Understanding Generalization in RL via the Epistemic POMDP

Anonymous Author(s)
Affiliation
Address
email

Abstract

Generalization is a central challenge for the deployment of reinforcement learning (RL) systems in the real world. In this paper, we show that the sequential structure of the RL problem necessitates new approaches to generalization beyond the well-studied techniques used in supervised learning. While supervised learning methods can generalize effectively without explicitly accounting for epistemic uncertainty, we describe why appropriate uncertainty handling can actually be essential in RL. We show that generalization to unseen test conditions from a limited number of training conditions induces a kind of implicit partial observability, effectively turning even fully-observed MDPs into POMDPs. Informed by this observation, we recast the problem of generalization in RL as solving the induced partially observed Markov decision process, which we call the epistemic POMDP. We demonstrate the failure modes of algorithms that do not appropriately handle this partial observability, and suggest a simple ensemble-based technique for approximately solving the partially observed problem. Empirically, we demonstrate that our simple algorithm derived from the epistemic POMDP achieves significant gains in generalization over current methods on the Procgen benchmark suite.

1 Introduction

Generalization is a central challenge in machine learning. However, much of the research on reinforcement learning (RL) has been concerned with the problem of optimization: how to master a specific task through online or logged interaction. Generalization to new test-time contexts has received comparatively less attention, although several works have observed empirically [1, 2, 3, 4] that generalization to new situations poses a significant challenge to RL policies learned from a fixed training set of situations. In standard supervised learning, it is known that in the absence of distribution shift and with appropriate inductive biases, optimizing for performance on the training set (i.e., empirical risk minimization) translates into good generalization performance. It is tempting to suppose that the generalization challenges in RL can be solved in the same manner as empirical risk minimization in supervised learning: when provided a training set of contexts, learn the optimal policy within these contexts and then use that policy in new contexts at test-time.

Perhaps surprisingly, we show that such "empirical risk minimization" approaches can be sub-optimal for generalizing to new contexts in RL, even when these new contexts are drawn from the same distribution as the training contexts. As an anecdotal example of why this sub-optimality arises, imagine a robotic zookeeper for feeding otters that must be trained on some set of zoos. When placed in a new zoo, the robot must find and enter the otter enclosure, and has two paths to do so: either peek through all the habitat windows looking for otters, which succeeds with 95% probability in all zoos, or to follow an image of a hand-drawn map of the zoo that unambiguously identifies the otter enclosure, which will succeed as long as the agent is able to successfully parse the image. In every

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

training zoo, the otters can be found more reliably using the image of the map, and so an agent trained to seek the optimal policy in the training zoos would learn a classifier to predict the identity of the otter enclosure from the map, and enter the predicted enclosure. This classification strategy is optimal on the training environments because the agent can learn to perfectly classify the training zoo maps, but it is *sub-optimal* for generalization, because the learned classifier will not be able to perfectly classify every new zoo map at test-time.[1] If the learned map classifier succeeds on anything less than 95% of new zoos at test-time, the strategy of peeking through the windows, although sub-optimal in all the training environments, turns out to be a more reliable strategy for finding the otter habitat in a *new* zoo, and results in higher expected returns at test-time.

Although with enough training zoos, the zookeeper can learn a policy by solving the supervised classification problem (of maps), to generalize optimally when given a limited number of zoos requires a very different strategy. How can we more generally describe the set of behaviors needed for a policy to generalize from a finite number of training contexts in the RL setting? We make the observation that, even in fully-observable domains, the agent's epistemic uncertainty renders the environment *implicitly* partially observed at test-time. In the zookeeper example, although the hand-drawn map provides the exact location of the otter enclosure (and so the enclosure's location is technically fully observed), the agent cannot identify the true parameters of the map classifier from the small set of maps seen at training time, and so the location of the otters is implicitly obfuscated from the agent. We formalize this observation, and show that generalizing optimally at test-time corresponds to solving a partially-observed Markov decision process that we call an **epistemic POMDP**, which is induced by the agent's epistemic uncertainty about the test environment.

That uncertainty about MDP parameters can be modelled as a POMDP is well-studied in Bayesian RL when training and testing on a single task in an online setting, primarily in the context of exploration [5, 6, 7, 8]. However, as we will discuss, this POMDP interpretation has significant and previously undescribed consequences for the generalization problem in RL, where an agent cannot collect more data online, and must instead learn a policy from a fixed set of training contexts that generalizes to new contexts at test-time. We show that the standard approaches that seek to learn policies under such uncertainty do not appropriately account for the induced partial observability, and can be arbitrarily sub-optimal for test-time generalization in theory and in practice. The epistemic POMDP underscores the difficulty of the generalization problem in RL, as compared to classification, and provides an avenue for understanding how we should approach generalization under the sequential nature and non-uniform reward structure of the RL problem. Maximizing expected return in the epistemic POMDP emerges as a principled approach to learning policies that generalize well, and we propose LEEP, an ensemble-based algorithm derived from the POMDP that properly handles the partial observability induced by the epistemic uncertainty about test-time contexts.

The primary contribution of this paper is to use Bayesian RL techniques to reframe generalization in RL as the problem of solving a partially observed Markov decision process, which we call the *epistemic POMDP*. The epistemic POMDP highlights the additional challenges needed for optimal generalization in RL, as compared to supervised learning. We show both theoretically and empirically the necessity of reasoning about this partial observability in order to maximize test-time performance, and suggest simple methods based on the POMDP formulation for doing so. Our work highlights the importance of not only finding ways to help neural networks in RL generalize better, but also on learning policies that degrade gracefully when the underlying neural network eventually does fail to generalize. Empirically, we demonstrate that our algorithm derived from the epistemic POMDP achieves significant gains in performance over current methods on several ProcGen benchmark tasks.

2 Related Work

Many empirical studies have demonstrated the tendency of RL algorithms to overfit significantly to their training environments [1, 2, 3, 4], and the more general increased difficulty of learning policies that generalize in RL as compared to seemingly similar supervised learning problems [9, 10, 11, 12]. These empirical observations have led to a newfound interest in algorithms for generalization in RL, and the development of benchmark RL environments that focus on generalization to new contexts from a limited set of training contexts sharing a similar structure (state and action spaces) but possibly different dynamics and rewards [13, 14, 15, 16, 17].

[1] Much as a supervised learning algorithm cannot attain exactly zero test error, even if it generalizes well.

Generalization in RL. Approaches for improving generalization in RL have fallen into two main categories: improving the ability of function approximators to generalize better with inductive biases, and incentivizing behaviors that are easier to generalize to unseen contexts. To improve the representations learned in RL, prior work has considered imitating environment dynamics [18, 19], seeking bisimulation relations [20, 21], and more generally, addressing representational challenges in the RL optimization process [22, 23]. In image-based domains, inductive biases imposed via neural network design have also been proposed to improve robustness to certain factors of variation in the state [24, 25, 26]. The challenges with generalization in RL that we will describe in this paper stem from the deficiencies of MDP objectives, and cannot be fully solved by choice of representations or functional inductive biases. In the latter category, one approach is domain randomization, varying environment parameters such as coefficients of friction or textures, to obtain behaviors that are effective across many candidate parameter settings [27, 28, 29, 30, 31]. Domain randomization sits within a class of methods that seek robust policies by injecting noise into the agent-environment loop, whether in the state [32], the action (e.g., via max-entropy RL) [14], or intermediary layers of a neural network policy (e.g., through information bottlenecks) [22, 33]. In doing so, these methods effectively introduce partial observability into the problem; while not necessarily equivalent to the partial observability that is induced by the epistemic POMDP, it may indicate why these methods generalize well empirically.

Bayesian RL: Our work recasts generalization in RL within the Bayesian RL framework, the problem of acting optimally under a belief distribution over MDPs (see Ghavamzadeh et al. [8] for a survey). Bayesian uncertainty has been studied in many sub-fields of RL [34, 35, 36, 37], the most prominent being for exploration and learning efficiently in the online RL setting. Bayes-optimal behavior in RL is often reduced to acting optimally in a POMDP, or equivalently, a belief-state MDP [6], of which our epistemic POMDP is a specific instantiation. Learning the Bayes-optimal policy exactly is intractable in all but the simplest problems [38, 39], and many works in Bayesian RL have studied relaxations that remain asymptotically optimal for learning, for example with value of perfect information [5, 40] or Thompson sampling [7, 41, 42]. Our main contribution is to revisit these classic ideas in the context of generalization for RL. We find that the POMDP interpretation of Bayesian RL [5, 6, 43] provides new insights on inadequacies of current algorithms used in practice, and explains why generalization in RL can be more challenging than in supervised learning. Being Bayesian in the generalization setting also requires new tools and algorithms beyond those classically studied in Bayesian RL, since test-time generalization is measured using regret over a *single* evaluation episode, instead of throughout an online training process. As a result, algorithms and policies that minimize short-term regret (i.e., are more exploitative) are preferred over traditional algorithms like Thompson sampling that explore thoroughly to ensure asymptotic optimality at the cost of short-term regret.

3 Problem Setup

We consider the problem of learning RL policies given a set of training contexts that generalize well to new unseen contexts. This problem can be formalized in a Markov decision process (MDP) where the agent does not have full access to the MDP at training time, but only particular initial states or conditions. Before we describe what this means, we must describe the MDP $\mathcal{M}$, which is given by a tuple $(\mathcal{S}, \mathcal{A}, r, T, \rho, \gamma)$, with state space $\mathcal{S}$, action space $\mathcal{A}$, Markovian transition function $T(s_{t+1}|s_t, a_t)$, bounded reward function $r(s_t, a_t)$, and initial state distribution $\rho(s_0)$. A policy π induces a discounted state distribution $d^\pi(s) = (1-\gamma)\mathbb{E}_\pi[\sum_{t\geq 0} \gamma^t 1(s_t = s)]$, and achieves return $J_\mathcal{M}(\pi) = \mathbb{E}_\pi[\sum_{t\geq 0} \gamma^t r(s_t, a_t)]$ in the MDP. Classical results establish that a deterministic Markovian policy π^* maximizes this objective amongst all history-dependent policies.

We focus on generalization in contextual MDPs where the agent is only trained on a subsample of contexts, and seeks to generalize well across unseen contexts. A contextual MDP is an MDP in which the state can be decomposed as $s_t = (c, s'_t)$, a context vector $c \in \mathcal{C}$ that remains constant throughout an episode, and a sub-state $s' \in \mathcal{S}'$ that may vary: $\mathcal{S} := \mathcal{C} \times \mathcal{S}'$. Each context vector corresponds to a different situation that the agent might be in, each with slightly different dynamics and rewards, but some shared structure across which an agent can generalize. During training, the agent is allowed to interact only within a sampled subset of contexts $\mathcal{C}_{\text{train}} \subset \mathcal{C}$. The generalization performance of the agent is measured by the return of the agent's policy in the full contextual MDP $J(\pi)$, corresponding to expected performance when placed in potentially new and unseen contexts. While our examples and experiments will be in contextual MDPs, our theoretical results also apply to other RL generalization settings where the full MDP cannot be inferred unambiguously from the data available during training, for example in offline reinforcement learning [44].

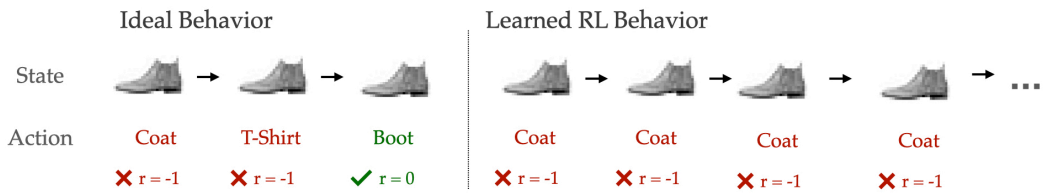

Figure 1: An RL version of classification where an agent must keep guessing the label for an image until it gets it correct. To avoid low test return, policies should change actions if the label guessed was incorrect, but standard RL methods fail to do so, instead guessing the same incorrect label repeatedly.

4 Warmup: Classification as an RL Problem

We begin our study of generalization in RL with an example problem that is intentionally set up to be close to a supervised learning task where generalization is relatively well understood: image classification on the FashionMNIST dataset [45]. In this environment (visualized in Figure 1), an image from the dataset is sampled (the context) at the beginning of an episode and held fixed; the agent must identify the label of the image to complete the episode. If the agent guesses correctly, it receives a reward of 0 and the episode ends; if incorrect, it receives a reward of -1 and the episode continues, so it must attempt another guess for the *same* image at the next time step.

The optimal policy in both the classification and RL version of the problem deterministically outputs the correct label for the image, because the image fully determines the label (in other words, it is a fully observed MDP). However, this optimal strategy generally cannot be learned from a finite training set, since some generalization error is unavoidable. With a fixed training set, the strategy for generalizing in classification remains the same: deterministically choose the label the agent is most confident about. However, the RL setting introduces two new factors: the agent gets multiple tries at classifying the same image, and it knows if an attempted label is incorrect. To generalize best to new test images, an RL policy must leverage this additional structure, for example by trying many possible labels, or by changing actions if the previous guess was incorrect.

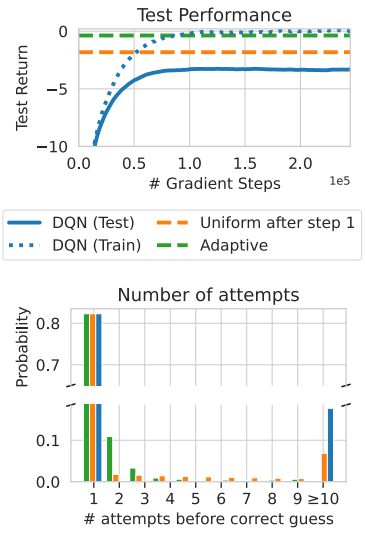

Figure 2: Performance of DQN on FashionMNIST.

A standard RL algorithm, which estimates the optimal policy on the empirical MDP defined by the dataset of training images does not learn to leverage these factors, and instead learns behavior highly sub-optimal for generalization. We obtained a policy by running DQN [46] (experimental details in Appendix A.1), whose policy deterministically chooses the same label for the image at every timestep. Determinism is not specific to DQN, and is inevitable in any RL method that models the problem as an MDP because the optimal policy in the MDP is always deterministic and Markovian. The learned deterministic policy either guesses the correct label immediately, or guesses incorrectly and proceeds to make the same incorrect guess on every subsequent time-step. We compare performance in Figure 2 with a version of the agent that starts to guess randomly if incorrect on the first timestep, and a different agent that acts by process of elimination: first choosing the action it is most confident about, if incorrect, then the second, and so forth. In Section 5.2, we will see that this process-of-elimination is, in some sense, the optimal way to generalize for this task. This experiment reveals a tension: learning policies for generalization that rely on an MDP model fail, even though the underlying environment *is* an MDP. This failure holds in any MDP model with limited data, whether the empirical MDP or more sophisticated MDPs that use uncertainty estimates in their construction.

5 Modeling Generalization in RL as an Epistemic POMDP

To better understand test-time generalization in RL, we study the problem under a Bayesian perspective. We show that training on limited training contexts leads to an implicit partial observability at test-time that we describe using a formalism called the epistemic POMDP.

5.1 The Epistemic POMDP

In the Bayesian framework, when learning given a limited amount of evidence $\mathcal{D}$ from an MDP $\mathcal{M}$, we can use a prior distribution $\mathcal{P}(\mathcal{M})$ to construct a posterior belief distribution $\mathcal{P}(\mathcal{M}|\mathcal{D})$ over the identity of the MDP. For learning in a contextual MDP, $\mathcal{D}$ corresponds to the training contexts $\mathcal{C}_{\text{train}}$ that the agent can interact with. Since the agent cannot fully identify the MDP from the evidence, when the agent is evaluated at test-time, it is uncertain as to which MDP from the posterior distribution $\mathcal{P}(\mathcal{M}|\mathcal{D})$ is being acted in. Following a reduction common in Bayesian RL [6, 8], we model this test-time uncertainty using a partially observed MDP that we will call the **epistemic POMDP**.

The epistemic POMDP is structured as follows: each new episode in the POMDP begins by sampling a single MDP $\mathcal{M} \sim \mathcal{P}(\mathcal{M}|\mathcal{D})$ from the posterior, and then the agent interacts with $\mathcal{M}$ until the episode ends in this MDP. The agent does not observe *which* MDP was sampled, and since the MDP remains fixed for the duration of the episode, this induces implicit partial observability. Effectively, each episode in the epistemic POMDP corresponds to one of the possible situations that the agent believes it might find itself in at test-time.

The epistemic POMDP is formally defined as the tuple $\mathcal{M}^{\text{po}} = (\mathcal{S}^{\text{po}}, \mathcal{O}^{\text{po}}, \mathcal{A}, T^{\text{po}}, r^{\text{po}}, \rho^{\text{po}}, \gamma)$. A state in this POMDP $s_t^{\text{po}} = (\mathcal{M}, s_t)$ contains the identity of the current MDP being acted in $\mathcal{M}$, and the current state in this MDP s_t; we write the state space as $\mathcal{S}^{\text{po}} = \mathbf{M} \times \mathcal{S}$, where $\mathbf{M}$ is the space of MDPs with support under the prior. The agent only gets to observe $o_t^{\text{po}} = s_t$, the state in the MDP ($\mathcal{O}^{\text{po}} = \mathcal{S}$). The initial state distribution is defined by the posterior distribution over MDPs: $\rho^{\text{po}}((\mathcal{M}, s_0)) = \mathcal{P}(\mathcal{M}|\mathcal{D})\rho_{\mathcal{M}}(s_0)$, and the transition and reward functions in the POMDP reflect the dynamics in the current MDP:

$$T^{\text{po}}((\mathcal{M}', s') \mid (\mathcal{M}, s), a) = \delta(\mathcal{M}' = \mathcal{M}) T_{\mathcal{M}}(s'|s, a) \qquad r^{\text{po}}((\mathcal{M}, s), a) = r_{\mathcal{M}}(s, a)\,. \tag{1}$$

What makes the epistemic POMDP a useful tool is that the expected return objective in the POMDP corresponds exactly to the expected return of the agent at test-time when the Bayesian prior is accurate (Eq 2). Therefore, the optimal policy in the epistemic POMDP, $\pi^{*\text{po}}$ is the Bayes-optimal policy for maximizing the expected test-time return.

$$J_{\mathcal{M}^{\text{po}}}(\pi) = J(\pi \mid \mathcal{D}) \coloneqq \mathbb{E}_{\mathcal{M} \sim \mathcal{P}(\mathcal{M})}[J_{\mathcal{M}}(\pi) \mid \mathcal{D}]. \tag{2}$$

The epistemic POMDP is based on well-understood concepts in Bayesian reinforcement learning, and Bayesian modeling more generally. However, in contrast to prior works on Bayesian RL, we are specifically concerned with settings where there is a training-test split, and performance is measured by a single test episode. While using Bayesian RL to accelerate exploration or minimize regret has been well-explored [8], we seek rather to utilize the Bayesian lens specifically to understand generalization – a perspective that is distinct from prior work on Bayesian RL. Towards this goal, the equivalence between test-time return and performance in the epistemic POMDP allows us to use the epistemic POMDP as a proxy for understanding how well current RL methods generalize.

5.2 Understanding Optimality in the Epistemic POMDP

We now study the structure of the epistemic POMDP, and use it to characterize properties of Bayes-optimal test-time behavior and the sub-optimality of alternative policy learning approaches. We begin by revisiting the image classification problem, and describing the induced epistemic POMDP.

Example (Image Classification): In the image classification task, the agent's uncertainty concerns how FashionMNIST images are mapped to labels. Each MDP $\mathcal{M}$ in the posterior distribution corresponds to a different potential classifier $Y_{\mathcal{M}}$, where $Y_{\mathcal{M}}(x)$ denotes the label of x according to $\mathcal{M}$. Acting optimally in the epistemic POMDP requires strategies that work well in expectation over the distribution of labels that is induced by the posterior distribution $p(y|x, \mathcal{D}) = \mathbb{E}_{\mathcal{M} \sim \mathcal{P}(\mathcal{M}|\mathcal{D})}[1(Y_{\mathcal{M}}(x) = y)]$. The Bayes-optimal strategy corresponds to a process of elimination: first choose the most likely label $a = \arg\max p(y|x, \mathcal{D})$; if this is incorrect, eliminate it and choose the next-most likely, repeating until the correct label is finally chosen. Amongst memoryless policies, the optimal behavior is stochastic, sampling labels according to the action distribution $\pi^*(a|x) \propto \sqrt{p(y|x, \mathcal{D})}$ (derivation in Appendix A.2).

The characteristics of the optimal policy in the epistemic POMDP for the image classification RL problem match well-established results that optimal POMDP policies are generally memory-based [47], and amongst memoryless policies, the optimal policy may be stochastic [48, 49]. Because of the equivalence between the epistemic POMDP and test-time behavior, these maxims are also true for Bayes-optimal behavior when maximizing test-time performance.

Remark 5.1. *The Bayes-optimal policy for maximizing test-time performance is in general non-Markovian. When restricted to Markovian policies, the Bayes-optimal policy is in general stochastic.*

Proofs of all statements in the paper are provided in Appendix B. The fact that acting optimally at test-time can require adaptivity (or stochasticity for memoryless policies) provides a new perspective on and formal backing to the various empirical studies that have found improved generalization performance using recurrent networks [50, 51] and stochastic regularization penalties [32, 14, 22, 33].

It is also useful to understand when partial observability *does not* play a significant role. When this is true, the epistemic POMDP objective can coincide with a surrogate MDP approximation, and Bayes-optimal solutions can be attained with fully-observed RL algorithms. In the epistemic POMDP, the identity of the MDP is partially observed, but when the optimal behavior in every MDP from the posterior is identical, then acting optimally just involves following this optimal MDP policy, instead of reasoning about the hidden identity. Perhaps surprisingly, being optimal everywhere is also *necessary* for such a strategy to be Bayes-optimal in general. The following proposition states that, even if a policy is optimal in many (but not all) of the possible MDPs from our posterior, this "optimal" policy can generalize poorly at test-time.

Proposition 5.1. *Let $\epsilon > 0$. There exists posterior distributions $\mathcal{P}(\mathcal{M}|\mathcal{D})$ where a deterministic Markov policy π is optimal with probability at least $1-\epsilon$,*

$$P_{\mathcal{M}\sim\mathcal{P}(\mathcal{M}|\mathcal{D})}\left(\pi \in \arg\max_{\pi'} J_{\mathcal{M}}(\pi')\right) \geq 1-\epsilon, \tag{3}$$

but is outperformed by a uniformly random policy in the epistemic POMDP: $J_{\mathcal{M}^{po}}(\pi) < J_{\mathcal{M}^{po}}(\pi_{unif})$.

This proposition indicates the brittleness of learning policies in an MDP model, since the learned policy may perform poorly at test-time, even if it captures behavior on a majority of environments in the posterior (for example, in the RL image classification task). In general, under partial observability, optimal policies for the MDPs in the posterior may serve as poor guidelines for Bayes-optimal behavior, and in Appendix B.2, we show that the Bayes-optimal policy may take actions that are sub-optimal in *every* environment in the posterior.

As Bayes-optimal memoryless policies are stochastic, one may wonder if simple strategies for inducing stochasticity, such as adding ϵ-greedy noise or entropy regularization, can alleviate the sub-optimality we discuss for deterministic policies. In some cases, this may be true; one particularly interesting result is that in certain goal-reaching problems, entropy-regularized RL can be interpreted as optimizing an epistemic POMDP objective with a specific form of posterior distribution over reward functions (Appendix B.3) [52]. For the more general setting, we show in Appendix B.4 that entropy regularization and other general-purpose techniques can similarly catastrophically fail in epistemic POMDPs. Intuitively, the degree of stochasticity used by the Bayes-optimal policy reflects the agent's epistemic uncertainty about the environment; since standard regularizations are agnostic to this uncertainty, the learned behaviors often do not reflect the appropriate level of stochasticity needed. A maze-solving agent acting Bayes-optimally, for example, may choose to act deterministically in mazes like those it has seen at training, and on others where it is less confident, rely on random exploration to exit the maze, inimitable behavior by regularizations agnostic to this uncertainty.

In general, the structure of the epistemic POMDP indicates that while MDP-based algorithms can serve as a useful starting point for acquiring generalizable skills, maximizing test-time performance may require more nuanced strategies that more carefully reason about this partial observability.

6 Tractably Learning Policies Under the Epistemic POMDP

We now turn our attention to deriving practical approximate methods for learning policies in the epistemic POMDP. We develop a lower bound on the return of a policy in the epistemic POMDP, and prove that optimizing this bound recovers the Bayes-optimal policy. We use this to motivate an ensemble-based algorithm for learning policies from a set of training contexts in a contextual MDP.

6.1 A Lower Bound on the Epistemic POMDP Objective

Towards a tractable algorithm, we assume access to n candidate MDPs sampled from the posterior distribution: $\{\mathcal{M}_i\}_{i\in[n]} \sim \mathcal{P}(\mathcal{M} \mid \mathcal{D})$. These samples define an empirical approximation of the posterior distribution, and consequently induce an empirical epistemic POMDP $\hat{\mathcal{M}}^{\text{po}}$. The objective of the empirical epistemic POMDP is given by $J_{\hat{\mathcal{M}}^{\text{po}}}(\pi) = \frac{1}{n}\sum_{i=1}^{n} J_{\mathcal{M}_i}(\pi_i)$.

As the empirical epistemic POMDP corresponds to a collection of n MDPs [2], we will decompose the optimization problem to mimic this structure, learning n policies $\pi_1, \cdots, \pi_n$, each one in one of the MDPs from the posterior, and combining these policies together to recover a single policy π. Reducing the policy learning problem into a set of MDP learning problems can allow us to leverage the many advances in deep RL for scalably solving MDPs. The following theorem, which lower-bounds the expected return of a policy π in the empirical epistemic POMDP using the policies $\{\pi_i\}_{i\in[n]}$, provides a natural objective for learning policies in this decoupled manner.

Theorem 6.1. *Let $\pi, \pi_1, \cdots \pi_n$ be $n+1$ memoryless policies, and let $r_{\max} = \max_{i,s,a} |r_{\mathcal{M}_i}(s,a)|$. The expected return of π in the empirical epistemic POMDP $J_{\hat{\mathcal{M}}^{po}}(\pi)$ is bounded below as:*

$$J_{\hat{\mathcal{M}}^{po}}(\pi) \geq \frac{1}{n}\sum_{i=1}^{n} J_{\mathcal{M}_i}(\pi_i) - \frac{\sqrt{2}r_{\max}}{(1-\gamma)^2 n}\sum_{i=1}^{n} \mathbb{E}_{s\sim d^{\pi_i}_{\mathcal{M}_i}}\left[\sqrt{D_{KL}\left(\pi_i(\cdot|s) \;||\; \pi(\cdot|s)\right)}\right], \tag{4}$$

This theorem indicates that if the policies in the collection $\{\pi_i\}_{i\in[n]}$ all achieve high return in their respective MDPs (first term) and are imitable by a single policy π (second term), then π is guaranteed to achieve high return in the epistemic POMDP. In contrast, if the policies cannot be closely imitated by a single policy, this collection of policies may not be useful for learning in the epistemic POMDP using the lower bound. This could be the case, for example, if each of the policies π_i is trained to maximize return on its MDP $\mathcal{M}_i$ selfishly without any consideration to the other policies or MDPs. To be useful for the lower bound, each policy π_i should balance between maximizing performance on its MDP and minimizing its deviation from the other policies in the set. The following proposition shows that if the policies are trained jointly to ensure this balance, it not only recovers a good policy, it in fact recovers the optimal policy in the empirical epistemic POMDP.

Proposition 6.1. *Let $f : \{\pi_i\}_{i\in[n]} \mapsto \pi$ be a function that maps n policies to a single policy satisfying $f(\pi, \cdots, \pi) = \pi$ for every policy π, and let α be a hyperparameter satisfying $\alpha \geq \frac{\sqrt{2}r_{max}}{(1-\gamma)^2 n}$. Then letting $\pi_1^*, \ldots \pi_n^*$ be the optimal solution to the following optimization problem:*

$$\{\pi_i^*\}_{i\in[n]} = \underset{\pi_1,\cdots,\pi_n}{\arg\max}\, \frac{1}{n}\sum_{i=1}^{n} J_{\mathcal{M}_i}(\pi_i) - \alpha\sum_{i=1}^{n} \mathbb{E}_{s\sim d^{\pi_i}_{\mathcal{M}_i}}\left[\sqrt{D_{KL}\left(\pi_i(\cdot|s) \;||\; f(\{\pi_i\})(\cdot|s)\right)}\right], \tag{5}$$

the policy $\pi^* := f(\{\pi_i^*\}_{i\in[n]})$ is optimal for the empirical epistemic POMDP.

This proposition shows that a principled way to learn the Bayes-optimal policy is to train a set of policies $\{\pi_i\}_{i\in[n]}$ with Equation 5, and use $\pi^* = f(\{\pi_i^*\}_{i\in[n]})$ as the final policy.

6.2 A Practical Algorithm: LEEP

We now derive a practical algorithm for learning policies when provided training contexts $\mathcal{C}_{\text{train}}$ from an unknown contextual MDP. Before turning Proposition 6.1 into an algorithm, we must discuss two problems: how posterior samples $\mathcal{M}_i \sim \mathcal{P}(\mathcal{M}|\mathcal{D})$ can be approximated, and how the function f that combines policies should be chosen.

Approximating the posterior distribution: Although exactly maintaining a posterior distribution over contextual MDPs can be difficult, especially with image-based observations [53, 54], we can approximate samples from the posterior via a bootstrap sampling technique [55]. To sample a candidate MDP $\mathcal{M}_i$, we sample with replacement from the training contexts $\mathcal{C}_{\text{train}}$ to get a new set of contexts $\mathcal{C}^i_{\text{train}}$, and define $\mathcal{M}_i$ to be the empirical MDP on this subset of training contexts. Rolling out trials from the posterior sample $\mathcal{M}_i$ then corresponds to selecting a context at random from $\mathcal{C}^i_{\text{train}}$, and then rolling out that context. Crucially, note that $\mathcal{M}_i$ still corresponds to a *distribution* over contexts, not a single context, since our goal is to sample from the posterior entire contextual MDPs.

Choosing a link function: The link function f in Proposition 6.1 that combines the set of policies together effectively serves as an inductive bias: since policy optimization in practice is approximate, different choices can yield combined policies with different characteristics. Since optimal behavior in the epistemic POMDP must consider all actions, even those that are potentially sub-optimal in all MDPs in the posterior (as discussed in Section 5.1), we use an "optimistic" link function

[2] Note that when the true environment is a contextual MDP, a sample does not correspond to a single training context within a contextual MDP — it represents an approximation to the *entire* contextual MDP.

Algorithm 1 Linked Ensembles for the Epistemic POMDP (LEEP)

1: Receive training contexts $\mathcal{C}_{\text{train}}$, number of ensemble members n
2: Bootstrap sample training contexts to create $\mathcal{C}^1_{\text{train}}, \ldots \mathcal{C}^n_{\text{train}}$, where $\mathcal{C}^i_{\text{train}} \subset \mathcal{C}_{\text{train}}$.
3: Initialize n policies: $\pi_1, \ldots \pi_n$
4: **for** iteration $k = 1, 2, 3, \ldots$ **do**
5: **for** policy $i = 1, \ldots, n$ **do**
6: Collect environment samples in training contexts $\mathcal{C}^i_{\text{train}}$ using policy π_i
7: Take gradient steps wrt π_i on these samples with augmented RL loss:

$$\pi_i \leftarrow \pi_i - \eta \nabla_i (\mathcal{L}^{RL}(\pi_i) + \alpha \mathbb{E}_{s \sim \pi_i, \mathcal{C}^i_{\text{train}}} [D_{KL}(\pi_i(a|s) \| \max_j \pi_j(a|s))])$$

8: Return $\pi = \max_i \pi$: $\pi(a|s) = \frac{\max_i \pi_i(a|s)}{\sum_{a'} \max_i \pi_i(a'|s)}$.

that does not dismiss any action that is considered by at least one of the policies, specifically $f(\{\pi_i\}_{i \in [n]}) = (\max_i \pi_i)(a|s) := \frac{\max \pi_i(a|s)}{\sum_{a'} \max \pi_i(a'|s)}$.

Algorithm: We learn a set of n policies $\{\pi_i\}_{i \in [n]}$, using a policy gradient algorithm to implement the update step. To update the parameters for π_i, we take gradient steps via a surrogate loss constructed via the standard policy gradient, augmented by a disagreement penalty between the policy and the combined policy $f(\{\pi_i\}_{i \in [n]})$ with a penalty parameter $\alpha > 0$, as in Equation 6:

$$\mathcal{L}(\pi_i) = \mathcal{L}^{RL}(\pi_i) + \alpha \mathbb{E}_{s \sim \pi_i, \mathcal{M}_i} [D_{KL}(\pi_i(a|s) \| \max_j \pi_j(a|s))]. \tag{6}$$

Combining these elements together leads to our method, LEEP, which we summarize in Algorithm 1. In our implementation, we use PPO for $\mathcal{L}^{RL}(\pi_i)$ [56]. In summary, LEEP bootstrap samples the training contexts to create overlapping sets of training contexts $\mathcal{C}^1_{\text{train}}, \ldots \mathcal{C}^n_{\text{train}}$. Every iteration, each policy π_i generates rollouts in training contexts chosen uniformly from its corresponding $\mathcal{C}^i_{\text{train}}$, and is then updated according to Equation 6, which both maximizes the expected reward and minimizes the disagreement penalty between each π_i and the combined policy $\pi = \max_j \pi_j$.

While this algorithm is structurally similar to algorithms for multi-task learning that train a separate policy for each context or group of contexts with a disagreement penalty [57, 58], the motivation and the interpretation of these approaches are completely distinct. In multi-task learning, the goal is to solve a given set of tasks, and these methods promote transfer via a loss that encourages the solutions to the tasks to be in agreement. In our setting, while we also receive a set of tasks (contexts), the goal is not to maximize performance on the training tasks, but rather to learn a policy that maximizes performance on unseen test tasks. The method also has a subtle but important distinction: each of our policies π_i acts on a sample from the contextual MDP posterior, *not* a single training context [57] or element from a disjoint partitioning [58]. This distinction is crucial, since our aim is not to make it easier to solve the training contexts, but the opposite: prevent the algorithm from overfitting to the individual contexts. Correspondingly, our experiments confirm that such multi-task learning approaches do not provide the same generalization benefits as our approach.

7 Experiments

We evaluate LEEP in the Procgen benchmark [16], a challenging suite of tasks testing generalization to unseen contexts. Our experiments seek to answer the following questions:

1. Does LEEP derived from the epistemic POMDP lead to improved test-time performance over standard RL methods?
2. Can LEEP prevent overfitting when provided a limited number of training contexts?
3. How do different algorithmic components of LEEP affect test-time performance ?

The Procgen benchmark is a set of procedurally generated games, each with different generalization challenges. In each game, during training, the algorithm can interact with 200 training levels, before it is asked to generalize to the full distribution of levels. The agent receives a $64 \times 64 \times 3$ image observation, and must output one of 15 possible actions. We instantiate our method using an ensemble of $n = 4$ policies, a penalty parameter of $\alpha = 1$, and PPO [56] to train the individual policies (full implementation details in Appendix C, code provided in supplementary).

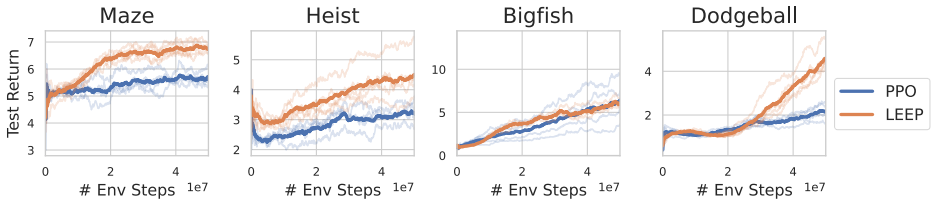

Figure 3: Test set return for LEEP and PPO throughout training in four Procgen environments (averaged across 5 random seeds). LEEP achieves higher test returns than PPO on three tasks (Maze, Heist and Dodgeball) and matches test return on Bigfish while having less variance across seeds.

We evaluate our method on four games in which prior work has found a large gap between training and test performance, and which we therefore conclude pose a significant generalization challenge [16, 23, 26]: Maze, Heist, BigFish, and Dodgeball. In Figure 3, we compare the test-time performance of the policies learned using our method to those learned by a PPO agent with entropy regularization. In three of these environments (Maze, Heist, and Dodgeball), our method outperforms PPO by a significant margin, and in all cases, we find that the generalization gap between training and test performance is lower for our method than PPO (Appendix D). To understand how LEEP behaves with fewer training contexts, we ran on the Maze task with only 50 levels (Figure 4 (top)); the test return of the PPO policy decreases through training, leading to final performance worse than the starting random policy, but our method avoids this degradation.

We perform an ablation study on the Maze and Heist environments (Maze in Figure 4, Heist in Appendix D) to rule out potential confounding causes for the improved generalization that our method displays on the Procgen benchmark tasks. First, to see if the performance benefit derives solely from the use of ensembles, we compare LEEP to a Bayesian model averaging strategy that trains an ensemble of policies without regularization ("Ensemble (no reg)"), and uses a mixture of these policies. This strategy does improve performance over the PPO policy, but does not match LEEP, indicating the usefulness of the regularization. Second, we compared to a version of LEEP that combines the ensemble policies together using the average $\frac{1}{n}\sum_{i=1}^{n}\pi_i(a|s)$ ("LEEP (avg)") achieves worse test-time performance than the optimistic version, which indicates that the inductive bias conferred by the $\max_i \pi_i$ link function is a useful component of the algorithm. Finally, we compare to Distral, a multi-task learning method with different motivations but similar structure to LEEP: this method helps accelerate learning on the provided training contexts (figures in Appendix D), but does not improve generalization performance as LEEP does.

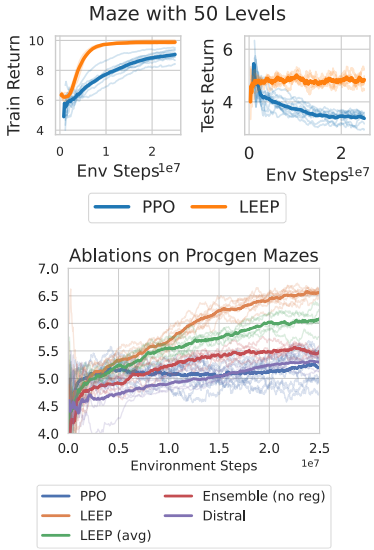

Figure 4: **(top)** Performance on Maze with 50 training levels. **(bottom)** Ablations of LEEP in Maze.

8 Discussion

It has often been observed experimentally that generalization in RL poses a significant challenge, but it has so far remained an open question as to whether the RL setting itself presents additional generalization challenges beyond those seen in supervised learning. In this paper, we answer this question in the affirmative, and show that, in contrast to supervised learning, generalization in RL results in a new type of problem that cannot be solved with standard MDP solution methods, due to partial observability induced by epistemic uncertainty. We call the resulting partially observed setting the epistemic POMDP, where uncertainty about the true underlying MDP results in a challenging partially observed problem. We present a practical approximate method that optimizes a bound for performance in the epistemic POMDP, and show empirically that this approach, which we call LEEP, attains significant improvements in generalization over other RL methods. A limitation of this approach is that it optimizes a crude approximation to the epistemic POMDP with a small number of posterior samples, and may be challenging to scale to better approximations to the true objective. Developing algorithms that better optimize the epistemic POMDP is an exciting avenue for future work, and we hope that this direction will lead to further improvements in generalization in RL.

References

[1] Jesse Farebrother, Marlos C. Machado, and Michael H. Bowling. Generalization and regularization in dqn. *ArXiv*, abs/1810.00123, 2018.

[2] C. Zhang, Oriol Vinyals, R. Munos, and S. Bengio. A study on overfitting in deep reinforcement learning. *ArXiv*, abs/1804.06893, 2018.

[3] Niels Justesen, R. Torrado, Philip Bontrager, Ahmed Khalifa, J. Togelius, and S. Risi. Illuminating generalization in deep reinforcement learning through procedural level generation. *arXiv: Learning*, 2018.

[4] Xingyou Song, Yiding Jiang, Stephen Tu, Yilun Du, and Behnam Neyshabur. Observational overfitting in reinforcement learning. *ArXiv*, abs/1912.02975, 2020.

[5] R. Dearden, N. Friedman, and Stuart J. Russell. Bayesian q-learning. In *AAAI/IAAI*, 1998.

[6] M. Duff and A. Barto. Optimal learning: computational procedures for bayes-adaptive markov decision processes. 2002.

[7] M. Strens. A bayesian framework for reinforcement learning. In *ICML*, 2000.

[8] M. Ghavamzadeh, Shie Mannor, Joelle Pineau, and Aviv Tamar. Bayesian reinforcement learning: A survey. *Found. Trends Mach. Learn.*, 8:359–483, 2015.

[9] A. Zhang, Yuxin Wu, and Joelle Pineau. Natural environment benchmarks for reinforcement learning. *ArXiv*, abs/1811.06032, 2018.

[10] A. Zhang, Nicolas Ballas, and Joelle Pineau. A dissection of overfitting and generalization in continuous reinforcement learning. *ArXiv*, abs/1806.07937, 2018.

[11] S. Whiteson, B. Tanner, Matthew E. Taylor, and P. Stone. Protecting against evaluation overfitting in empirical reinforcement learning. *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 120–127, 2011.

[12] Zhuang Liu, Xuanlin Li, Bingyi Kang, and Trevor Darrell. Regularization matters in policy optimization – an empirical study on continuous control. *arXiv: Learning*, 2020.

[13] Alex Nichol, Vicki Pfau, Christopher Hesse, Oleg Klimov, and John Schulman. Gotta learn fast: A new benchmark for generalization in rl. *ArXiv*, abs/1804.03720, 2018.

[14] K. Cobbe, Oleg Klimov, Christopher Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *ICML*, 2019.

[15] Heinrich Kuttler, Nantas Nardelli, Alexander H. Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. The nethack learning environment. *ArXiv*, abs/2006.13760, 2020.

[16] K. Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. *ArXiv*, abs/1912.01588, 2020.

[17] Austin Stone, Oscar Ramirez, K. Konolige, and Rico Jonschkowski. The distracting control suite - a challenging benchmark for reinforcement learning from pixels. *ArXiv*, abs/2101.02722, 2021.

[18] Max Jaderberg, V. Mnih, Wojciech Czarnecki, Tom Schaul, Joel Z. Leibo, D. Silver, and K. Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *ArXiv*, abs/1611.05397, 2017.

[19] Adam Stooke, Kimin Lee, P. Abbeel, and M. Laskin. Decoupling representation learning from reinforcement learning. *ArXiv*, abs/2009.08319, 2020.

[20] A. Zhang, Rowan McAllister, R. Calandra, Y. Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *ArXiv*, abs/2006.10742, 2020.

[21] Rishabh Agarwal, Marlos C. Machado, P. S. Castro, and Marc G. Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. *ArXiv*, abs/2101.05265, 2021.

[22] Maximilian Igl, K. Ciosek, Yingzhen Li, Sebastian Tschiatschek, C. Zhang, Sam Devlin, and Katja Hofmann. Generalization in reinforcement learning with selective noise injection and information bottleneck. In *NeurIPS*, 2019.

[23] Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. *ArXiv*, abs/2010.03934, 2020.

[24] Kimin Lee, Kibok Lee, Jinwoo Shin, and H. Lee. Network randomization: A simple technique for generalization in deep reinforcement learning. In *ICLR*, 2020.

[25] Ilya Kostrikov, Denis Yarats, and R. Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *ArXiv*, abs/2004.13649, 2020.

[26] Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov, and R. Fergus. Automatic data augmentation for generalization in deep reinforcement learning. *ArXiv*, abs/2006.12862, 2020.

[27] Fereshteh Sadeghi and Sergey Levine. (cad)2rl: Real single-image flight without a single real image. *ArXiv*, abs/1611.04201, 2017.

[28] Joshua Tobin, Rachel Fong, Alex Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017.

[29] A. Rajeswaran, Sarvjeet Ghotra, Sergey Levine, and Balaraman Ravindran. Epopt: Learning robust neural network policies using model ensembles. *ArXiv*, abs/1610.01283, 2017.

[30] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8, May 2018.

[31] Katie Kang, Suneel Belkhale, Gregory Kahn, Pieter Abbeel, and Sergey Levine. Generalization through simulation: Integrating simulated and real data into deep reinforcement learning for vision-based autonomous flight. In *2019 international conference on robotics and automation (ICRA)*, pages 6008–6014. IEEE, 2019.

[32] F. Stulp, E. Theodorou, J. Buchli, and S. Schaal. Learning to grasp under uncertainty. *2011 IEEE International Conference on Robotics and Automation*, pages 5703–5708, 2011.

[33] X. Lu, Kimin Lee, P. Abbeel, and Stas Tiomkin. Dynamics generalization via information bottleneck in deep reinforcement learning. *ArXiv*, abs/2008.00614, 2020.

[34] Deepak Ramachandran and E. Amir. Bayesian inverse reinforcement learning. In *IJCAI*, 2007.

[35] A. Lazaric and M. Ghavamzadeh. Bayesian multi-task reinforcement learning. In *ICML*, 2010.

[36] Wonseok Jeon, Seokin Seo, and Kee-Eung Kim. A bayesian approach to generative adversarial imitation learning. In *NeurIPS*, 2018.

[37] Luisa M. Zintgraf, K. Shiarlis, Maximilian Igl, Sebastian Schulze, Y. Gal, Katja Hofmann, and S. Whiteson. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. *ArXiv*, abs/1910.08348, 2020.

[38] R. Weber. On the gittins index for multiarmed bandits. *Annals of Applied Probability*, 2:1024–1033, 1992.

[39] P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete bayesian reinforcement learning. *Proceedings of the 23rd international conference on Machine learning*, 2006.

[40] R. Dearden, N. Friedman, and D. Andre. Model based bayesian exploration. In *UAI*, 1999.

[41] Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In *NIPS*, 2013.

[42] Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Math. Oper. Res.*, 39:1221–1243, 2014.

[43] Stephane Ross, Brahim Chaib-draa, and Joelle Pineau. Bayes-adaptive pomdps. In *NIPS*, pages 1225–1232, 2007.

[44] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[45] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[46] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[47] George E Monahan. State of the art—a survey of partially observable markov decision processes: theory, models, and algorithms. *Management science*, 28(1):1–16, 1982.

[48] Satinder P. Singh, Tommi S. Jaakkola, and Michael I. Jordan. Learning without state-estimation in partially observable markovian decision processes. In *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*, page 284–292, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[49] Guido Montúfar, K. Zahedi, and N. Ay. Geometry and determinism of optimal stationary control in partially observable markov decision processes. *ArXiv*, abs/1503.07206, 2015.

[50] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[51] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. *CoRR*, abs/1710.06537, 2017.

[52] Benjamin Eysenbach and Sergey Levine. If maxent rl is the answer, what is the question? *arXiv preprint arXiv:1910.01913*, 2019.

[53] Alex X. Lee, Richard Zhang, F. Ebert, P. Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *ArXiv*, abs/1804.01523, 2018.

[54] Danijar Hafner, T. Lillicrap, Ian S. Fischer, R. Villegas, David R Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. *ArXiv*, abs/1811.04551, 2019.

[55] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[56] John Schulman, F. Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

[57] Yee Teh, Victor Bapst, Wojciech M. Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[58] Dibya Ghosh, Avi Singh, Aravind Rajeswaran, Vikash Kumar, and Sergey Levine. Divide-and-conquer reinforcement learning. In *International Conference on Learning Representations*, 2018.

[59] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *In Proc. 19th International Conference on Machine Learning*. Citeseer, 2002.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This paper focuses on fundamental algorithms for improving reinforcement learning, and as such we do not expect any negative societal impacts beyond those related more broadly to reinforcement learning methods.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Proofs for the theorems in our paper can be found in the Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We include the code to run our method on the Procgen benchmark in the supplementary.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We provide details for the FashionMNIST experiment in Appendix A, and details for the Procgen benchmark evaluation in Appendix C.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Results on ProcGen in our paper are with 5 seeds: we plot traces of all seeds, bolding the average performance.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Details can be found in Appendix C.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite the Fashion-MNIST dataset [45], the Procgen environment [16], and the codebase [23] upon which our algorithm is built.
   (b) Did you mention the license of the assets? [N/A] All materials used have open source licenses; [45, 16] under MIT licensing, and [23] under CC BY-NC 4.0.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The code to run our Procgen experiments are provided in the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]