Grounding Aleatoric Uncertainty for Unsupervised Environment Design

Minqi Jiang[*] UCL & Meta AI
Michael Dennis UC Berkeley
Jack Parker-Holder University of Oxford
Andrei Lupu MILA & Meta AI
Heinrich Küttler[‡] Inflection AI
Edward Grefenstette[‡] UCL & Cohere
Tim Rocktäschel[‡] UCL
Jakob Foerster FLAIR, U of Oxford

Abstract

Adaptive curricula in reinforcement learning (RL) have proven effective for producing policies robust to discrepancies between the train and test environment. Recently, the Unsupervised Environment Design (UED) framework generalized RL curricula to generating sequences of entire environments, leading to new methods with robust minimax regret properties. Problematically, in partially-observable or stochastic settings, optimal policies may depend on the ground-truth distribution over aleatoric parameters of the environment in the intended deployment setting, while curriculum learning necessarily shifts the training distribution. We formalize this phenomenon as *curriculum-induced covariate shift* (CICS), and describe how its occurrence in aleatoric parameters can lead to suboptimal policies. Directly sampling these parameters from the ground-truth distribution avoids the issue, but thwarts curriculum learning. We propose SAMPLR, a minimax regret UED method that optimizes the ground-truth utility function, even when the underlying training data is biased due to CICS. We prove, and validate on challenging domains, that our approach preserves optimality under the ground-truth distribution, while promoting robustness across the full range of environment settings.

1 Introduction

Adaptive curricula, which dynamically adjust the distribution of training environments to best facilitate learning, have played a key role in many recent achievements in deep reinforcement learning (RL). Applications have spanned both single-agent RL [32, 50, 55, 22], where adaptation occurs over environment variations, and multi-agent RL, where adaptation can additionally occur over co-players [40, 49, 41]. These methods demonstrably improve the sample efficiency and robustness of the final policy [25, 8, 21, 20], e.g. by presenting the agent with challenges at the threshold of its abilities.

In this paper we introduce and address a fundamental problem relevant to adaptive curriculum learning methods for RL, which we call *curriculum-induced covariate shift* (CICS). Analogous to the covariate shift that occurs in supervised learning (SL) [18], CICS refers to a mismatch between the *input distribution* at training and test time. In the case of RL, we will show this becomes problematic when the shift occurs over the *aleatoric parameters* of the environment—those aspects of the environment holding irreducible uncertainty even in the limit of infinite experiential data [9]. While in some cases, CICS may impact model performance in SL, adaptive curricula for SL have generally not been found to be as impactful as in RL [52]. Therefore, we focus on addressing CICS specifically as it arises in the RL setting, leaving investigation of its potential impact in SL to future work.

[*]Correspondence to msj@meta.com. ‡ Work done while at Meta AI.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

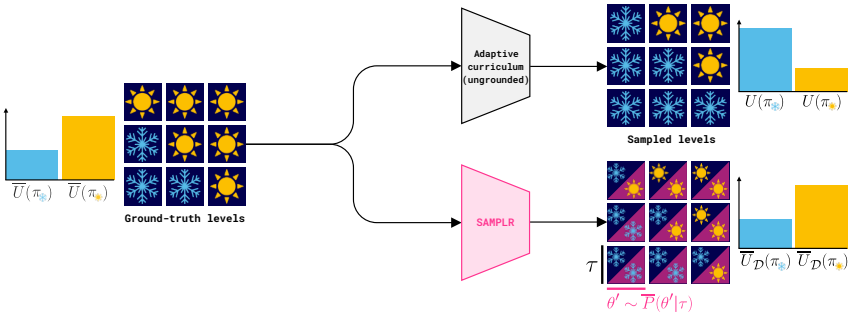

Figure 1: Adaptive curricula can result in covariate shifts in environment parameters with respect to the ground-truth distribution $\overline{P}(\Theta)$ (top path), e.g. whether a road is icy or not, which can cause the policy to be optimized for a utility function U differing from the ground-truth utility function $\overline{U}$ based on $\overline{P}$ (See Equation 1). Here, the policies $\pi_{❄}$ and $\pi_{☀}$ drive assuming ice and no ice respectively. SAMPLR (bottom path) matches the distribution of training transitions to that under $\overline{P}(\Theta|\tau)$ (pink triangles), thereby ensuring the optimal policy trained under a biased curriculum retains optimality for the ground-truth distribution $\overline{P}$.

To establish precise language around adaptive curricula, we cast our discussion under the lens of Unsupervised Environment Design [UED, 8]. UED provides a formal problem description for which an optimal curriculum is the solution, by defining the *Underspecified POMDP* (UPOMDP; see Section 2), which expands the classic POMDP with a set of *free parameters* Θ, representing the aspects of the environment that may vary. UED then seeks to adapt distributions over Θ to maximize some objective, potentially tied to the agent's performance. UED allows us to view adaptive curricula as emerging via a multi-player game between a *teacher* that proposes environments with parameters $\theta \sim P(\Theta)$ and a *student* that learns to solve them. In addition to notational clarity, this formalism enables using game theoretic constructs, such as Nash equilibria [NE, 26], to analyze curricula.

This game-theoretic view has led to the development of curriculum methods with principled robustness guarantees, such as PAIRED [8] and Robust Prioritized Level Replay [PLR$^{\perp}$, 20], which aim to maximize a student's regret and lead to minimax regret [37] policies at NE. Thus, at NE, the student can solve all solvable environments within the training domain. However, in their current form the UED robustness guarantees are misleading: if the UED curriculum deviates from a ground-truth distribution $\overline{P}(\Theta)$ of interest, i.e. the distribution at deployment, with respect to aleatoric parameters $\Theta' \subset \Theta$, the resulting policies may be suboptimal under the ground-truth distribution $\overline{P}$.

For a concrete example of how CICS can be problematic, consider the case of training a self-driving car to navigate potentially icy roads, when icy conditions rarely occur under $\overline{P}$. When present, the ice is typically hard to spot in advance; thus, the aleatoric parameters Θ' correspond to whether each section of the road is icy. A priori, a curriculum should selectively sample more challenging icy settings to facilitate the agent's mastery over such conditions. However, this approach risks producing an overly-pessimistic agent (i.e. one that assumes that ice is common), driving slowly even in fair weather. Such a policy leads to inadequate performance on $\overline{P}$, which features ice only rarely.

We can preserve optimality on $\overline{P}$ by *grounding the policy*—that is, ensuring that the agent acts optimally with respect to the *ground-truth utility function* for any action-observation history τ and the implied ground-truth posterior over Θ:

$$\overline{U}(\pi|\tau) = \mathbb{E}_{\theta \sim \overline{P}(\theta|\tau)} \left[\overline{U}(\pi|\tau, \theta) \right], \tag{1}$$

where the ground-truth utility conditioned on X, $\overline{U}(\pi|X)$, is defined to be $\mathbb{E}_{\tau,\theta \sim \overline{P}(\theta|X)} \left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$, for rewards r_t and a discount γ.

We can ground the policy by *grounding the training distribution*, which means constraining the training distribution of aleatoric parameters $P(\Theta')$ to match $\overline{P}(\Theta')$. This is trivially accomplished by directly sampling $\theta' \sim \overline{P}(\Theta')$, which we call *naive grounding*. Unfortunately, this approach makes many curricula infeasible by removing the ability to selectively sample environment settings over aleatoric parameters. Applying this strategy to the self-driving agent may result in a policy that is optimal in expectation under $\overline{P}$ where there is rarely ice, but nevertheless fails to drive safely on ice.

We wish to maintain the ability to bias a training distribution, since it is required for curriculum learning, while ensuring the resulting decisions remain optimal in expectation under $\overline{P}$. This goal is captured by the following objective:

$$\overline{U}_{\mathcal{D}}(\pi) = \mathbb{E}_{\tau \sim \mathcal{D}}\left[\, \overline{U}(\pi|\tau)\right], \tag{2}$$

where $\mathcal{D}$ is the training distribution of τ. Under naive grounding, $\mathcal{D}$ is equal to $\overline{P}(\tau)$ and Equation 2 reduces to $\overline{U}(\pi)$. To overcome the limitations of naive grounding, we develop an approach that allows $\mathcal{D}$ to deviate from $\overline{P}(\tau)$, e.g. by prioritizing levels most useful for learning, but still grounds the policy by evaluating decisions following potentially biased training trajectories τ according to $\overline{U}(\pi|\tau)$. Figure 1 summarizes this approach, and contrasts it with an ungrounded adaptive curriculum.

In summary this work presents the following contributions: i) We first formalize the problem of CICS in RL in Section 3. ii) Then, we present SAMPLR, which extends PLR$^{\perp}$, a state-of-the-art UED method, to preserve optimality on $\overline{P}$ while training under a usefully biased training distribution in Section 4. iii) We prove in Section 5 that SAMPLR promotes Bayes-optimal policies that are robust over all environment settings $\theta \sim \overline{P}(\Theta)$. iv) Our experiments validate these conclusions in two challenging domains, where SAMPLR learns highly robust policies, while PLR$^{\perp}$ fails due to CICS.

2 Background

2.1 Unsupervised Environment Design

Unsupervised Environment Design [UED, 8] is the problem of automatically generating an adaptive distribution of environments which will lead to policies that successfully transfer within a target domain. The domain of possible environment settings is represented by an Underspecified POMDP (UPOMDP), which models each environment instantiation, or *level*, as a specific setting of the *free parameters* that control how the environment varies across instances. Examples of free parameters are the position of walls in a maze or friction coefficients in a physics-based task. Formally a UPOMDP is defined as a tuple $\mathcal{M} = \langle A, O, \Theta, \mathcal{S}, \mathcal{T}, \mathcal{I}, \mathcal{R}, \gamma \rangle$, where A is the action space, O is the observation space, Θ is the set of free parameters, S is the state space, $\mathcal{T} : S \times A \times \Theta \to \mathbf{\Delta}(S)$ is the transition function, $\mathcal{I} : S \to O$ is the observation function, $\mathcal{R} : S \to \mathbb{R}$ is the reward function, and γ is the discount factor. UED typically approaches the curriculum design problem as training a *teacher* agent that co-evolves an adversarial curriculum for a *student* agent, e.g. by maximizing the student's regret.

2.2 Prioritized Level Replay

We focus on a recent UED algorithm called Robust Prioritized Level Replay [PLR$^{\perp}$, 21], which performs environment design via random search. PLR$^{\perp}$ maintains a buffer of the most useful levels for training, according to an estimate of learning potential—typically based on regret, approximated by a function of the temporal-difference (TD) errors incurred on each level. For each episode, with probability p, PLR$^{\perp}$ actively samples the next training level from this buffer, and otherwise evaluates regret on a new level $\theta \sim \overline{P}(\Theta)$ without training. This sampling mechanism provably leads to a minimax regret policy for the student at NE, and has been shown to improve sample-efficiency and generalization. The resultant regret-maximizing curricula naturally avoid unsolvable levels, which have no regret. We provide implementation details for PLR$^{\perp}$ in Appendix A.

3 Curriculum-Induced Covariate Shift

Since UED algorithms formulate curriculum learning as a multi-agent game between a teacher and a student agent, we can formalize when CICS becomes problematic by considering the equilibrium point of this game: Let Θ be the environment parameters controlled by UED, $\overline{P}(\Theta)$, their ground-truth distribution, and $P(\Theta)$, their curriculum distribution at equilibrium. We use τ_t to refer to the joint action-observation history (AOH) of the student until time t (and simply τ when clear from context). Letting $V(\pi|\tau_t)$ denote the value function under the curriculum distribution $P(\Theta)$, we characterize an instance of CICS over Θ as *problematic* if the optimal policy under $P(\Theta)$ differs from that under the ground-truth $\overline{P}(\Theta)$ for some τ_t, so that

$$\arg\max_{\pi} V(\pi|\tau_t) \neq \arg\max_{\pi} \overline{V}(\pi|\tau_t).$$

The value function $\overline{V}(\pi|\tau_t)$ with respect to $\overline{P}(\Theta)$ can be expressed as a marginalization over θ:

$$\overline{V}(\pi|\tau_t) = \sum_{\theta} \overline{P}(\theta|\tau_t)\tilde{V}(\pi|\tau_t,\theta) \propto \sum_{\theta} \overline{P}(\theta)\tilde{P}(\tau_t|\theta)\tilde{V}(\pi|\tau_t,\theta). \tag{3}$$

Here, the notation $\overline{P}(\theta)$ means $\overline{P}(\Theta = \theta)$, and the tilde on the $\tilde{P}$ and $\tilde{V}$ terms indicates independence from any distribution over Θ, as they both condition on θ. Importantly, the value function under the curriculum distribution $V(\pi|\tau_t)$ corresponds to Equation 3 with $\overline{P}$ replaced by P. We see that $\overline{V}(\pi|\tau_t)$ is unchanged for a given τ_t when $\overline{P}(\theta)$ is replaced with $P(\theta)$ if 1) $\overline{P}(\theta^*|\tau_t) = 1$ for some θ^*, and 2) $\overline{P}$ shares support with P. Then $\tilde{P}(\tau_t|\theta) = 1$ iff $\theta = \theta^*$ and zero elsewhere. In this case, the sums reduce to $\overline{V}(\pi|\tau_t, \theta^*)$, regardless of changing the ground-truth distribution $\overline{P}$ to P. In other words, when Θ is fully determined given the current history τ, covariate shifts over Θ with respect to $\overline{P}(\Theta)$ have no impact on policy evaluation and thus the value function for the optimal policy. If the first condition does not hold, the uncertainty over the value of some subset $\Theta' \subset \Theta$ is irreducible given τ, making Θ' aleatoric parameters for the history τ. Thus, assuming the curriculum shares support with the ground-truth distribution, covariate shifts only alter the optimal policy at τ when they occur over aleatoric parameters given τ. Such parameters can arise when the environment is inherently stochastic or when the cost of reducing uncertainty is high.

Crucially, our analysis assumes P and $\overline{P}$ share support over Θ. When this assumption is broken, the policy trained under the curriculum can be suboptimal for environment settings θ, for which $P(\theta) = 0$ and $\overline{P}(\theta) > 0$. In this paper, we specifically assume that P and $\overline{P}$ share support and focus on addressing suboptimality under the ground-truth $\overline{P}$ due to CICS over the aleatoric parameters Θ'.

This discussion thus makes clear that problematic CICS can be resolved by *grounding the training distribution*, i.e. enforcing the constraint $P(\Theta'|\tau) = \overline{P}(\Theta'|\tau)$ for the aleatoric parameters of the environment. This constraint results in *grounding the policy*, i.e. ensuring it is optimal with respect to the ground-truth utility function based on $\overline{P}$ (Equation 1). As discussed, naive grounding satisfies this constraint by directly sampling $\theta' \sim \overline{P}(\Theta')$, at the cost of curricula over Θ'. This work develops an alternative for satisfying this constraint while admitting curricula over Θ'.

4 Sample-Matched PLR (SAMPLR)

We now describe a general strategy for addressing CICS, and apply it to PLR$^{\perp}$, resulting in Sample-Matched PLR (SAMPLR). This new UED method features the robustness properties of PLR$^{\perp}$ while mitigating the potentially harmful effects of CICS over the aleatoric parameters Θ'.

Algorithm 1: Sample-Matched PLR (SAMPLR)

Randomly initialize policy $\pi(\phi)$, an empty level buffer $\mathbf{\Lambda}$ of size K, and belief model $\mathcal{B}(s_t|\tau)$.
while *not converged* **do**
  Sample replay-decision Bernoulli, $d \sim \overline{P}_D(d)$
  if $d = 0$ or $|\Lambda| = 0$ **then**
    Sample level θ from level generator
    Collect π's trajectory τ on θ, with a stop-gradient $\phi_{\perp}$
  else
    Use PLR to sample a replay level from the level store, $\theta \sim \mathbf{\Lambda}$
    Collect fictitious trajectory τ' on θ, based on $s'_t \sim \mathcal{B}$
    Update π with rewards $\boldsymbol{R}(\tau')$
  end
  Compute PLR score, $S = \textbf{score}(\tau', \pi)$
  Update $\mathbf{\Lambda}$ with θ using score S
end

As discussed in Section 3, CICS become problematic when the covariate shift occurs over some aleatoric subset Θ' of the environment parameters Θ, such that the expectation over Θ' influences the optimal policy. Adaptive curriculum methods like PLR$^{\perp}$ prioritize sampling of environment settings where the agent experiences the most learning. While such a curriculum lets the agent focus on correcting its largest errors, the curriculum typically changes the distribution over aleatoric parameters Θ', inducing bias in the resulting decisions. Ideally, we can eliminate this bias, ensuring the resulting policy makes optimal decisions with respect to the ground-truth utility function, conditioned on the current trajectory:

$$\overline{U}(\pi|\tau) = \mathbb{E}_{\theta' \sim \overline{P}(\theta'|\tau)}\left[\, \overline{U}(\pi|\tau, \theta')\right]. \tag{4}$$

A naive solution for grounding is to simply exclude Θ' from the set of environment parameters under curriculum control. That is, for each environment setting proposed by the curriculum, we

resample $\theta' \sim \overline{P}$. We refer to this approach as *naive grounding*. Naive grounding forces the expected reward and next state under each transition at the current AOH τ to match that under $\overline{P}$. Thus, optimal policies under naive grounding must be optimal with respect to the ground-truth distribution over θ'.

While technically simple, naive grounding suffers from lack of control over Θ'. This limitation is of no concern when the value of Θ' does not alter the distribution of τ until the terminal transition, e.g. when Θ' is the correct choice in a binary choice task, thereby only influencing the final, sparse reward when the right choice is made. In fact, our initial experiment in Section 6 shows naive grounding performs well in such cases. However, when the value of Θ' changes the distribution of τ before the terminal transition, the agent may benefit from a curriculum that actively samples levels which promote learning robust behaviors under unlikely events. Enabling the full benefits of the curriculum in such cases requires the curriculum to selectively sample values of Θ'. Instead of naive grounding, we aim to ground only the policy updates, allowing the curriculum to bias the training distribution. This can be accomplished by optimizing the following objective:

$$\overline{U}_{\mathcal{D}}(\pi) = \mathbb{E}_{\tau \sim \mathcal{D}} \left[\overline{U}(\pi|\tau) \right]. \tag{5}$$

To achieve this, we replace the reward r_t and next state s_{t+1} with counterfactual values that would be experienced if θ' were consistent with τ and $\overline{P}$, so that $\theta' \sim \overline{P}(\theta'|\tau)$. This substitution occurs by simulating a *fictitious transition*, where the fictitious state is sampled as $s'_t \sim \mathcal{B}(s'_t|\tau)$, the action as $a_t \sim \pi(\cdot|\tau)$ (as per usual), the fictitious next state as $s'_{t+1} = \mathcal{T}(s'_t, a_t)$, and the fictitious reward as $r'_t = \mathcal{R}(s'_{t+1})$. The belief model $\mathcal{B}(s'_t|\tau)$ is the ground-truth posterior of the current state given τ:

$$\mathcal{B}(s_t|\tau) = \sum_{\theta'} \overline{P}(s_t|\tau, \theta')\overline{P}(\theta'|\tau). \tag{6}$$

Fictitious transitions, summarized in Figure 2, ground the observed rewards and state transitions to $\overline{P}$. Should training on these transitions lead to an optimal policy over Θ, this policy will also be optimal with respect to $\overline{P}$. We prove this property in Section 5. Fictitious transitions thus provide the benefit of naive grounding without giving up curriculum control over Θ'.

In general, we implement $\mathcal{B}$ as follows: Given $\overline{P}(\Theta')$ as a prior, we model the posterior $\overline{P}(\theta'|\tau)$ with Bayesian inference. The posterior could be learned via supervised learning with trajectories collected from the environment for a representative selection of θ'. Further, we may only have limited access to $\overline{P}(\Theta)$ throughout training, for example, if sampling $\overline{P}(\Theta)$ is costly. In this case, we can learn an estimate $\hat{P}(\Theta')$ from samples we do collect from $\overline{P}(\Theta)$, which can occur online. We can then use $\hat{P}(\Theta')$ to inform the belief model.

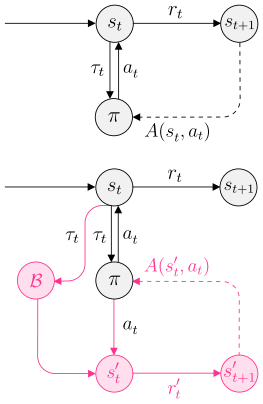

Figure 2: A standard RL transition (top) and a fictitious transition used by SAMPLR (bottom). A is the advantage function.

SAMPLR, summarized in Algorithm 1, incorporates this fictitious transition into PLR$^{\perp}$ by replacing the transitions experienced in replay levels sampled by PLR$^{\perp}$ with their fictitious counterparts, as PLR$^{\perp}$ only trains on these trajectories. PLR$^{\perp}$ uses PPO with the Generalized Advantage Estimator [GAE, 38] as the base RL algorithm, where both advantage estimates and value losses can be written in terms of one-step TD errors δ_t at time t. Training on fictitious transitions then amounts to computing these TD errors with fictitious states and rewards: $\delta_t = r'_t + V(s'_t) - V(s'_{t+1})$. Importantly, because PLR$^{\perp}$ provably leads to policies that minimize worst-case regret over all θ at NE, SAMPLR enjoys the same property for $\theta \sim \overline{P}(\Theta)$. A proof of this fact is provided in Section 5.

Applying SAMPLR requires two key assumptions: First, the simulator can be reset to a specific state, which is often true, as RL largely occurs in resettable simulators or those that can be made to do so. When a resettable simulator is not available, a possible solution is to learn a model of the environment which we leave for future work. Second, we have knowledge of $\overline{P}(\Theta')$. Often, we know $\overline{P}$ *a priori*, e.g. via empirical data or as part of the domain specification, as in games of chance.

5 The Grounded Optimality of SAMPLR

Training on fictitious transitions is a method for learning an optimal policy with respect to the ground-truth utility function $\overline{U}_{\mathcal{D}}(\pi)$ over the distribution $\mathcal{D}$ of training trajectories τ, defined in Equation 5.

When $\mathcal{D}$ corresponds to the distribution of trajectories on levels $\theta \sim \overline{P}(\Theta)$, $\overline{U}_{\mathcal{D}}(\pi)$ reduces to the ground-truth utility function, $\overline{U}(\pi)$. For any UED method, our approach ensures that, in equilibrium, the resulting policy is Bayes-optimal with respect to $\overline{P}(\Theta)$ for all trajectories in the support of $\mathcal{D}$.

Remark 1. *If π^* is optimal with respect to the ground-truth utility function $\overline{U}_{\mathcal{D}}(\pi)$ then it is optimal with respect to the ground-truth distribution $\overline{P}(\Theta)$ of environment parameters on the support of $\mathcal{D}$.*

Proof. By definition we have $\pi^* \in \arg\max_{\pi \in \Pi}\{\overline{U}_{\mathcal{D}}(\pi)\} = \arg\max_{\pi \in \Pi}\{\mathbb{E}_{\tau \sim \mathcal{D}}\left[\overline{U}(\pi|\tau)\right]\}$. Since π can condition on the initial trajectory τ, the action selected after each trajectory can be independently optimized. Therefore, for all $\tau \in \mathcal{D}$, $\pi^* \in \arg\max_{\pi \in \Pi}\{\overline{U}(\pi|\tau)\}$ implying that π^* is the optimal policy maximizing $\overline{U}(\pi|\tau)$. □

Thus, assuming the base RL algorithm finds Bayes-optimal policies, a UED method that optimizes the ground-truth utility function, as done by SAMPLR, results in Bayes-optimal performance over the ground-truth distribution. If the UED method maximizes worst-case regret, we can prove an even stronger property we call *robust ϵ-Bayes optimality*.

Let $\overline{U}_\theta(\pi)$ be the ground-truth utility function for π on the distribution $\mathcal{D}_\theta^\pi$ of initial trajectories sampled from level θ, so that $\overline{U}_\theta(\pi) = \overline{U}_{\mathcal{D}_\theta^\pi}(\pi)$. Given a policy $\overline{\pi}$ maximizing $\overline{U}_\theta(\pi)$, we say that $\overline{\pi}$ is robustly ϵ-Bayes optimal iff for all θ in the domain of $\overline{P}(\Theta)$ and all π', we have

$$\overline{U}_\theta(\overline{\pi}) \geq \overline{U}_\theta(\pi') - \epsilon.$$

Note how this property differs from being simply ϵ-Bayes optimal, which would only imply that

$$\overline{U}(\overline{\pi}) \geq \overline{U}(\pi') - \epsilon.$$

Robust ϵ-Bayes optimality requires $\overline{\pi}$ to be ϵ-optimal on all levels θ in the support of the ground-truth distribution, even those rarely sampled under $\overline{P}(\Theta)$. We will show that at ϵ-Nash equilibrium, SAMPLR results in a robustly ϵ-Bayes optimal policy for the ground-truth utility function $\overline{U}_\theta(\pi)$. In contrast, training directly on levels $\theta \sim \overline{P}(\Theta)$ results in a policy that is only ϵ-Bayes optimal.

Theorem 1. *If π^* is ϵ-Bayes optimal with respect to $\overline{U}_{\widehat{\mathcal{D}}}(\pi)$ for the distribution $\widehat{\mathcal{D}}$ of trajectories sampled under π over levels maximizing the worst-case regret of π, as occurs under SAMPLR, then π^* is robustly ϵ-Bayes optimal with respect to the ground-truth utility function, $\overline{U}(\pi)$.*

Proof. Let π^* be ϵ-optimal with respect to $\overline{U}_{\widehat{\mathcal{D}}}(\pi)$ where $\widehat{\mathcal{D}}$ is the trajectory distribution under π on levels maximizing the worst-case regret of π. Let $\overline{\pi}^*$ be an optimal grounded policy. Then for any θ,

$$\overline{U}_\theta(\overline{\pi}^*) - \overline{U}_\theta(\pi^*) \leq \overline{U}_{\widehat{\mathcal{D}}}(\overline{\pi}^*) - \overline{U}_{\widehat{\mathcal{D}}}(\pi^*) \leq \epsilon \tag{7}$$

The first inequality follows from $\widehat{\mathcal{D}}$ being trajectories from levels that maximize worst-case regret with respect to π^*, and the second follows from π^* being ϵ-optimal on $\overline{U}_{\widehat{\mathcal{D}}}(\pi)$. Rearranging terms gives the desired condition. □

6 Experiments

Our experiments first focus on a discrete, stochastic binary choice task, with which we validate our theoretical conclusions by demonstrating that CICS can indeed lead to suboptimal policies. Moreover, we show that naive grounding suffices for learning robustly optimal policies in this setting. However, as we have argued, naive grounding gives up control of the aleatoric parameters Θ' and thus lacks the ability to actively sample scenarios helpful for learning robust behaviors—especially important when such scenarios are infrequent under the ground-truth distribution $\overline{P}(\Theta)$. SAMPLR induces potentially biased curricula, but retains optimality under $\overline{P}(\Theta)$ by matching transitions under $P(\Theta')$ with those under $\overline{P}(\Theta')$. We assess the effectiveness of this approach in our second experimental domain, based on the introductory example of driving icy roads. In this continuous-control driving domain, we seek to validate whether SAMPLR does in fact learn more robust policies that transfer to tail cases under $\overline{P}(\Theta')$, while retaining high expected performance on the whole distribution $\overline{P}(\Theta')$.

All agents are trained using PPO [39] with the best hyperparameters found via grid search using a set of validation levels. We provide extended descriptions of both environments alongside the full details of our architecture and hyperparameter choices in Appendix C.

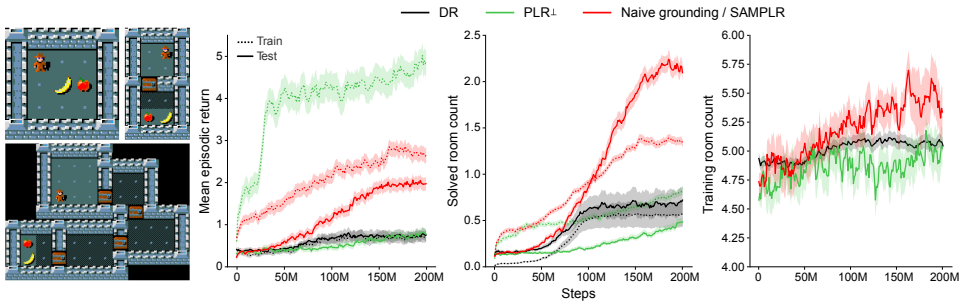

Figure 3: Left: Example Stochastic Fruit Choice levels. The plots show mean and standard error (over 10 runs) of episodic returns (left); room count of solved levels (middle), during training (dotted lines) and test on the ground-truth distribution (solid lines), for $q = 0.7$; and the room count of levels presented at training (right).

6.1 Stochastic Fruit Choice

We aim to demonstrate the phenomenon of CICS in Stochastic Fruit Choice, a binary choice task, where the aleatoric parameter determines the correct choice. This task requires the agent to traverse up to eight rooms, and in the final room, decide to eat either the apple or banana. The correct choice θ' is fixed for each level, but hidden from the agent. Optimal decision-making depends on the ground-truth distribution over the correct fruit, $\overline{P}(\Theta')$. This task benefits from a curriculum over the number of rooms, but a curriculum that selectively samples over both room layout and correct fruit choice can lead to suboptimal policies. Figure 3 shows example levels from this environment.

This domain presents a hard exploration challenge for RL agents, requiring robust navigation across multiple rooms. Further, this environment is built on top of MiniHack [36], enabling integration of select game dynamics from the NetHack Learning Environment [23], which the agent must master to succeed: To go from one room to the next, the agent needs to learn to kick the locked door until it opens. Upon reaching the final room, the agent must then apply the eat action on the correct fruit.

Let π_A be the policy that always chooses the apple, and π_B, the banana. If the probability that the goal is the apple is $\overline{P}(A) = q$, then the expected return is $R_A q$ under π_A and $R_B(1-q)$ under π_B. The optimal policy is π_A when $q > R_B/(R_A + R_B)$, and π_B otherwise. Domain randomization (DR), which directly samples each level $\theta \sim \overline{P}(\theta)$, optimizes for the correct ground-truth $\overline{P}(\Theta')$, but will predictably struggle to solve the exploration challenge. PLR$^\perp$ may induce curricula easing the exploration problem, but can be expected make the correct fruit choice oscillate throughout training to maximize regret, leading to problematic CICS.

We set $R_A = 3$, $R_B = 10$, and $q = 0.7$, making π_B optimal with an expected return of 3.0. We compare the train and test performance of agents trained with DR, PLR$^\perp$, and PLR$^\perp$ with naive grounding over 200M training steps in Figure 3. In this domain, SAMPLR reduces to naive grounding, as θ' only effects the reward of a terminal transition, making fictitious transitions equivalent to real transitions for all intermediate time steps. We see that DR struggles to learn an effective policy, plateauing at a mean return around 1.0, while PLR$^\perp$ performs the worst. Figure 6 in Appendix B shows that the PLR$^\perp$ curriculum exhibits much higher variance in q, rapidly switching the optimal choice of fruit to satisfy its regret-maximizing incentive, making learning more difficult. In contrast, PLR$^\perp$ with naive grounding constrains $q = 0.7$, while still exploiting a curriculum over an increasing number of rooms, as visible in Figure 6. This grounded curriculum results in a policy that solves more complex room layouts at test time. Figures 5 and 6 in Appendix B additionally show how the SAMPLR agent's choices converge to π_B and how the size of SAMPLR's improvement varies under alternative choices of q in $\{0.5, 0.3\}$.

6.2 Zero-Shot Driving Formula 1 Tracks with Black Ice

We now turn to a domain where the aleatoric parameters influence the distribution of τ_t at each t, thereby creating opportunities for a curriculum to actively sample specific θ' to promote learning on biased distributions of τ_t. We base this domain on the black ice driving scenario from the introduction of this paper, by modifying the CarRacingBezier environment in [20]. In our version, each track tile has black ice with probability q, in which case its friction coefficient is 0, making acceleration and

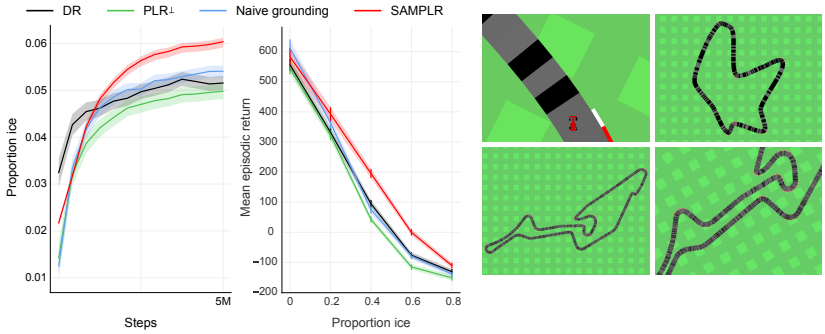

Figure 4: Charts show mean and standard error (over 10 runs) of fraction of visited tiles with ice during training (left) and zero-shot performance on the full Formula 1 benchmark as a function of ice rate (right). Top row screenshots show the agent approaching black ice ($q = 0.4$) and an example training track ($q = 0.6$). Bottom row shows a Formula 1 track ($q = 0.2$) at two zoom scales.

braking impossible. This task is especially difficult, since the agent cannot see black ice in its pixel observations. Figure 4 shows example tracks with ice rendered for illustration purposes. The episodic returns scale linearly with how much of the track is driven and how quickly this is accomplished. As success requires learning to navigate the challenging dynamics over ice patches, a curriculum targeting more difficult ice configurations should lead to policies more robust to black ice. Here, the ground-truth distribution $\overline{P}(\Theta')$ models the realistic assumption that most days see little to no ice. We therefore model the probability of ice per tile as $q \sim \text{Beta}(\alpha, \beta)$, where $\alpha = 1$, $\beta = 15$.

We test the hypothesis that SAMPLR's regret-maximizing curriculum results in policies that preserve optimal performance on the ground-truth distribution $\overline{P}(\Theta')$, while being more robust to tail cases compared to DR and PLR$^{\perp}$ with naive grounding. We expect standard PLR$^{\perp}$ to underperform all methods due to CICS, leading to policies that are either too pessimistic or too optimistic with respect to the amount of ice. These baselines provide the controls needed to distinguish performance changes due to the two grounding approaches and those due to the underlying curriculum learning method.

Table 1: Icy F1 returns, mean ± standard error over 10 runs.

Condition	DR	PLR	Naive	SAMPLR
Ground truth				
$q \sim \text{Beta}(1, 15)$	581 ± 23	543 ± 21	$\mathbf{618 \pm 6}$	$\mathbf{616 \pm 6}$
Zero-shot				
$q = 0.2$	$\mathbf{332 \pm 63}$	$\mathbf{323 \pm 60}$	363 ± 15	$\mathbf{393 \pm 13}$
$q = 0.4$	94.7 ± 41	43 ± 38	75 ± 39	$\mathbf{195 \pm 11}$
$q = 0.6$	-76.3 ± 24	-115 ± 12	-79 ± 25	$\mathbf{-1 \pm 17}$
$q = 0.8$	-131.1 ± 11	-151 ± 6.0	-139 ± 9	$\mathbf{-111 \pm 7}$

We train agents with each method for 5M and test zero-shot generalization performance on the Formula 1 (F1) tracks from the CarRacingF1 benchmark, extended to allow each track segment to have black ice with probability q in $\{0.0, 0.2, 0.4, 0.6, 0.8\}$. These test tracks are significantly longer and more complex than those seen at training, as well as having a higher rate of black ice.

To implement SAMPLR's belief model, we use a second simulator as a perfect model of the environment. At each time step, this second simulator, which we refer to as the *fictitious simulator*, resets to the exact physics state of the primary simulator, and its icy tiles are resampled according to the exact posterior over the aleatoric parameter $q = \theta'$, such that $\theta' \sim \overline{P}(\theta'|\tau)$, ensuring the future uncertainty is consistent with the past. The agent decides on action a_t based on the current real observation o_t, and observes the fictitious return r'_t and next state s'_{t+1} determined by the fictitious simulator after applying a_t in state $s'_t \sim \overline{P}(s'_t|\tau, \theta')$. This dual simulator arrangement, fully detailed in Appendix A.2, allows us to measure the impact of training on fictitious transitions independently of the efficacy of a model-based RL approach. Further, as the training environment in RL is most often simulation (e.g. in sim2real), this approach is widely applicable.

SAMPLR outperforms all baselines in zero-shot transfer to higher ice rates on the full F1 benchmark and attains a statistically significant improvement at $p < 0.001$ when transferring to $q = 0.4$ and $q = 0.6$, and $p < 0.05$ when $q = 0.8$. Importantly, SAMPLR outperforms PLR$^{\perp}$ with naive grounding, indicating that SAMPLR exploits specific settings of Θ' to better robustify the agent against rare icy conditions in the tail of $\overline{P}(\Theta')$. Indeed, Figure 4 shows that on average, SAMPLR exposes the agent to more ice per track tile driven, while PLR$^{\perp}$ underexposes the agent to ice compared to DR and naive grounding, suggesting that under PLR$^{\perp}$ agents attain higher regret on ice-free tracks—a likely outcome as ice-free tracks are easier to drive and lead to returns, with

which regret scales. Unfortunately, this results in PLR$^{\perp}$ being the worst out of all methods on the ground-truth distribution. SAMPLR and naive grounding avoid this issue by explicitly matching transitions to those under $\overline{P}$ at τ. As reported in Table 1, SAMPLR matches the baselines in mean performance across all F1 tracks under $\overline{P}(\Theta')$, indicating that despite actively sampling challenging θ', it preserves performance under $\overline{P}(\Theta')$, i.e. the agent does not become overly cautious.

7 Related Work

The mismatch between training and testing distributions of input features is referred to as *covariate shift*, and has long served as a fundamental problem for the machine learning community. Covariate shifts have been extensively studied in supervised learning [48, 18, 5, 2]. In RL, prior works have largely focused on covariate shifts due to training on off-policy data [44, 34, 11, 14, 13, 46] including the important case of learning from demonstrations [31, 33]. Recent work also aimed to learn invariant representations robust to covariate shifts [53, 54]. More generally, CICS is a form of sample-selection bias [15]. Previous methods like OFFER [7] considered correcting biased transitions via importance sampling [43] when optimizing for expected return on a single environment setting, rather than robust policies over all environments settings. We believe our work provides the first general formalization and solution strategy addressing curriculum-induced covariate shifts (CICS) for RL.

The importance of addressing CICS is highlighted by recent results showing curricula to be essential for training RL agents across many of the most challenging domains, including combinatorial gridworlds [55], Go [40], StarCraft 2 [49], and achieving comprehensive task mastery in open-ended environments [41]. While this work focuses on PLR$^{\perp}$, other methods include minimax adversarial curricula [30, 50, 51] and curricula based on changes in return [25, 32]. Curriculum methods have also been studied in goal-conditioned RL [12, 6, 42, 27], though CICS does not occur here as goals are observed by the agent. Lastly, domain randomization [DR, 35, 29] can be seen as a degenerate form of UED, and curriculum-based extensions of DR have also been studied [19, 47].

Prior work has also investigated methods for learning Bayes optimal policies under uncertainty about the task [56, 28], based on the framework of Bayes-adaptive MDPs (BAMDPs) [3, 10]. In this setting, the agent can adapt to an unknown MDP over several episodes by acting to reduce its uncertainty about the identity of the MDP. In contrast, SAMPLR learns a robustly Bayes-optimal policy for zero-shot transfer. Further unlike these works, our setting assumes the distribution of some aleatoric parameters is biased during training, which would bias the *a posteriori* uncertainty estimates with respect to the ground-truth distribution when optimizing for the BAMDP objective. Instead, SAMPLR proposes a means to correct for this bias assuming knowledge of the true environment parameters, to which we can often safely assume access in curriculum learning.

Deeply related, Off-Belief Learning [OBL, 16] trains cooperative agents in self-play using fictitious transitions assuming all past actions of co-players follow a base policy, e.g. a uniformly random one. Enforcing this assumption prevents agents from developing conventions that communicate private information to co-players via arbitrary action sequences. Such conventions hinder coordination with independently trained agents or, importantly, humans. SAMPLR can be viewed as adapting OBL to single-agent curriculum learning, where a co-player sets the environment parameters at the start of each episode (see Appendix D). This connection highlights how single-agent curriculum learning is inherently a multi-agent problem, and thus problems afflicting multi-agent learning also surface in this setting; moreover, methods addressing such issues in one setting can then be adapted to the other.

8 Conclusion

This work characterized how curriculum-induced covariate shifts (CICS) over aleatoric environment parameters Θ' can lead to suboptimal policies under the ground-truth distribution over these parameters, $\overline{P}(\Theta')$. We introduced a general strategy for correcting CICS, by training the agent on fictitious rewards and next states whose distribution is guaranteed to match what would be experienced under $\overline{P}(\Theta')$. Our method SAMPLR augments PLR$^{\perp}$ with this correction. By training on fictitious transitions, SAMPLR actively samples specific values of θ' that induce trajectories with greater learning potential, while still grounding the training data to $\overline{P}(\Theta')$. Crucially, our experiments in challenging environments with aleatoric uncertainty showed that SAMPLR produces robust policies outperforming those trained with competing baselines that do not correct for CICS.

A core assumption made by SAMPLR and all other UED methods is the ability to reset the environment to arbitrary configurations of some set of free parameters. While such resets can be difficult or impossible to perform in real world environments, in practice, this assumption is nearly always satisfied, as RL training largely occurs under a sim2real paradigm due to the additional costs of training in the wild. Most RL simulators can either be directly reset to specific environment configurations or be straightforwardly made to do so. SAMPLR thus provides a means to more fully exploit the affordances of a simulator to produce more robust policies: Policies trained with SAMPLR retain optimality when transferred to the ground-truth distribution of aleatoric parameters in the real environment—a crucial property not satisfied by prior UED methods. Importantly, the approach based on fictitious transitions used by SAMPLR can, in principle, be generally applied to prior UED methods to provide them with this desirable property.

Acknowledgments and Disclosure of Funding

We thank Robert Kirk and Akbir Khan for providing useful feedback on earlier drafts of this work. Further, we are grateful to our anonymous reviewers for their valuable feedback. MJ is supported by the FAIR PhD program at Meta AI. This work was funded by Meta.

References

[1] M. Andrychowicz, A. Raichuk, P. Stanczyk, M. Orsini, S. Girgin, R. Marinier, L. Hussenot, M. Geist, O. Pietquin, M. Michalski, S. Gelly, and O. Bachem. What matters in on-policy reinforcement learning? A large-scale empirical study. *CoRR*, abs/2006.05990, 2020.

[2] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *CoRR*, abs/1907.02893, 2019.

[3] R. Bellman. A problem in the sequential design of experiments. *Sankhyā: The Indian Journal of Statistics (1933-1960)*, 16(3/4):221–229, 1956.

[4] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840, 2002.

[5] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10:2137–2155, 2009.

[6] A. Campero, R. Raileanu, H. Kuttler, J. B. Tenenbaum, T. Rocktäschel, and E. Grefenstette. Learning with AMIGo: Adversarially motivated intrinsic goals. In *International Conference on Learning Representations*, 2021.

[7] K. A. Ciosek and S. Whiteson. OFFER: off-environment reinforcement learning. In S. P. Singh and S. Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 1819–1825. AAAI Press, 2017.

[8] M. Dennis, N. Jaques, E. Vinitsky, A. Bayen, S. Russell, A. Critch, and S. Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[9] A. Der Kiureghian and O. Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

[10] M. O. Duff. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. University of Massachusetts Amherst, 2002.

[11] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, S. Legg, and K. Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1407–1416. PMLR, 10–15 Jul 2018.

[12] C. Florensa, D. Held, X. Geng, and P. Abbeel. Automatic goal generation for reinforcement learning agents. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1515–1528. PMLR, 10–15 Jul 2018.

[13] C. Gelada and M. G. Bellemare. Off-policy deep reinforcement learning by bootstrapping the covariate shift. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3647–3655. AAAI Press, 2019.

[14] A. Hallak and S. Mannor. Consistent on-line off-policy evaluation. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1372–1383. PMLR, 06–11 Aug 2017.

[15] J. J. Heckman. Sample selection bias as a specification error. *Econometrica: Journal of the econometric society*, pages 153–161, 1979.

[16] H. Hu, A. Lerer, B. Cui, L. Pineda, N. Brown, and J. N. Foerster. Off-belief learning. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 4369–4379. PMLR, 2021.

[17] H. Hu, A. Lerer, A. Peysakhovich, and J. Foerster. "Other-play" for zero-shot coordination. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4399–4410. PMLR, 13–18 Jul 2020.

[18] J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. Smola. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19:601–608, 2006.

[19] N. Jakobi. Evolutionary robotics and the radical envelope-of-noise hypothesis. *Adaptive Behavior*, 6(2):325–368, 1997.

[20] M. Jiang, M. Dennis, J. Parker-Holder, J. Foerster, E. Grefenstette, and T. Rocktäschel. Replay-guided adversarial environment design. *Advances in Neural Information Processing Systems*, 34, 2021.

[21] M. Jiang, E. Grefenstette, and T. Rocktäschel. Prioritized level replay. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 4940–4950. PMLR, 2021.

[22] N. Justesen, R. R. Torrado, P. Bontrager, A. Khalifa, J. Togelius, and S. Risi. Procedural level generation improves generality of deep reinforcement learning. *CoRR*, abs/1806.10729, 2018.

[23] H. Küttler, N. Nardelli, A. H. Miller, R. Raileanu, M. Selvatici, E. Grefenstette, and T. Rocktäschel. The NetHack Learning Environment. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[24] X. Ma. Car racing with pytorch. https://github.com/xtma/pytorch_car_caring, 2019.

[25] T. Matiisen, A. Oliver, T. Cohen, and J. Schulman. Teacher-student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, PP, 07 2017.

[26] J. F. Nash et al. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

[27] O. OpenAI, M. Plappert, R. Sampedro, T. Xu, I. Akkaya, V. Kosaraju, P. Welinder, R. D'Sa, A. Petron, H. P. de Oliveira Pinto, A. Paino, H. Noh, L. Weng, Q. Yuan, C. Chu, and W. Zaremba. Asymmetric self-play for automatic goal discovery in robotic manipulation, 2021.

[28] I. Osband, D. Russo, and B. V. Roy. (more) efficient reinforcement learning via posterior sampling. In C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3003–3011, 2013.

[29] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. *CoRR*, abs/1710.06537, 2017.

[30] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, pages 2817–2826. PMLR, 2017.

[31] D. Pomerleau. ALVINN: an autonomous land vehicle in a neural network. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 1, [NIPS Conference, Denver, Colorado, USA, 1988]*, pages 305–313. Morgan Kaufmann, 1988.

[32] R. Portelas, C. Colas, K. Hofmann, and P.-Y. Oudeyer. Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments. In L. P. Kaelbling, D. Kragic, and K. Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 835–853. PMLR, 30 Oct–01 Nov 2020.

[33] S. Ross and D. Bagnell. Efficient reductions for imitation learning. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 661–668, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[34] M. Rowland, W. Dabney, and R. Munos. Adaptive trade-offs in off-policy learning. In S. Chiappa and R. Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 34–44. PMLR, 26–28 Aug 2020.

[35] F. Sadeghi and S. Levine. CAD2RL: real single-image flight without a single real image. In N. M. Amato, S. S. Srinivasa, N. Ayanian, and S. Kuindersma, editors, *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*, 2017.

[36] M. Samvelyan, R. Kirk, V. Kurin, J. Parker-Holder, M. Jiang, E. Hambro, F. Petroni, H. Kuttler, E. Grefenstette, and T. Rocktäschel. Minihack the planet: A sandbox for open-ended reinforcement learning research. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

[37] L. J. Savage. The theory of statistical decision. *Journal of the American Statistical association*, 46(253):55–67, 1951.

[38] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. In Y. Bengio and Y. LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017.

[40] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. P. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[41] A. Stooke, A. Mahajan, C. Barros, C. Deck, J. Bauer, J. Sygnowski, M. Trebacz, M. Jaderberg, M. Mathieu, N. McAleese, N. Bradley-Schmieg, N. Wong, N. Porcel, R. Raileanu, S. Hughes-Fitt, V. Dalibard, and W. M. Czarnecki. Open-ended learning leads to generally capable agents. *CoRR*, abs/2107.12808, 2021.

[42] S. Sukhbaatar, Z. Lin, I. Kostrikov, G. Synnaeve, A. Szlam, and R. Fergus. Intrinsic motivation and automatic curricula via asymmetric self-play. In *International Conference on Learning Representations*, 2018.

[43] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[44] R. S. Sutton, A. R. Mahmood, and M. White. An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of Machine Learning Research*, 17(73):1–29, 2016.

[45] Y. Tang, D. Nguyen, and D. Ha. Neuroevolution of self-interpretable agents. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2020.

[46] P. Thomas and E. Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2139–2148, New York, New York, USA, 20–22 Jun 2016. PMLR.

[47] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, pages 23–30. IEEE, 2017.

[48] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, pages 264–280, 1971.

[49] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, Ç. Gülçehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. P. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver. Grandmaster level in starcraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[50] R. Wang, J. Lehman, J. Clune, and K. O. Stanley. Paired open-ended trailblazer (POET): endlessly generating increasingly complex and diverse learning environments and their solutions. *CoRR*, abs/1901.01753, 2019.

[51] R. Wang, J. Lehman, A. Rawal, J. Zhi, Y. Li, J. Clune, and K. Stanley. Enhanced POET: Open-ended reinforcement learning through unbounded invention of learning challenges and their solutions. In *Proceedings of the 37th International Conference on Machine Learning*, pages 9940–9951, 2020.

[52] X. Wu, E. Dyer, and B. Neyshabur. When do curricula work? In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[53] A. Zhang, Z. C. Lipton, L. Pineda, K. Azizzadenesheli, A. Anandkumar, L. Itti, J. Pineau, and T. Furlanello. Learning causal state representations of partially observable environments. *CoRR*, abs/1906.10437, 2019.

[54] A. Zhang, R. T. McAllister, R. Calandra, Y. Gal, and S. Levine. Learning invariant representations for reinforcement learning without reconstruction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[55] V. Zhong, T. Rocktäschel, and E. Grefenstette. RTFM: generalising to new environment dynamics via reading. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[56] L. M. Zintgraf, K. Shiarlis, M. Igl, S. Schulze, Y. Gal, K. Hofmann, and S. Whiteson. Varibad: A very good method for bayes-adaptive deep RL via meta-learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Checklist

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] See Section 4.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix E.
   (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] See Section 5. In particular, we highlight that our proof of SAMPLR's robustly ϵ-Bayes optimal property assumes the underlying optimization procedure reaches an approximate Nash equilibrium.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section 5.
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] The code reproducing the experimental results is included in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 6 and Appendix C.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See figure captions in Section 6.
   (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] See compute estimates 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? [N/A]
   (b) Did you mention the license of the assets? [Yes] We include the license for the code reproducing our experiments in the supplemental material.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We include the code reproducing our experimental results 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]

A Algorithms

A.1 Robust Prioritized Level Replay

We briefly describe the PLR$^\perp$ algorithm and provide its pseudocode for completeness. For a detailed discussion of PLR$^\perp$ and its theoretical guarantees, we point the reader to [20].

For a given UPOMDP, PLR$^\perp$ induces a curriculum by using random search over the set of possible environment configurations θ to find the high regret levels for training. In order to estimate regret for level θ, PLR$^\perp$ uses an estimator **score**(τ, π) that is a function of the last trajectory on that level τ and the agent's policy π that generated τ. In practice these estimates are based on variations of the time-averaged value loss over τ. At the start of each episode with probability $1-p$, PLR$^\perp$ samples a new level θ from the training distribution $\overline{P}$, and uses the ensuing episode purely for evaluation, without training the agent on the collected experiences. It maintains a level buffer $\mathbf{\Lambda}$ of the top-K highest regret levels found throughout training. With probability p, episodes are used for training on levels sampled from this set, according to the *replay distribution* P_{replay}. Given staleness coefficient ρ, temperature β, a prioritization function h (e.g. rank), level buffer scores S, level buffer timestamps C, and the current episode count c (i.e. current timestamp), P_{replay} is defined as

$$
\begin{aligned}
P_{\text{replay}} &= (1-\rho)\cdot P_S + \rho \cdot P_C, \\
P_S &= \frac{h(S_i)^{1/\beta}}{\sum_j h(S_j)^{1/\beta}}, \\
P_C &= \frac{c - C_i}{\sum_{C_j \in C} c - C_j}.
\end{aligned}
$$

The first component P_S weighs each level based on its estimated regret, and the second, P_C, based on the age of the regret estimate. The staleness term mitigates regret values drifting off policy during training. The staleness coefficient ρ controls the contribution of each component. This process is summarized in Algorithm 2, and the details of how PLR$^\perp$ updates the top-K high regret levels in $\mathbf{\Lambda}$, in Algorithm 3.

Algorithm 2: Robust PLR (PLR$^\perp$)

Randomly initialize policy $\pi(\phi)$ and an empty level buffer, $\mathbf{\Lambda}$ of size K.
while *not converged* **do**
  Sample replay-decision Bernoulli, $d \sim P_D(d)$
  if $d = 0$ **then**
    Sample level θ from level generator
    Collect π's trajectory τ on θ, with a stop-gradient $\phi_\perp$
  else
    Use PLR to sample a replay level from the level store, $\theta \sim \mathbf{\Lambda}$
    Collect policy trajectory τ on θ and update π with rewards $\boldsymbol{R}(\tau)$
  end
  Compute PLR score, $S = \textbf{score}(\tau, \pi)$
  Update $\mathbf{\Lambda}$ with θ using score S
end

Algorithm 3: PLR level-buffer update rule

Input: Level buffer $\mathbf{\Lambda}$ of size K with scores S and timestamps C; level θ; level score S_θ; and current episode count c
if $|\mathbf{\Lambda}| < K$ **then**
  Insert θ into $\mathbf{\Lambda}$, and set $S(\theta) = S_\theta$, $C(\theta) = c$
else
  Find level with minimal support, $\theta_{\text{min}} = \arg\min_\theta P_{\text{replay}}(\theta)$
  if $S(\theta_{\text{min}}) < S_\theta$ **then**
    Remove θ_{min} from $\mathbf{\Lambda}$
    Insert θ into $\mathbf{\Lambda}$, and set $S(\theta) = S_\theta$, $C(\theta) = c$
    Update P_{replay} with latest scores S and timestamps C
  end
end

A.2 SAMPLR Implementation Details

In the car racing environment, the aleatoric parameters θ' determine whether each track tile contains black ice. Thus, the training distribution $P(\Theta')$ directly impacts the distribution over τ. In order to correct for the biased trajectories τ generated under its minimax regret curriculum, we must train the policy to maximize the ground-truth utility function conditioned on τ, $\overline{U}(\pi|\tau)$. SAMPLR

accomplishes this by training the policy on fictitious transitions that replace the real transitions observed. Each fictitious transition corresponds to the reward r'_t and s'_{t+1} that would be observed if the agent's action were performed in a level such that $\theta' \sim \overline{P}(\theta'|\tau)$. By training the agent on a POMDP whose future evolution conditioned on τ is consistent with $\overline{P}$, we ensure any optimal policy produced under the biased training distribution will also be optimal under $\overline{P}(\Theta')$.

Recall from Equation 6 that $\mathcal{B}(s'_t|\tau) = \sum_{\theta'} \overline{P}(s'_t|\tau, \theta')\overline{P}(\theta'|\tau)$. We can thus sample a fictitious state s'_t according to $\mathcal{B}$ by first sampling $\theta' \sim \overline{P}(\theta'|\tau)$, and then $s'_t \sim \overline{P}(s'_t|\tau, \theta')$. We implement SAMPLR for this domain by assuming perfect models for both the posterior $P(\theta'|\tau)$ and $\overline{P}(s'_t|\tau, \theta')$.

Simulating a perfect posterior over θ' is especially straightforward, as we assume each tile has ice sampled I.I.D. with probability $q \sim \text{Beta}(\alpha, \beta)$, where we make use of the conjugate prior. As τ contains the entire action-observation history up to the current time, it includes information that can be used to infer how much ice was already seen. In order to simulate a perfect posterior over θ', we thus track whether each visited track tile has ice and use these counts to update an exact posterior over q, equal to $\text{Beta}(\alpha + N_+, \beta + N_-)$, where N_+ and N_- correspond to the number of visited tiles with and without ice respectively. We then effectively sample $\theta' \sim \overline{P}(\theta'|\tau)$ by resampling all unvisited tiles from this posterior.

In order to sample from $\overline{P}(s'_t|\tau, \theta')$ and similarly from the grounded transition distribution $\overline{P}(s'_{t+1}|a_t, \tau, \theta')$, we make use of a second simulator we call the *fictitious simulator*, which acts as a perfect model of the environment. We could otherwise learn this model via online or offline supervised learning. Our design choice using a second simulator in place of such a model allows us to cleanly isolate the effects of SAMPLR's correction for CICS from potential errors due to the inherent difficulties of model-based RL.

Let us denote the primary simulator by $\mathcal{E}$, and the fictitious simulator by $\mathcal{E}'$. We ensure that the parameters of both simulators always match for $\theta \in \Theta \setminus \Theta'$. Before each training step, we first set the physics state of $\mathcal{E}'$ to that of $\mathcal{E}$ exactly, ensuring both simulators correspond to the same s_t, and then resample $\theta' \sim \overline{P}(\theta'|\tau)$ for the fictitious simulator as described above. We then take the resulting state of $\mathcal{E}'$ as s'_t. The agent next samples an action from its policy, $a_t \sim \pi(a_t|s'_t)$. Stepping forward $\mathcal{E}'$ in state s'_t with action a_t then produces a sample of s'_{t+1} from a perfect grounded belief model, along with the associated reward, r'_t. Throughout PPO training, the 1-step TD-errors δ_t for time t are computed using these fictitious transitions. Similarly, the PLR$^\perp$ mechanism underlying SAMPLR estimates regret using δ_t based on fictitious transitions.

B Additional Experimental Results

B.1 Stochastic Fruit Choice

When the probability of apple being the correct goal is $q = 0.7$, and the payoff for correctly choosing the apple (banana) is 3 (10), the optimal policy in expectation is to always choose the banana, resulting in an expected return of 3.0. Figure 5 shows that both DR and PLR$^\perp$ fail to learn to consistently eat the banana after 200M steps of training. In contrast, SAMPLR learns to always choose the banana when it successfully solves a level.

In additional experiments, we find that the impact of CICS varies with q. When $q = 10/13$, the expected returns for the policy always choosing apple (π_B) equals that for the policy always choosing banana (π_B). The top row of Figure 6 shows that the negative impact of CICS on PLR$^\perp$ and thus the benefits of SAMPLR diminish the farther q is from this equilibrium value. Intuitively, for q closer to the equilibrium value, smaller covariate shifts suffice to flip the policy, making it easier for PLR$^\perp$ to rapidly oscillate the optimal policy during training. We see in the bottom row of 6 that PLR$^\perp$ indeed produces large adversarial oscillations in q. This makes it difficult for the agent to settle on the optimal policy with respect to any ground-truth distribution. In contrast, SAMPLR grounds PLR's otherwise wild shifts in q with respect to its ground-truth value, allowing the agent to learn a well-grounded policy.

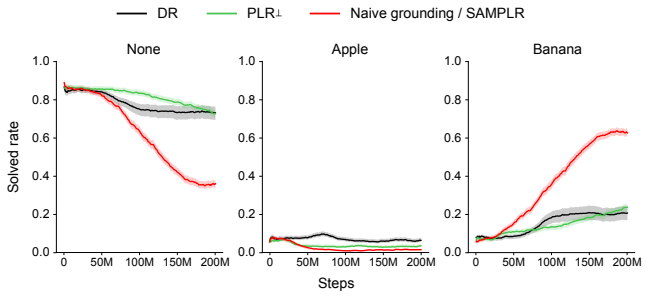

Figure 5: Left: Proportion of training episodes for $q = 0.7$ in which the agent fails to eat any fruit; eats the apple; or eats the banana. Right: Number of rooms in levels during training. Plots show mean and standard error of 10 runs.

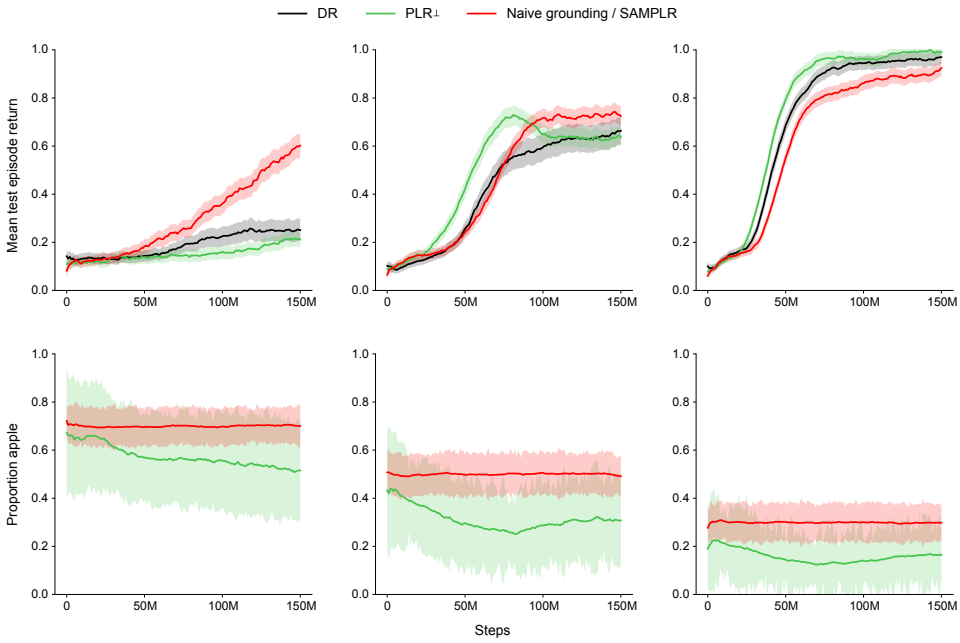

Figure 6: Top: Mean and standard error of episodic test returns as the probability q of the apple being the correct choice varies. Bottom: The proportion of training levels chosen by each method where apple is the correct choice. The mean and standard deviation are shown.

B.2 Car Racing with Black Ice

We see that on a gradient-update basis, SAMPLR and PLR$^{\perp}$ with naive grounding begin to overtake domain randomization much earlier in training. This is because PLR$^{\perp}$, does not perform gradient updates after each rollout, but rather only trains on a fraction p of episodes which sample high regret replay levels. In our experiments, we set the replay rate $p = 0.5$, meaning for the same number of training steps, the PLR$^{\perp}$-based methods only take half as many gradient steps as the DR baseline. Figure 7 shows each method's training and zero-shot transfer performance on validation F1 tracks with $q = 0.2$, as a function of the number of environment steps, while Figure 8 plots the same results as a function of the number of gradient updates performed. On a gradient basis, SAMPLR sees even more pronounced gains in robustness compared to DR—that is, SAMPLR learns more robust policies with less data.

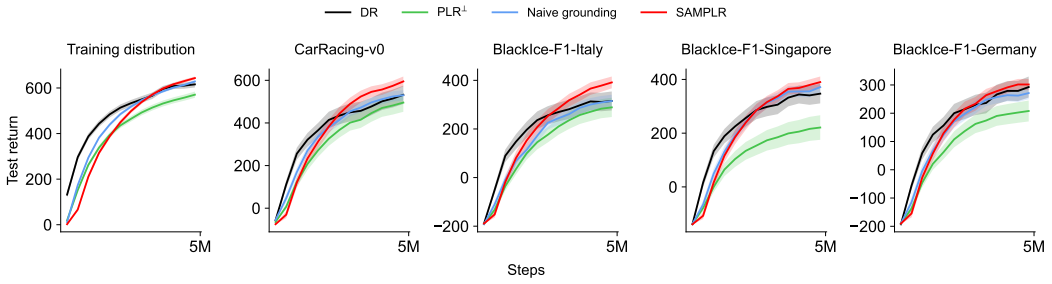

Figure 7: Performance of each method on the training distribution and zero-shot transfer environments with probability of ice per tile $q = 0.2$. These results show mean and standard error of episodic test returns as a function of number of environment steps collected during training.

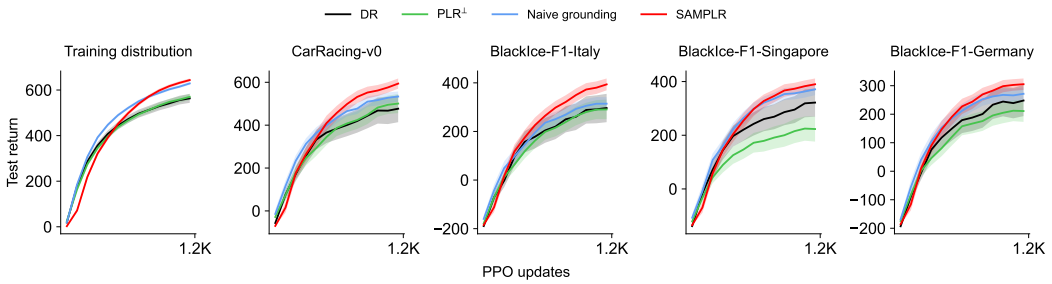

Figure 8: The same performance results reported in Figure 7, but as a function of number of gradient updates performed.

C Experimental Details and Hyperparameters

In this section, we provide detailed descriptions of each of the experimental domains featured in this work. In addition, we describe our architecture and hyperparameter choices for each setting. Table 2 summarizes our choice of hyperparameters for each domain.

C.1 Stochastic Fruit Choice

Environment details We make use of MiniHack [36], a library built on top of the NetHack Learning Environment [NLE, 23], for creating custom environments in the NetHack runtime. Our Stochastic Fruit Choice environment embeds a stochastic binary choice task within a challenging hard-exploration problem. The agent must navigate through up to eight rooms in each level, and in the final room, choose the correct piece of fruit, either the apple or banana to receive a reward. If the agent eats the wrong fruit for the level, it receives a reward of 0. With probability q, the apple is the correct fruit to eat. Eating either fruit terminates the episode. The episode also terminates once the budget of 250 steps is reached. Notably, passage into adjacent rooms requires first kicking down a locked door. As per NLE game dynamics, locked doors may require a random number of kicks before they give way. To complicate the learning of this kicking skill, kicking the stone walls of the room will lower the agent's health points; multiple misguided kicks can then lead to the agent dying, ending the episode.

The agent's observation consists of two primary elements: The nethack `glyph` and `blstats` tensors. The `glyph` tensor represents a 2D symbolic observation of the dungeon. This glyph tensor contains a 21×79 window of glyph identifiers, which can each be one of the 5991 possible glyphs in NetHack, which represent monsters, items, environment features, and other game entities. The `blstats` vector contains character-centric values, such as the agent's coordinates and the information in the "bottom-line stats" of the game, such as the agent's health stats, attribute levels, armor class, and experience points.

The action space includes the eight navigational actions, corresponding to moving toward each cell in the agent's Moore neighborhood, in addition to two additional actions for kicking (doors) and eating (apples and bananas).

Architecture We make use of the same agent architecture from [23]. The policy applies a ConvNet to all visible glyph embeddings and a separate ConvNet to a 9×9 egocentric crop around the agent—which was found to improve generalization—producing two latent vectors. These are then concatenated with an MLP encoding of the `blstats` vector, the resulting vector is further processed by an MLP layer, and finally, input through an LSTM to produce the action distribution. We used the policy architecture provided in `https://github.com/facebookresearch/nle`.

Choice of hyperparameters Our choice of PPO hyperparameters, shared across all methods, was based on a grid search, in which we train agents with domain randomization on a 15×15 maze, in which the goal location and initial agent location, along with up to 50 walls, are randomly placed. For each setting, we average results over 3 training runs. We chose this environment to perform the grid search, as it allows for significantly faster training than the multi-room environment featured in our main experiments. Specifically, we swept over the following hyperparameter values: number of PPO epochs in $\{5, 20\}$, number of PPO minibatches in $\{1, 4\}$, PPO clip parameter in $\{0.1, 0.2\}$, learning rate in $\{$1e-3, 1e-4$\}$, and discount factor γ in $\{0.99, 0.995\}$. Fixing these hyperparameters to the best setting found, we then performed a separate grid search over PLR's replay rate p in $\{0.5, 0.95\}$ and replay buffer size in $\{4000, 5000, 8000\}$, evaluating settings based on evaluation levels sampled via domain randomization after 50M steps of training.

Compute All agents in the Stochastic Fruit Choice environment were trained using Tesla V100 GPUs and Intel Xeon E5-2698 v4 CPUs. DR, PLR$^{\perp}$, and naive grounding all required approximately 24 hours to complete the 50M training steps for each run of the hyperparameter sweep and 192 hours to complete the 200 million training steps for each full multi-room run. In total, our experimental results, including hyperparameter sweeps, required roughly 8,500 hours (around 350 days) of training.

C.2 Car Racing with Black Ice

Environment details This environment extends the CarRacingBezier environment from [20] to optionally contain black ice on each track tile with probability q, where for a given track, q may first be sampled from an arbitrary prior distribution and after which, ice is sampled I.I.D. per tile. It is impossible to accelerate or brake over ice tiles, which have a friction coefficient of 0, making icy settings much more challenging. Like in CarRacingBezier, each track shape is generated by sampling a random set of 12 control points that define a closed Bézier curve. For a track composed of L tiles, the agent receives a reward of $1000/L$ for each tile visited, and a per-step penalty of -0.1. Like [20], we adopt the methodology outlined in [24] and do not penalize the agent for driving out of bounds, terminate episodes early when the agent drives too far off the track, and repeat each action for 8 frames. At each time step, the agent receives a $96 \times 96 \times 3$ RGB frame consisting of a local birdseye view of the car on the track and a dashboard showing the agent's latest action and return. Actions take the form of a three-dimensional, continuous vector. Each component represents control values for torque in $[-1.0, 1.0]$, acceleration in $[0.0, 1.0]$, and deceleration in $[0.0, 1.0]$.

Architecture We adopt a policy architecture used across several prior works [20, 24, 45], consisting of a stack of 2D convolutions feeding into a fully-connected ReLU layer. The convolutions have square kernels of size 2, 2, 2, 2, 3, 3, output channels of dimension 8, 16, 32, 64, 128, 256, and stride lengths of 2, 2, 2, 2, 1, 1. The resulting 256-dimensional embedding is then fed into alpha, beta, and value heads. The alpha and beta heads are each fully-connected softplus layers, to whose outputs we add 1 to produce the α and β parameters for the Beta distribution parameterizing each action dimension (i.e. each action is sampled from Beta(α, β) and then translated into the appropriate range). The value head is a fully-connected ReLU layer. All hidden layers are 100-dimensional.

Choice of hyperparameters We selected hyperparameters based on evaluation performance over 5 episodes on the Italy, Singapore, and Germany F1 tracks with ice probability per tile fixed to $q = 0.2$, when trained under the ice distribution featured in our main results, where $q \sim$ Beta(1,15). For each setting, we averaged results over 3 runs. PPO hyperparameters, shared across methods, were selected based on the performance of agents trained with domain randomization across settings in a grid search covering learning rate in $\{0.001, 0.0003, 0.0001, 0.00001\}$, number of epochs in $\{3, 8\}$, number of minibatches in $\{2, 4, 16\}$, and value loss coefficient in $\{0.5, 1.0, 2.0\}$. The remaining PPO

hyperparameters, as well as PLR$^{\perp}$-specific hyperparameters were based on those used in [20], with the exception of a smaller level buffer size, which we found helped improve validation performance. Additionally, for each method, we also swept over the choice of whether to initialize the policy to ensure actions are initially close to zero. Initializing the policy in this way has been shown to reduce variance in performance across seeds [1].

Compute All car racing agents were trained using Tesla V100 GPUs and Intel Xeon E5-2698 v4 CPUs. DR, PLR$^{\perp}$, and naive grounding all required approximately 24 hours to complete the 5M training steps in each experiment run, while SAMPLR required 42 hours. In total, our experimental results, including hyperparameter sweeps, required roughly 11,500 hours (around 480 days) of training.

Table 2: Hyperparameters used for training each method.

Parameter	Stochastic Fruit Choice	Black-Ice Car Racing
PPO		
γ	0.995	0.99
λ_{GAE}	0.95	0.9
PPO rollout length	256	125
PPO epochs	5	3
PPO minibatches per epoch	1	4
PPO clip range	0.2	0.2
PPO number of workers	32	16
Adam learning rate	1e-4	1e-4
Adam ϵ	1e-5	1e-5
PPO max gradient norm	0.5	0.5
PPO value clipping	yes	no
return normalization	no	yes
value loss coefficient	0.5	1.0
entropy coefficient	0.0	0.0
PLR$^{\perp}$ and *SAMPLR*		
Replay rate, p	0.95	0.5
Buffer size, K	4000	500
Scoring function	positive value loss	positive value loss
Prioritization	rank	power
Temperature, β	0.3	1.0
Staleness coefficient, ρ	0.3	0.7

D Connection to Off-Belief Learning

In cooperative multi-agent reinforcement learning (MARL), self-play promotes the formation of cryptic conventions—arbitrary sequences of actions that allow agents to communicate information about the environment state. These conventions are learned jointly among all agents during training, but are arbitrary and hence, indecipherable to independently-trained agents or humans at test time. Crucially, this leads to policies that fail to perform zero-shot coordination [ZSC, 17], where independently-trained agents must cooperate successfully without additional learning steps—a setting known as ad-hoc team play. Off-Belief Learning [OBL; 16] resolves this problem by forcing agents to assume their co-players act according to a fixed, known policy π_0 until the current time t, and optimally afterwards, conditioned on this assumption. If π_0 is playing uniformly random, this removes the possibility of forming arbitrary conventions.

Formally, let G be a decentralized, partially-observable MDP [Dec-POMDP, 4], with state s, joint action a, observation function $\mathcal{I}^i(s)$ for each player i, and transition function $\mathcal{T}(s, a)$. Let the historical trajectory $\tau = (s_1, a_1, ...a_{t-1}, s_t)$, and the action-observation history (AOH) for agent i be $\tau^i = (\mathcal{I}^i(s_1), a_1, ..., a_{t-1}, \mathcal{I}^i(s_t))$. Further, let π_0 be an arbitrary policy, such as a uniformly random policy, and $\mathcal{B}_{\pi_0}(\tau|\tau^i) = P(\tau_t|\tau^i_t, \pi_0)$, a belief model predicting the current state, conditioned on the AOH of agent i and the assumption of co-players playing policy π_0 until the current time t, and optimally according to π_1 from t and beyond. OBL aims to find the policy π_1 with the optimal, *counter-factual value function*,

$$V^{\pi_0 \to \pi_1}(\tau^i) = \mathbb{E}_{\tau \sim \mathcal{B}_{\pi_0}(\tau^i)}\left[V^{\pi_1}(\tau)\right]. \tag{8}$$

Thus, the agent conditions its policy on the realized AOH τ^i, while optimizing its policy for transition dynamics based on samples from $\mathcal{B}_{\pi_0}$, which are consistent with the assumption that co-players play according to π_0 until time t. Therefore, if π_0 is a uniformly random policy, π_1 can no longer benefit from conditioning on the action sequences of its co-players, thereby preventing the formation of cryptic conventions that harm ZSC.

Similarly, in single-agent curriculum learning, we can view the UED teacher as a co-player that performs a series of environment design decisions that defines the environment configuration θ at the start of the episode, and subsequently performs no-ops for the remainder of the episode. As discussed in Section 3, curriculum-induced covariate shifts (CICS) can cause the final policy to be suboptimal with respect to the ground-truth distribution $\overline{P}$ when the teacher produces a curriculum resulting in the training distribution of aleatoric parameters $P(\Theta')$ deviating from the ground-truth distribution $\overline{P}(\Theta')$. We then see that the fictitious transitions used by SAMPLR are equivalent to those used by OBL, where the belief model $\mathcal{B}$ assumes the teacher makes its design choices such that the resulting distribution of aleatoric parameters Θ' matches the ground-truth $\overline{P}(\Theta')$. This connection reveals that SAMPLR can be viewed as an adaptation of OBL to the single-agent curriculum learning setting, whereby the UED teacher, which designs the environment configuration, is viewed as the co-player.

E Broader Impact Statement

SAMPLR trains policies that are optimal under some ground-truth distribution $\overline{P}$ despite using data sampled from a curriculum distribution that may be biased with respect to $\overline{P}$. However, often in practice, our knowledge of $\overline{P}$ may be based on estimations that are themselves biased. For example, when applying SAMPLR with respect to an estimated ground-truth distribution of user demographics, the resulting policy may still be biased toward some demographics, if the ground-truth estimation is biased. Therefore, when applying SAMPLR, we must still take care to ensure our notion of the ground truth is sound.