq \mathrm{H}[p] - \log\left(\frac{1-\delta}{\delta}D(\mathcal{M}_+;\delta)\right),\end{aligned} \tag{25}$$

where we have used the fact that $\frac{\rho_{+,\delta}(\cdot)}{D(\mathcal{M}_+;\delta)}$ is a normalized probability distribution.

Now, suppose the state space is finite and $\delta > \max_s p(s)$. According to Equation (25), we want to show that we can make $D_{\mathrm{KL}}\left(p \parallel \frac{\rho_{+,\delta}(\cdot)}{D(\mathcal{M}_+;\delta)}\right)$ arbitrarily small with a suitable choice of A_+. Construct A_+ as follows. Let the number of skills $|A_+| - |A_0|$ be some large number $K \gg \max\{|A_0|, 1/\min_{s:p(s)>0} p(s)\}$. For each solvable state s with $p(s) > 0$, let $\lfloor Kf(s) \rfloor$ skills send s directly to the goal state and the remaining $K - \lfloor Kf(s) \rfloor$ send s back to s itself, where $f(s) = \frac{\delta}{\delta-(1-\delta)p(s)} p(s) \in (0,1)$. (For solvable states s with $p(s) = 0$, simply let all K skills send s back to s itself.) Let's now show that $\rho_{+,\delta}(s) \to p(s)$ as $K \to \infty$ for every solvable state s.

$\rho_{+,\delta}(s)$ is the probability that an action sequence σ with actions uniformly chosen from A_+ and length $|\sigma| \sim \mathrm{Geometric}(\delta)$ solves s. Among all such action sequences, the total probability of those that have a base action is no more than the total probability of all actions sequences that have a base action. The latter is given by

$$\begin{aligned}
1 - \sum_{\sigma \in (A_+ \setminus A_0)^+} \delta(1-\delta)^{|\sigma|-1} |A_+|^{-|\sigma|} &= 1 - \sum_{l=1}^{\infty} (|A_+| - |A_0|)^l \delta(1-\delta)^{l-1} |A_+|^{-l} \\
&= 1 - \frac{\delta}{1-\delta} \sum_{l=1}^{\infty} \left((1-\delta)\left(1 - \frac{|A_0|}{|A_+|}\right)\right)^l \\
&= 1 - \frac{\delta\left(1 - \frac{|A_0|}{|A_+|}\right)}{1 - (1-\delta)\left(1 - \frac{|A_0|}{|A_+|}\right)} \\
&\to 0, \quad \text{as } |A_0|/|A_+| \to 0.
\end{aligned}$$

It now remains to show that the total probability of solutions to s that consist only of skills approximates $p(s)$ arbitrarily well as $K \to \infty$. For s with $p(s) = 0$, no such solutions exist and so their total probability is 0. For s with $p(s) > 0$,

$$\begin{aligned}
\sum_{\sigma \in \mathrm{Sol}_+(s) \cap (A_+ \setminus A_0)^+} \delta(1-\delta)^{|\sigma|-1} |A_+|^{-|\sigma|} &= \sum_{l=1}^{\infty} \lfloor Kf(s) \rfloor (K - \lfloor Kf(s) \rfloor)^{l-1} \delta(1-\delta)^{l-1} |A_+|^{-l} \\
&= \frac{\delta \lfloor Kf(s) \rfloor}{|A_+|} \sum_{l=1}^{\infty} \left((1-\delta) \frac{K - \lfloor Kf(s) \rfloor}{|A_+|} \right)^{l-1} \\
&= \frac{\delta \lfloor Kf(s) \rfloor}{|A_+|} \frac{1}{1 - (1-\delta)\frac{K - \lfloor Kf(s) \rfloor}{|A_+|}} \\
&\to \delta f(s) \frac{1}{1 - (1-\delta)(1-f(s))} \qquad \text{(as } K \to \infty\text{)} \\
&= p(s).
\end{aligned}$$

By now, we have shown that $\rho_{+,\delta}(s) \to p(s)$ as $K \to \infty$ for every solvable state s. Since S is finite, this convergence is uniform, so the KL-divergence between p and the normalized version of $\rho_{+,\delta}$ tends to zero as $K \to \infty$, as desired. □

Proof of Corollary 5.3. Since $\mathcal{M}_0$ is solution-separable and $\mathcal{M}_+$ is a macroaction augmentation of $\mathcal{M}_0$, $\mathcal{M}_+$ is also solution-separable. Thus, $D(\mathcal{M}_+;\delta) \le 1$. By Theorem 5.2,

$$J_{\text{explore}}(\mathcal{M}_+; p, \delta) \ge \mathrm{H}[p] - \log\left(\frac{1-\delta}{\delta} D(\mathcal{M}_+;\delta)\right) \ge \mathrm{H}[p] - \log\left(\frac{1-\delta}{\delta}\right),$$

whereas

$$J_{\text{explore}}(\mathcal{M}_0; p, \delta) = \mathbb{E}_{s\sim p}[-\log q_{0,\delta}(s)] \le \mathbb{E}_{s\sim p}\left[-\log\left(\left(\frac{1-\delta}{|A_0|}\right)^{d_0(s)}\right)\right] = \mathbb{E}_{s\sim p}[d_0(s)] \log\left(\frac{|A_0|}{1-\delta}\right).$$

Thus,

$$\frac{J_{\text{explore}}(\mathcal{M}_+; p, \delta)}{J_{\text{explore}}(\mathcal{M}_0; p, \delta)} \ge \frac{\mathrm{H}[p] - \log\left(\frac{1-\delta}{\delta}\right)}{\mathbb{E}_{s\sim p}[d_0(s)] \log\left(\frac{|A_0|}{1-\delta}\right)},$$

as desired. □

Proof of Theorem 5.4. The construction given in the proof of Theorem 5.2 allows us to make $D_{\mathrm{KL}}\left(p \parallel \frac{\rho_{+,\delta}(s)}{D(\mathcal{M}_+;\delta)}\right)$ arbitrarily close to 0 and $D(\mathcal{M}_+;\delta) = \sum_s \rho_{+,\delta}(s)$ arbitrarily close to 1 with sufficient large $K = |A_+| - |A_0|$. Recalling Equation (25), this means that for any $\varepsilon > 0$, the construction gives

$$J_{\text{explore}}(\mathcal{M}_+; p, \delta) < \mathrm{H}[p] - \log\left(\frac{1-\delta}{\delta}\right) + \varepsilon$$

for sufficiently large K.

On the other hand, let $p', \rho'_{0,\delta}$ be distributions defined on solvable states in addition to a dummy state s_d such that $p'(s) = p(s)$ and $\rho'_{0,\delta}(s) = \rho_{0,\delta}(s)$ whenever $s \neq s_d$, whereas $p'(s_d) = 0$ and $\rho'_{0,\delta}(s_d) = 1 - \sum_{s \neq s_d} \rho_{0,\delta}(s)$. (Note that $\rho'_{0,\delta}(s_d) \geq 0$ because $\mathcal{M}_0$ is solution-separable.) Then $D_{\mathrm{KL}}\left(p' \parallel \rho'_{0,\delta}\right) > 0$ since $p' \not\equiv \rho'_{0,\delta}$. This gives

$$\begin{aligned}
J_{\text{explore}}(\mathcal{M}_0; p, \delta) &= \mathbb{E}_{s\sim p}\left[-\log \rho_{0,\delta}(s)\right] - \log\left(\frac{1-\delta}{\delta}\right) \\
&= \mathrm{H}[p] + \mathbb{E}_{s\sim p}\left[-\log \frac{\rho_{0,\delta}(s)}{p(s)}\right] - \log\left(\frac{1-\delta}{\delta}\right) \\
&= \mathrm{H}[p] + D_{\mathrm{KL}}\left(p' \parallel \rho'_{0,\delta}(s)\right) - \log\left(\frac{1-\delta}{\delta}\right) \\
&> \mathrm{H}[p] - \log\left(\frac{1-\delta}{\delta}\right).
\end{aligned} \tag{26}$$

As a result, for sufficiently large K, $J_{\text{explore}}(\mathcal{M}_+; p, \delta) < J_{\text{explore}}(\mathcal{M}_0; p, \delta)$.

Now, let's show that the construction in the proof of Theorem 5.2 can be made more precise to allow all states with $p(s) > 0$ to have distinct canonical shortest solutions in A_+. Simply choose K large enough so that, for all s with $p(s) > 0$, the number of skills $\lfloor Kf(s) \rfloor$ that send s directly to g is at least the number of states with $p(s) > 0$. Then the number of shortest solutions to every s with $p(s) > 0$ is at least the number of such s, so it is possible to choose one shortest solution for every such s so that all the chosen solutions are distinct. □

Proof of Corollary 5.5. Define A_+ as in Theorem 5.4, so that $J_{\text{explore}}(\mathcal{M}_+; p, \delta) < J_{\text{explore}}(\mathcal{M}_0; p, \delta)$ and Theorem 4.2 gives

$$\frac{J_{\text{learn}}(\mathcal{M}_+; p)}{J_{\text{learn}}(\mathcal{M}_0; p)} \geq \frac{|A_+| \log |A_0|}{|A_0| \log |A_+|} \mathrm{IC}(\mathcal{M}_0; p), \tag{27}$$

which is identical to Equation (12). Then the proof that the additional condition in the corollary implies $J_{\text{learn}}(\mathcal{M}_+; p) > J_{\text{learn}}(\mathcal{M}_0; p)$ is identical to the proof of Corollary 4.5. □

Proof of Theorem 5.6. Augment the state space of $\mathcal{M}_0$ with a state s_1 that is solved by every length-1 sequence that is not already the solution to any other state. (Furthermore, all actions that do not result in the goal state instead transition to a dead state.) Denote by $\bar{\mathcal{M}}_0$ the resultant DSMDP and for simplicity of notation we write $\bar{\rho}_{0,\delta}$ for $\rho_{\bar{\mathcal{M}}_0,\delta}$ and $\bar{d}_0$ for $d_{\bar{\mathcal{M}}_0}$. Let $\bar{\mathcal{M}}_+$ denote the A_+-macroaction augmentation of $\bar{\mathcal{M}}_0$. Then the solutions to s_1 in A_+ are exactly the same as those in A_0 since macroactions always have length greater than 1. We will write $\bar{\rho}_{+,\delta}$ to mean $\rho_{\bar{\mathcal{M}}_+,\delta}$. Let $\bar{p}$ be a distribution over the solvable states of $\bar{\mathcal{M}}_0$ so that $\bar{p} = p$ on the solvable states of $\mathcal{M}_0$ and $\bar{p}(s_1) = 0$.

As in the proof of Theorem 5.4, we define distributions $p', \bar{\rho}'_{0,\delta}, \bar{\rho}'_{+,\delta}$ over the solvable states in addition to a dummy state s_d to be equal to $\bar{p}, \bar{\rho}_{0,\delta}, \bar{\rho}_{+,\delta}$ whenever $s \neq s_d$, whereas $p'(s_d) = 0$ and $\bar{\rho}'_{0,\delta}(s_d), \bar{\rho}'_{+,\delta}(s_d)$ are such that $\bar{\rho}'_{0,\delta}, \bar{\rho}'_{+,\delta}$ are normalized probability distributions.

First, let's show that

$$\sum_s |\bar{\rho}'_{+,\delta}(s) - \bar{\rho}'_{0,\delta}(s)| \geq \frac{\delta}{|A_0| + 1}. \tag{28}$$

If s is distance 1 away from the goal in $\bar{\mathcal{M}}_0$, then

$$\bar{\rho}'_{+,\delta}(s) = \delta\frac{n(s)}{|A_+|}, \quad \bar{\rho}'_{0,\delta}(s) = \delta\frac{n(s)}{|A_0|},$$

where $n(s)$ denotes the number of solutions to s in $\bar{\mathcal{M}}_0$ (or equivalently, $\bar{\mathcal{M}}_+$), all of which have length 1. Thus,

$$\sum_s |\bar{\rho}'_{+,\delta}(s) - \bar{\rho}'_{0,\delta}(s)| \geq \sum_{s:\bar{d}_0(s)=1} \delta n(s)\left(\frac{1}{|A_0|} - \frac{1}{|A_+|}\right) = \delta|A_0|\left(\frac{1}{|A_0|} - \frac{1}{|A_+|}\right) = \delta\left(1 - \frac{|A_0|}{|A_+|}\right) \geq \frac{\delta}{|A_0|+1},$$

where the last inequality used the fact that $|A_+| \geq |A_0| + 1$.

We will now use Equation (28) to prove the theorem. By the triangle inequality,

$$\sum_s |\bar{\rho}'_{+,\delta}(s) - \bar{p}'(s)| \geq \sum_s |\bar{\rho}'_{+,\delta}(s) - \bar{\rho}'_{0,\delta}(s)| - \sum_s |\bar{\rho}'_{0,\delta}(s) - \bar{p}'(s)| \geq \frac{\delta}{|A_0|+1} - \sum_s |\bar{\rho}'_{0,\delta}(s) - \bar{p}'(s)|.$$

Pinsker's inequality says that $D_{\mathrm{KL}}(p \parallel q) \geq \frac{1}{2}\left(\sum_x |p(x) - q(x)|\right)^2 \log e$ for any two probability mass functions p, q. Thus, if

$$D_{\mathrm{KL}}\left(\bar{p}' \parallel \bar{\rho}'_{0,\delta}\right) = D_{\mathrm{KL}}\left(p' \parallel \rho'_{0,\delta}\right) < \frac{\delta^2 \log e}{8(|A_0|+1)^2},$$

then

$$\sum_s |\bar{\rho}'_{+,\delta}(s) - \bar{p}'(s)| > \frac{\delta}{|A_0|+1} - \sqrt{\frac{2}{\log e} \cdot \frac{\delta^2 \log e}{8(|A_0|+1)^2}} = \frac{\delta}{2(|A_0|+1)}$$

and so

$$D_{\mathrm{KL}}\left(p' \parallel \rho'_{+,\delta}\right) = D_{\mathrm{KL}}\left(\bar{p}' \parallel \bar{\rho}'_{+,\delta}\right) > \frac{1}{2}\left(\frac{\delta}{2(|A_0|+1)}\right)^2 \log e > D_{\mathrm{KL}}\left(p' \parallel \rho'_{0,\delta}\right).$$

Now, by Equation (26), this is equivalent to $J_{\text{explore}}(\mathcal{M}_+; p, \delta) > J_{\text{explore}}(\mathcal{M}_0; p, \delta)$, as desired. □

The proof of Theorem 5.7 is omitted as the stronger version and its proof are given in Appendix F.4.

F. Additional Theoretical Results

F.1. Preliminary Results on Stochastic Environments

Here, we provide preliminary generalizations of our results for stochastic sparse-reward MDPs, which are SDMDPs (Definition 2.1) where the transition kernel T may be stochastic (i.e., $T(s,a)$ is now a distribution over S).

In a (possibly stochastic) sparse-reward MDP, let $W_{\sigma s}$ be the probability that taking actions σ starting in s results in the goal state. For an ordering $\sigma_1, \sigma_2, \ldots$ of all positive-length action sequences in non-decreasing length, define

$$w_{\sigma_k s} = \begin{cases} W_{\sigma_k s} & 1 \leq k < k_{max} \\ 1 - \sum_{i=1}^{k_{max}-1} W_{\sigma_i s} & k = k_{max} \\ 0 & k > k_{max} \end{cases}$$

where k_{max} is the largest k such that $\sum_{i=1}^{k-1} W_{\sigma_i s} < 1$. As a result, $\sum_\sigma w_{\sigma s} = 1$.

Let's redefine $d_{\mathcal{M}}(s)$ to be the weighted mean $\sum_\sigma w_{\sigma s}|\sigma|$, so that p-learning difficulty (Equation (4)) and A_+-merged p-incompressibility (Equation (7)) are now defined using this new notion of shortest solution length. Furthermore, in the definition of A_+-merged p-incompressibility (Equation (7)), redefine P_+ to be $P_+(\sigma) = \sum_s p(s) w_{\sigma s}$ so that $\sum_\sigma P_+(\sigma) = 1$. Note that the new definitions match the old definitions when the environment is deterministic. The stochasticity effectively spreads the responsibility of being a "shortest solution" over several short solutions whose success probabilities $W_{\sigma s}$ add up to 1.

Theorem F.1 (Generalization of Theorem 4.2)**.** *Under the above redefinitions for stochastic sparse-reward MDPs, Equation (9) of Theorem 4.2 continues to hold.*

Proof. The proof is identical to that of the original Theorem 4.2. □

In stochastic environments, we can keep the original definition of p-exploration difficulty (Equation (5)) since the probabilistic definition of $q_{\mathcal{M},\delta}(s)$ continues to make sense when there's stochasticity. (As a reminder, it is the probability that a uniformly random policy that terminates with probability δ before each step solves s.) Similarly, we keep the definition of δ-discounted solution density (Definition 5.1), which is also defined in terms of q.

Theorem F.2 (Generalization of the first half of Theorem 5.2)**.** *Under the above redefinitions for stochastic sparse-reward MDPs, Equation* (15) *of Theorem 5.2 continues to hold.*

Proof. The proof is identical to that of the original Theorem 5.2. □

F.2. Incorporating Skill Expressivity in Theorem 4.2

In Theorem F.5 below, we provide a version of Theorem 4.2 that eliminates the dependence of $\mathrm{IC}_{A_+}(\mathcal{M}_0; p)$ on A_+ and makes it depend explicitly on a quantitative measure of skill expressivity instead. This new measure of incompressibility (Equation (29)), which we call E-expressive p-incompressibility, decreases in E. This is expected as an environment is more compressible when the available skills are more expressive.

Definition F.3 (Quantifying skill expressivity)**.** With respect to a DSMDP $\mathcal{M} = (S, A, T, g)$, define the *behavior variety expressivity* E_z of a skill $z : S \to A^*$ to be $|z(S)|$, i.e., the number of distinct action sequences that z can produce.

Definition F.4 (E-expressive p-incompressibility)**.** For a DSMDP $\mathcal{M} = (S, A, T, g)$ with finite $|A| > 1$, define its *E-expressive p-incompressibility* to be

$$\mathrm{IC}(\mathcal{M}; p, E) = \sup_{0<\varepsilon<1} \frac{\min_P \mathrm{H}[P] - \log\left(\frac{1-\varepsilon}{\varepsilon}\right)}{\mathbb{E}_{s\sim p}[d_{\mathcal{M}}(s)] \log\left(\frac{|A|E}{1-\varepsilon}\right)} \tag{29}$$

where the $\min_P$ is taken over all choices of canonical (not necessarily shortest) solutions to all states.[6] Note that expressivity E occurs once in the denominator, so that larger E results in smaller $\mathrm{IC}(\mathcal{M}; p, E)$.

Theorem F.5 (Expressivity and p-learning difficulty improvability)**.** *Assuming the setup to Theorem 4.2, the following modified version of Equation* (9) *holds:*

$$\frac{J_{learn}(\mathcal{M}_+; p)}{J_{learn}(\mathcal{M}_0; p)} \geq \frac{|A_+| \log |A_0|}{|A_0| \log |A_+|} \mathrm{IC}(\mathcal{M}_0; p, E) \tag{30}$$

where $E := \max_{z\in A_+\setminus A_0} E_z$ is the maximum behavior variety expressivity of a skill in the skill augmentation. Higher expressivity E thus reduces incompressibility and allows skills to improve p-learning difficulty more, as expected.

Proof. Given any choice of canonical shortest solutions in A_+, define the random variables $\sigma_+ \in (A_+)^+$ and $\sigma_0 \in (A_0)^+$ as follows. For $s \sim p$, σ_+ is the canonical solution to s in A_+, and σ_0 is the same solution but with skills expanded into base actions. Then the distribution of σ_+ is just P_+, and let P_0 be the distribution of σ_0.

Note that

$$\mathrm{H}[P_+] + \mathrm{H}[\sigma_0|\sigma_+] = \mathrm{H}[(\sigma_+, \sigma_0)] \geq \mathrm{H}[P_0]. \tag{31}$$

Furthermore, since any σ_+ can expand to at most $E^{|\sigma_+|}$ different base action sequences,

$$\mathrm{H}[\sigma_0|\sigma_+] \leq \mathbb{E}_{\sigma_+\sim P_+}[|\sigma_+| \log E]. \tag{32}$$

In addition, recall from the proof of Theorem 4.2 that, for any $0 < \varepsilon < 1$,

$$\mathbb{E}_{s\sim p}[d_+(s)] \log\left(\frac{|A_+|}{1-\varepsilon}\right) + \log\left(\frac{1-\varepsilon}{\varepsilon}\right) \geq \mathrm{H}[P_+]. \tag{33}$$

[6] Recall that, given a choice of canonical solutions to all states, $P(\sigma)$ is the sum over $p(s)$ of all states s that have σ as their canonical solution. As a result, $\mathrm{H}[P] \leq \mathrm{H}[p]$ and equality holds in solution-separable DSMDPs.

Thus, substituting Equations (32) and (33) into Equation (31) yields

$$\begin{aligned}
\mathbb{E}_{s\sim p}[d_+(s)]\log\left(\frac{|A_+|}{1-\varepsilon}\right)+\log\left(\frac{1-\varepsilon}{\varepsilon}\right)+\mathbb{E}_{\sigma_+\sim P_+}[|\sigma_+|]\log E &\geq \mathrm{H}[P_0]\\
\mathbb{E}_{s\sim p}[d_+(s)]\log\left(\frac{|A_+|}{1-\varepsilon}\right)+\mathbb{E}_{s\sim p}[d_+(s)]\log E &\geq \mathrm{H}[P_0]-\log\left(\frac{1-\varepsilon}{\varepsilon}\right)\\
\mathbb{E}_{s\sim p}[d_+(s)]\log\left(\frac{|A_+|E}{1-\varepsilon}\right) &\geq \mathrm{H}[P_0]-\log\left(\frac{1-\varepsilon}{\varepsilon}\right)\\
\frac{\mathbb{E}_{s\sim p}[d_+(s)]\log\left(\frac{|A_+|E}{1-\varepsilon}\right)}{\mathbb{E}_{s\sim p}[d_0(s)]\log\left(\frac{|A_0|E}{1-\varepsilon}\right)} &\geq \frac{\min_{P_0}\mathrm{H}[P_0]-\log\left(\frac{1-\varepsilon}{\varepsilon}\right)}{\mathbb{E}_{s\sim p}[d_0(s)]\log\left(\frac{|A_0|E}{1-\varepsilon}\right)}\\
\frac{J_{\text{learn}}(\mathcal{M}_+;p)}{J_{\text{learn}}(\mathcal{M}_0;p)} &\geq \frac{|A_+|\log\left(\frac{|A_0|E}{1-\varepsilon}\right)}{|A_0|\log\left(\frac{|A_+|E}{1-\varepsilon}\right)}\frac{\min_{P_0}\mathrm{H}[P_0]-\log\left(\frac{1-\varepsilon}{\varepsilon}\right)}{\mathbb{E}_{s\sim p}[d_0(s)]\log\left(\frac{|A_0|E}{1-\varepsilon}\right)}
\end{aligned}$$

where

$$\frac{\log\left(\frac{|A_0|E}{1-\varepsilon}\right)}{\log\left(\frac{|A_+|E}{1-\varepsilon}\right)}\geq\frac{\log|A_0|}{\log|A_+|}$$

since $\frac{E}{1-\varepsilon}>1$ and $|A_+|\geq|A_0|$. Thus,

$$\frac{J_{\text{learn}}(\mathcal{M}_A;p)}{J_{\text{learn}}(\mathcal{M}_B;p)}\geq\frac{|A_+|\log|A_0|}{|A_0|\log|A_+|}\frac{\min_{P_0}\mathrm{H}[P_0]-\log\left(\frac{1-\varepsilon}{\varepsilon}\right)}{\mathbb{E}_{s\sim p}[d_0(s)]\log\left(\frac{|A_0|E}{1-\varepsilon}\right)},$$

which is true for all $0<\varepsilon<1$, as desired. □

F.3. Relaxing Solution-Separability Assumption in Corollary 4.4

Corollary F.6 (Generalization of Corollary 4.4)**.** *Relaxing the solution-separability assumption, Corollary 4.4 holds if we replace* $\mathrm{H}[p]$ *in the definition of* $\mathrm{IC}(\mathcal{M}_0;p)$ *with* $\min_{P_0}\mathrm{H}[P_0]$*. Here,* P_0 *is the distribution of canonical solutions to states sampled from* p*, and the minimum is taken over all possible choices of canonical solutions. Thus,* $\mathrm{H}[P_0]$ *can be understood as the entropy of the state distribution if states with the same canonical solution are merged into one "super-state."*

Proof. The result follows directly from Theorem F.5 by setting $E=1$. □

F.4. Stronger Version of Theorem 5.7

Note: For notational simplicity, we will write $q_{\mathcal{M}}$ to mean $q_{\mathcal{M},\delta=0}$.

Before stating the stronger version of Theorem 5.7, we need to first define *solution-length separations* of state spaces.

Definition F.7. For a DSMDP $\mathcal{M}=(S,A,T,g)$, let S_{solvable} denote the set of solvable states. The *solution-length separation* $\tilde{S}_{\text{solvable}}$ of S_{solvable} is the result of separating every solvable state $s\in S_{\text{solvable}}$ into a set $\tilde{S}(s)$ of *sub-states* corresponding to the lengths of solutions to s. Formally, we write

$$\tilde{S}_{\text{solvable}}:=\bigcup_{s\in S_{\text{solvable}}}\tilde{S}(s),\quad \tilde{S}(s):=\{(s,l)\mid l>0\text{ s.t. }\exists\sigma\in\mathrm{Sol}_{\mathcal{M}}(s)\text{ with }|\sigma|=l\}.$$

Furthermore, for a sub-state $\tilde{s}=(s,l)$ of s corresponding to solution length l, we naturally define its solutions to be the length-l solutions to s. Formally,

$$\tilde{\mathrm{Sol}}_{\mathcal{M}}((s,l)):=\{\sigma\in\mathrm{Sol}_{\mathcal{M}}(s)\mid|\sigma|=l\}$$

where the $\tilde{}$ is used to make it explicit that we're applying the operation to sub-states.

Functions on S_{solvable} defined using solutions to states can therefore be naturally extended to $\tilde{S}$. For example, $\tilde{d}(\tilde{s})$ for $\tilde{s} = (s, l) \in \tilde{S}$ is just l, and

$$\tilde{q}_{\mathcal{M}}(\tilde{s}) := \sum_{\sigma \in \tilde{\text{Sol}}_{\mathcal{M}}(\tilde{s})} |A|^{-|\sigma|} = \left|\tilde{\text{Sol}}_{\mathcal{M}}(\tilde{s})\right| |A|^{-l} \quad \text{if } \tilde{s} = (s, l).$$

For an arbitrary function $f : S_{\text{solvable}} \to \mathbb{R}$, there is a family of natural extensions to $\tilde{S}_{\text{solvable}}$. Specifically, we say that $\tilde{f} : \tilde{S}_{\text{solvable}} \to \mathbb{R}$ is a *solution-length-separated additive extension* if, for all $s \in S_{\text{solvable}}$,

$$f(s) = \sum_{\tilde{s} \in \tilde{S}(s)} \tilde{f}(\tilde{s}),$$

and $f(s) = \tilde{f}(s)$ for $s \notin S_{\text{solvable}}$. For example, $\tilde{q}_{\mathcal{M}}$ as defined above is a solution-length-separated additive extension to $q_{\mathcal{M}}$.

Theorem F.8 (Generalization of Theorem 5.7)**.** *Let $\mathcal{M}_+ = (S, A_+, T_+, g)$ be the A_+-macroaction augmentation of the solution-separable DSMDP $\mathcal{M}_0 = (S, A_0, T_0, g)$ with a finite action space, and p a probability distribution over solvable states. Then there exists a solution-length-separated additive extension $\tilde{p}$ to p in $\mathcal{M}_0$ such that*

$$J_{\text{explore}}(\mathcal{M}_+; p) - J_{\text{explore}}(\mathcal{M}_0; p) \geq \frac{|A_0|}{|A_+|}\left(1 - \frac{|A_0|}{|A_+|}\right) - D_{\text{KL}}\left(\tilde{p} \parallel \tilde{\lambda}\tilde{q}_0\right). \tag{34}$$

Here, $(\tilde{\lambda}\tilde{q}_0)((s, l)) := \tilde{\lambda}(l)\tilde{q}_0((s, l))$, where

$$\tilde{\lambda}(l) := \sum_{\tilde{s}' : \tilde{d}_0(\tilde{s}') = l} \tilde{p}(\tilde{s}')$$

is the total probability (under $\tilde{p}$) of sub-states with solution length l and

$$\tilde{q}_0((s, l)) = \left|\tilde{\text{Sol}}_0((s, l))\right| |A_0|^{-l}$$

is the probability that a uniformly random action sequence of length l is a solution to s. To make $\tilde{\lambda}\tilde{q}_0$ a normalized probability distribution, we introduce a dummy sub-state $\tilde{s}_d(l)$ for each solution length l with $\tilde{p}(\tilde{s}_d(l)) := 0$ and $\tilde{q}_0(\tilde{s}_d(l)) := 1 - \sum_{s:(s,l) \in \tilde{S}_{\text{solvable}}} \tilde{q}_0((s, l))$. Note that $\tilde{q}_0(\tilde{s}_d(l)) \geq 0$ because of solution-separability, and it is equal to zero when every action sequence of length l is the solution to some state.

Proof of Theorem F.8. For each solvable state s, denote by $\tilde{S}(s)$ the set of sub-states resultant from separating s by solution length in the base environment. Define the solution-length-separated additive extension $\tilde{p}$ to p such that $\tilde{p}(\tilde{s}) \propto \tilde{q}_+(\tilde{s})$ for $\tilde{s} \in \tilde{S}(s)$, or more precisely,

$$\tilde{p}(\tilde{s}) = p(s)\frac{\tilde{q}_+(\tilde{s})}{q_+(s)}, \quad q_+(s) = \sum_{\tilde{s} \in \tilde{S}(s)} \tilde{q}_+(\tilde{s}).$$

Here,

$$\tilde{q}_+((s, l)) := \sum_{\substack{\sigma \in \text{Sol}_+(s) \\ \sigma \text{ expands to } l \text{ base actions}}} |A_+|^{-|\sigma|}$$

denotes the q of the A_+-macroaction augmentation of $\tilde{\mathcal{M}}_0$ (the solution-length separation of $\mathcal{M}_0$ with respect to A_0), and not the solution-length separation of $\mathcal{M}_+$ (the A_+-macroaction augmentation of $\mathcal{M}_0$).

Then

$$\log \frac{q_0(s)}{q_+(s)} = \log \sum_{\tilde{s} \in \tilde{S}(s)} \frac{\tilde{q}_+(\tilde{s})}{q_+(s)} \frac{\tilde{q}_0(\tilde{s})}{\tilde{q}_+(\tilde{s})} \geq \sum_{\tilde{s} \in \tilde{S}(s)} \frac{\tilde{q}_+(\tilde{s})}{q_+(s)} \log \frac{\tilde{q}_0(\tilde{s})}{\tilde{q}_+(\tilde{s})}$$

by Jensen's inequality, so

$$J_{\text{explore}}(\mathcal{M}_+; p) - J_{\text{explore}}(\mathcal{M}_0; p) = \mathbb{E}_{s \sim p}\left[\log \frac{q_0(s)}{q_+(s)}\right] \geq \mathbb{E}_{\tilde{s} \sim \tilde{p}}\left[\log \frac{\tilde{q}_0(\tilde{s})}{\tilde{q}_+(\tilde{s})}\right].$$

It thus suffices to lower-bound the latter. Let's consider base solution lengths l separately.

Fix some $l \geq 1$. Define $\tilde{p}_l$ to be the conditional distribution of $\tilde{p}$ on sub-states with solution length l. In other words, if $\tilde{S}_l$ denotes the set of sub-states with solution length l, then $\tilde{p}_l$ is a distribution over $\tilde{S}_l$ defined as

$$\tilde{p}_l(\tilde{s}) = \frac{\tilde{p}(\tilde{s})}{\tilde{\lambda}(l)}, \quad \tilde{\lambda}(l) = \sum_{\tilde{s}' \in \tilde{S}_l} \tilde{p}(\tilde{s}').$$

We write

$$\mathbb{E}_{\tilde{s}\sim\tilde{p}_l}\left[\log \frac{\tilde{q}_0(\tilde{s})}{\tilde{q}_+(\tilde{s})}\right] = \mathbb{E}_{\tilde{s}\sim\tilde{p}_l}\left[\log \frac{\tilde{q}_0(\tilde{s})}{\tilde{q}_+(\tilde{s})/\sum_{\tilde{s}'\in\tilde{S}_l^*}\tilde{q}_+(\tilde{s}')}\right] - \log \sum_{\tilde{s}'\in\tilde{S}_l^*} \tilde{q}_+(\tilde{s}'), \tag{35}$$

where $\tilde{S}_l^*$ denotes the set $\tilde{S}_l$ of sub-states with solution length l, along with a dummy state $\tilde{s}_d$ for every length-l action sequence that isn't a solution to any state. Note that $\tilde{q}_0(\tilde{s}_d) = |A_0|^{-l}$ for each dummy state so that $\sum_{\tilde{s}'\in\tilde{S}_l^*}\tilde{q}_0(\tilde{s}') = |A_0|^l|A_0|^{-l} = 1$, whereas $\tilde{q}_+(\tilde{s}_d) = \sum_{\sigma \in (A_+)^+ \text{ expands to } \alpha} |A_+|^{-|\sigma|}$ where α is the action sequence assigned as the solution to $\tilde{s}_d$. As usual for dummy states, we define $\tilde{p}_l(\tilde{s}_d) = 0$.

Let's first lower-bound the first term on the RHS of Equation (35):

$$\mathbb{E}_{\tilde{s}\sim\tilde{p}_l}\left[\log \frac{\tilde{q}_0(\tilde{s})}{\tilde{q}_+(\tilde{s})/\sum_{\tilde{s}'\in\tilde{S}_l^*}\tilde{q}_+(\tilde{s}')}\right] = D_{\mathrm{KL}}\left(\tilde{p}_l \parallel \frac{\tilde{q}_+(\cdot)}{\sum_{\tilde{s}'\in\tilde{S}_l^*}\tilde{q}_+(\tilde{s}')}\right) - D_{\mathrm{KL}}\left(\tilde{p}_l \parallel \tilde{q}_0\right) \geq -D_{\mathrm{KL}}\left(\tilde{p}_l \parallel \tilde{q}_0\right). \tag{36}$$

Let's now upper-bound the sum in the second term on the RHS of Equation (35). We write

$$\sum_{\tilde{s}'\in\tilde{S}_l^*} \tilde{q}_+(\tilde{s}') = \sum_{\substack{\sigma\in(A_+)^* \\ \sigma \text{ expands to } l \text{ base actions}}} |A_+|^{-|\sigma|}$$

which is a function $f_l(x_2, \ldots, x_K)$ where x_k is the number of macroactions of length k divided by $|A_+|$ and K is the maximum length of any macroaction. To see that this is a function of only l and x_k, notice that changing the number of macroactions of every length as well as the number of base actions by the same factor ξ (which keeps all x_k unchanged) will result in $\xi^{l'}$ times more sequences $\sigma \in (A_+)^*$ such that $|\sigma| = l'$ and σ expands to l base actions, whereas the $|A_+|^{-l'}$ summand is multiplied by a factor of $\xi^{-l'}$ for these sequences. The two factors cancel out, thus leaving the entire sum unchanged.

Now, let's derive a recursive formula for $f_l(x_2, \ldots, x_K)$ where the x_i are treated as parameters. To do this, we separate the sum over σ into cases depending on whether the first action in σ is a macroaction, and its length if yes. If the first action in σ is a base action, then the rest of σ expands to length $l-1$, so the contribution to the sum is $x_1 f_{l-1}(x_2, \ldots, x_K)$, where $x_1 := 1 - \sum_{k=2}^K x_k$ is the number of base actions divided by $|A_+|$. If the first action in σ is a macroaction of length k, then the rest of σ expands to length $l-k$, so the contribution to the sum is $x_k f_{l-k}(x_2, \ldots, x_K)$. To summarize,

$$f_l = \sum_{k=1}^K x_k f_{l-k},$$

where it is understood that $f_i = 0$ for $i < 0$. The base case is $f_0 = 1$. Since the sum of the coefficients x_k in the recursive formula equals 1, f_l is just a weighted average of $f_{l-1}, f_{l-2}, \ldots, f_{l-K}$. Thus, if $f_l \leq a$ for $1 \leq l \leq K$ then $f_l \leq a$ for all $l \geq 1$.

Let's show by induction on K that

$$f_l \leq 1 - x_1 + x_1^2 \tag{37}$$

for all $l \geq 1$. It suffices to show that $f_l \leq 1 - x_1 + x_1^2$ for $1 \leq l \leq K$.

For $K = 1$, $f_1 = x_1 = 1 = 1 - x_1 + x_1^2$. For $K = 2$, $f_1 = x_1 \leq x_1 + (1-x_1)^2 = 1 - x_1 + x_1^2$ and $f_2 = x_1 f_1 + (1 - x_1) = 1 - x_1 + x_1^2$.

Now, for $K \geq 3$, assume the $K-1$ and $K-2$ cases hold.

Let's upper-bound f_l for the following two cases separately: (i) $1 \le l \le K-1$; (ii) $l = K$.

(i) Define $x'_k = x_k/\bar{x}_K$ for $1 \le k \le K-1$ where $\bar{x}_K := 1 - x_K = \sum_{i=1}^{K-1} x_i$. Define the sequence $f'_l = \sum_{k=1}^{K-1} x'_k f'_{l-k}$ with $f'_i = 0$ for $i < 0$ and $f'_0 = 1$. Then the inductive hypothesis gives $f'_l \le 1 - x'_1 + x'^2_1$ for $1 \le l \le K-1$. It is easy to show by induction on l that $f_l = (\bar{x}_K)^l f'_l$ for $0 \le l \le K-1$, so for $1 \le l \le K-1$,

$$f_l \le \bar{x}_K(1 - x'_1 + x'^2_1) = \bar{x}_K - x_1 + \frac{x_1^2}{\bar{x}_K},$$

where $\bar{x}_K$ is restricted to the range $x_1 \le \bar{x}_K \le 1$. Since $\bar{x}_K + \frac{x_1^2}{\bar{x}_K}$ is increasing for $\bar{x}_K \ge x_1$, its maximum is reached when $\bar{x}_K = 1$, i.e.,

$$f_l \le \bar{x}_K - x_1 + \frac{x_1^2}{\bar{x}_K} \le 1 - x_1 + x_1^2.$$

(ii) Note that recursively expanding the recursion formula for f_l until we reach the base cases results in a polynomial in $x_1, \ldots, x_K$. It is easy to see by induction on l that, for $1 \le l \le K$, no term contains x_k where $k > l$ and there is a single term containing x_l which is just x_l. So $f_{K-1} = P_1(x_1, \ldots, x_{K-2}) + x_{K-1}$ for some polynomial P_1, and

$$\begin{aligned} f_K &= \sum_{k=1}^{K} x_k f_{K-k} \\ &= x_1(P_1(x_1, \ldots, x_{K-2}) + x_{K-1}) + \sum_{k=2}^{K-2} x_k f_{K-k} + x_{K-1} f_1 + x_K \\ &= P_2(x_1, \ldots, x_{K-2}) + 2x_1 x_{K-1} + x_K \end{aligned}$$

for some polynomial P_2. Substituting $x_K = 1 - \sum_{k=1}^{K-1} x_k$ results in

$$f_K = P_3(x_1, \ldots, x_{K-2}) + x_{K-1}(2x_1 - 1)$$

for some polynomial P_3. This is linear in x_{K-1} where $0 \le x_{K-1} \le 1 - \sum_{i=1}^{K-2} x_i$, so

$$f_K \le \max\left\{ f_K|_{x_{K-1}=0}, f_K|_{x_K=0} \right\}$$

where

$$\begin{aligned} f_K|_{x_{K-1}=0} &= P_3(x_1, \ldots, x_{K-2}) \\ f_K|_{x_K=0} &= P_3(x_1, \ldots, x_{K-2}) + \left(1 - \sum_{i=1}^{K-2} x_i\right)(2x_1 - 1). \end{aligned}$$

$f_K|_{x_K=0} \le 1 - x_1 + x_1^2$ by the inductive hypothesis. Now let's upper-bound $f_K|_{x_{K-1}=0}$.

Note that, regardless of the value of x_{K-1}, we have $f_0 = 1$ and $f_k \le 1 - x_1 + x_1^2$ for $1 \le k \le K-2$ by the inductive hypothesis, since these values of f_k are independent of x_{K-1}. Thus,

$$f_{K-1}|_{x_{K-1}=0} = \sum_{k=1}^{K} x_k f_{K-1-k} \le \sum_{k=1}^{K-2} x_k(1 - x_1 + x_1^2) = (1 - x_K)(1 - x_1 + x_1^2)$$

$$\begin{aligned} f_K|_{x_{K-1}=0} &= \sum_{k=1}^{K} x_k f_{K-k} \\ &\le x_1(1 - x_K)\left(1 - x_1 + x_1^2\right) + \sum_{k=2}^{K-2} x_k\left(1 - x_1 + x_1^2\right) + x_K \\ &= \left(x_1(1 - x_K) + 1 - x_1 - x_K\right)\left(1 - x_1 + x_1^2\right) + x_K \\ &= \left(1 - x_K(1 + x_1)\right)\left(1 - x_1 + x_1^2\right) + x_K \\ &= 1 - x_1 + x_1^2 - x_K\left(1 + x_1^3\right) + x_K \\ &\le 1 - x_1 + x_1^2. \end{aligned}$$

Thus, we have shown that $f_K \leq 1 - x_1 + x_1^2$, which completes the inductive step. This concludes the proof of Equation (37).

Now,

$$-\log \sum_{\tilde{s}' \in \tilde{S}_l^*} \tilde{q}_+(\tilde{s}') = -\log f_l \geq 1 - f_l \geq \frac{|A_0|}{|A_+|}\left(1 - \frac{|A_0|}{|A_+|}\right), \tag{38}$$

where the last inequality follows from Equation (37) with $x_1 = |A_0|/|A_+|$.

We now substitute Equations (36) and (38) into Equation (35) to obtain

$$\mathbb{E}_{\tilde{s}\sim\tilde{p}_l}\left[\log \frac{\tilde{q}_0(\tilde{s})}{\tilde{q}_+(\tilde{s})}\right] \geq \frac{|A_0|}{|A_+|}\left(1 - \frac{|A_0|}{|A_+|}\right) - D_{\mathrm{KL}}\left(\tilde{p}_l \parallel \tilde{q}_0\right).$$

Thus, we finally have

$$\begin{aligned}
J_{\text{explore}}(\mathcal{M}_+; p) - J_{\text{explore}}(\mathcal{M}_0; p) &\geq \mathbb{E}_{\tilde{s}\sim\tilde{p}}\left[\log \frac{\tilde{q}_0(\tilde{s})}{\tilde{q}_+(\tilde{s})}\right] \\
&= \sum_{l=1}^{\infty} \tilde{\lambda}(l)\mathbb{E}_{\tilde{s}\sim\tilde{p}_l}\left[\log \frac{\tilde{q}_0(\tilde{s})}{\tilde{q}_+(\tilde{s})}\right] \\
&\geq \sum_{l=1}^{\infty} \tilde{\lambda}(l)\left(\frac{|A_0|}{|A_+|}\left(1 - \frac{|A_0|}{|A_+|}\right) - D_{\mathrm{KL}}\left(\tilde{p}_l \parallel \tilde{q}_0\right)\right) \\
&\geq \frac{|A_0|}{|A_+|}\left(1 - \frac{|A_0|}{|A_+|}\right) - D_{\mathrm{KL}}\left(\tilde{p} \parallel \tilde{\lambda}\tilde{q}_0\right).
\end{aligned}$$

□

G. Relating p-Incompressibility to Skill Learning

The intuition that skills should optimally compress successful trajectories has been previously used by skill-discovery algorithms like LOVE and LEMMA. Using the incompressibility measures introduced in this paper, we can approach skill learning more rigorously. There are two approaches to converting p-incompressibility into a skill-learning objective.

The first approach is to find A_+ that minimizes the lower bound on the p-learning difficulty increase ratio as given in Theorem 4.2. This is equivalent to minimizing

$$\mathcal{L}_1(A_+) = \frac{|A_+|}{\log|A_+|} \sup_{0<\varepsilon<1} \frac{\mathrm{H}[P_+] - \log\left(\frac{1-\varepsilon}{\varepsilon}\right)}{\log\left(\frac{|A_0|}{1-\varepsilon}\right)}, \tag{39}$$

where the sup factor is proportional to the A_+-merged p-incompressibility. Usually, $\mathrm{H}[P_+]$ is large, as a result of which the maximizing ε satisfies $\varepsilon \ll 1$ and $\mathrm{H}[P_+] \gg \log\left(\frac{1-\varepsilon}{\varepsilon}\right)$. Thus, minimizing $\mathcal{L}_1(A_+)$ becomes equivalent to minimizing

$$\mathcal{L}_2(A_+) = \frac{|A_+|}{\log|A_+|}\mathrm{H}[P_+]. \tag{40}$$

When $|A_+|$ is known or given as a hyperparameter, then the objective is to minimize

$$\mathcal{L}_3(A_+) = \mathrm{H}[P_+]. \tag{41}$$

Note that in practice, it is not possible to compute P_+, the distribution of shortest solutions using actions from A_+ to states generated by the environment. However, we do have a training set of offline experience, so we can use our skills to rewrite these solutions and define $\hat{P}_+$ to be the resultant empirical distribution of abstracted solutions. The P_+ appearing in the objectives $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ should thus be interpreted as $\hat{P}_+$ as calculated from our training set.

However, the resultant approximation of $\mathrm{H}[P_+]$ will be a significant under-approximation if the training set is much smaller than the number of states that cover most of the state space under p. In this case, we recommend modeling P_+ with the assumption that it is generated by sampling i.i.d. actions from a distribution $p_{a,+}$ over A_+, with solution length sampled

from a distribution $p_{l,+}$. Then the maximum-likelihood (ML) estimates of $p_{a,+}$ and $p_{l,+}$ are just the empirical distribution of actions and the empirical distribution of solution lengths in the abstracted training set. If we define $\tilde{P}_+$ to be the distribution of action sequences defined by this choice of $p_{a,+}$ and $p_{l,+}$, then we can approximate $\mathrm{H}[P_+] \approx \mathrm{H}[\hat{P}_+, \tilde{P}_+] = \mathrm{H}[p_{l,+}] + \bar{l}_+ \mathrm{H}[p_{a,+}]$, where $\bar{l}_+ := \mathbb{E}_{l \sim p_{l,+}}[l]$ is the average solution length. Under this approximation, $\mathcal{L}_3$ becomes

$$\mathcal{L}_4(A_+) = \mathrm{H}[p_{l,+}] + \bar{l}_+ \mathrm{H}[p_{a,+}]. \tag{42}$$

(We can similarly apply this approximation to $\mathcal{L}_1$ and $\mathcal{L}_2$.) It is often the case that $\mathrm{H}[p_{l,+}]$ is much smaller than $\bar{l}_+ \mathrm{H}[p_{a,+}]$, so neglecting that term results in the objective

$$\mathcal{L}_5(A_+) = \bar{l}_+ \mathrm{H}[p_{a,+}]. \tag{43}$$

Note that $\mathcal{L}_5$ is exactly the minimum description length (MDL) objective used by LOVE (Jiang et al., 2022). It represents the average number of bits required to encode an abstracted solution, where the encoding of actions is optimized for the empirical distribution of actions in the abstracted training set.

The second approach to deriving a skill learning objective from p-incompressibility is based on the idea that the maximally abstracted environment is the least compressible. Using unmerged p-incompressibility to measure incompressibility, this corresponds to the *maximization* objective

$$\mathcal{J}_6(A_+) = \mathrm{IC}(\mathcal{M}_+; p) = \sup_{0<\varepsilon<1} \frac{\mathrm{H}[p] - \log\left(\frac{1-\varepsilon}{\varepsilon}\right)}{\mathbb{E}_{s\sim p}[d_+(s)] \log\left(\frac{|A_+|}{1-\varepsilon}\right)}. \tag{44}$$

Similar to $\mathrm{H}[P_+]$ in $\mathcal{L}_1$, $\mathrm{H}[p]$ in $\mathcal{J}_6$ is often large, in which case the maximizing ε satisfies $\varepsilon \ll 1$ and $\mathrm{H}[p] \gg \log\left(\frac{1-\varepsilon}{\varepsilon}\right)$. Under this approximation, the maximization objective becomes the minimization objective

$$\mathcal{L}_7(A_+) = \mathbb{E}_{s\sim p}[d_+(s)] \log |A_+|. \tag{45}$$

As with P_+, $\mathbb{E}_{s\sim p}[d_+(s)]$ cannot be computed exactly, so we approximate it with the average solution length in the abstracted training set, i.e., $\bar{l}_+$. As a result,

$$\mathcal{L}_7(A_+) = \bar{l}_+ \log |A_+|, \tag{46}$$

which is just $\mathcal{L}_5$ but with a uniform distribution for $p_{a,+}$. It can thus also be interpreted as an MDL objective where the encoding of actions is a uniform code. Note that this is exactly the objective used by LEMMA (Li et al., 2022).