Feasible Adversarial Robust Reinforcement Learning for Underspecified Environments

JB Lanier, Stephen McAleer, Pierre Baldi, Roy Fox

Carnegie Mellon University

UCI University of California, Irvine

Overview

- **Robust Reinforcement Learning** maximizes worst-case performance in parameterizable environments. It can be challenging to apply Robust RL to complex configurable environments without current methods focusing on **unrealistically difficult task variations** that break learning. You need to carefully tune which variations are allowed in training to fix this.
- Feasible Adversarial Robust Reinforcement learning (FARR), **automatically tunes the set of allowed task variations we can train on by filtering based on a target difficulty** so that all training tasks are feasible.
- FARR produces agents that are **more robust** to environment task variations within a target difficulty than existing alternatives like minimax, domain randomization, and regret [1] objectives.

Method Description

Task performing protagonist and task selecting adversary agents play a two-player zero-sum game. The adversary selects the hardest task variations but is penalized if a pre-specified threshold reward of at least λ can't be achieved by a best-response agent:

$$U_p^{\lambda}(\pi_p,\theta)=\begin{cases}C & \text{if } U_p(\mathbb{BR}(\theta),\theta)<\lambda \quad \textit{(Infeasible Task Penalty)}\\ U_p(\pi_p,\theta) & \text{otherwise.} \quad \textit{(Normal RARL Minimax Utility)}\end{cases}$$

We find approximate Nash equilibria to the FARR game using a variant of Policy-Space Response Oracles (PSRO)[2]:

Algorithm 1 FARR Optimized through PSRO

```
Input: Initial policy sets $\Pi = (\Pi_p, \Pi_\theta)$ for Protagonist player and Adversary player
Compute expected FARR payoff matrix $U_\lambda^\Pi$ as utilities $U_p^\lambda(\pi_p, \theta)$ for each joint $(\pi_p, \theta) \in \Pi$
repeat
  Compute Normal-Form restricted NE $\sigma = (\sigma_p, \sigma_\theta)$ over population policies $\Pi$ using $U_\lambda^\Pi$
  Calculate new Protagonist policy $\pi_p$ (e.g. $\mathbb{BR}(\sigma_\theta)$)
  $\Pi_p = \Pi_p \cup \{\pi_p\}$
  for at least one iteration do
    Calculate new Adversary strategy $\theta$ and associated estimator for $\mathbb{BR}(\theta)$
    $\Pi_\theta = \Pi_\theta \cup \{\theta\}$
  end for
  Compute missing entries in $U_\lambda^\Pi$ from $\Pi$
until terminated early or no novel policies can be added
Output: current Protagonist restricted NE strategy $\sigma_p$
```

Experiments and Results

FARR produces a Nash equilibrium equivalent to robust RL only on tasks with a target minimum achievable reward λ:

🟩 = Pure Strategy NE 🟥 = Penalized Adversary Strategy

Original Game

		Adversary		
		Left	Middle	Right
Protagonist	Grab	20	10	-10
	Don't Grab	0	0	🟩 0

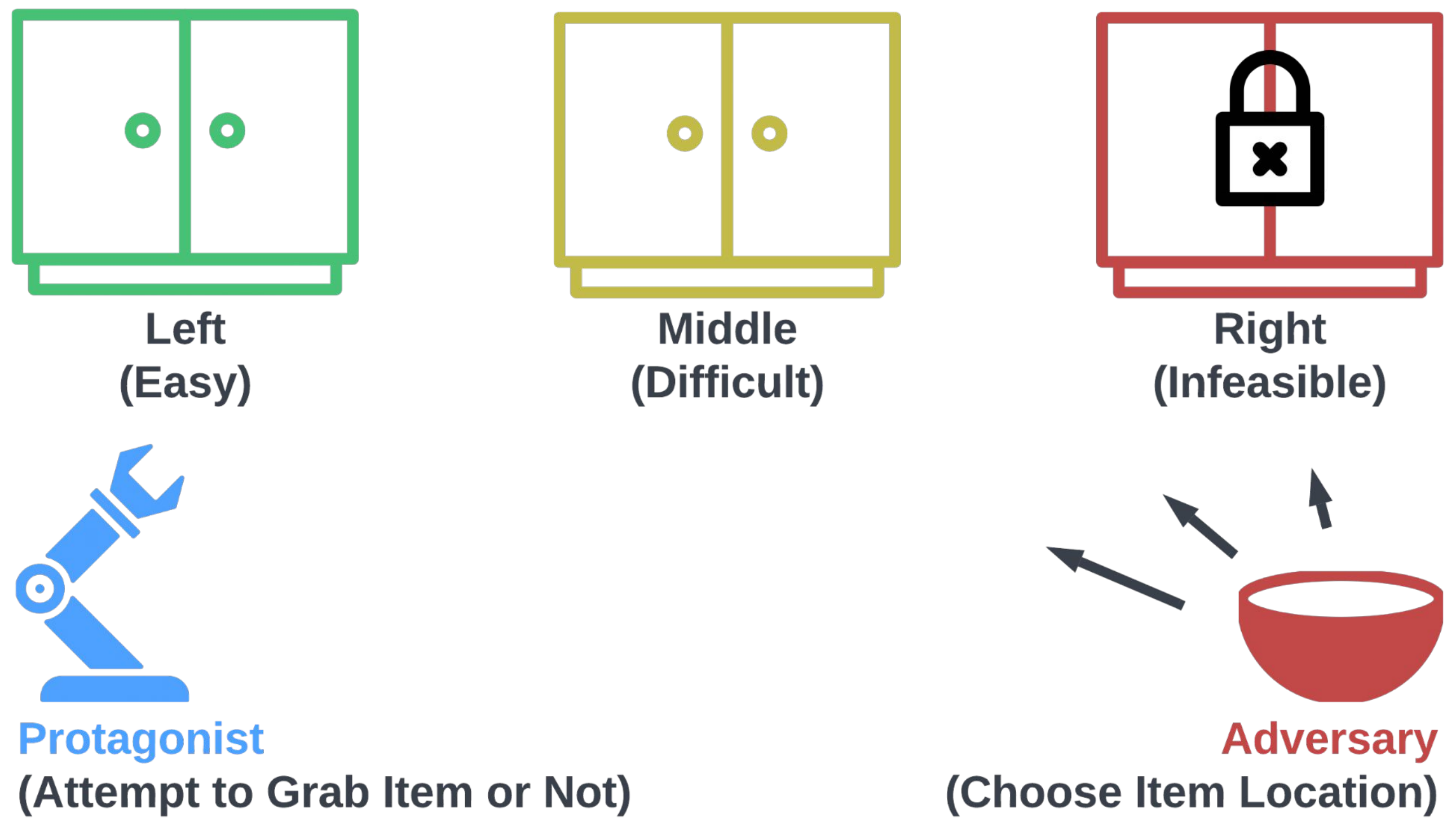
(Infeasible Adversary NE) ✗

→

FARR Game (λ = 1, C = 500)

		Adversary		
		Left	Middle	Right
Protagonist	Grab	20	🟩 10	🟥 500
	Don't Grab	0	0	🟥 500

(Feasible Adversary NE) ✓



Left **(Easy)**

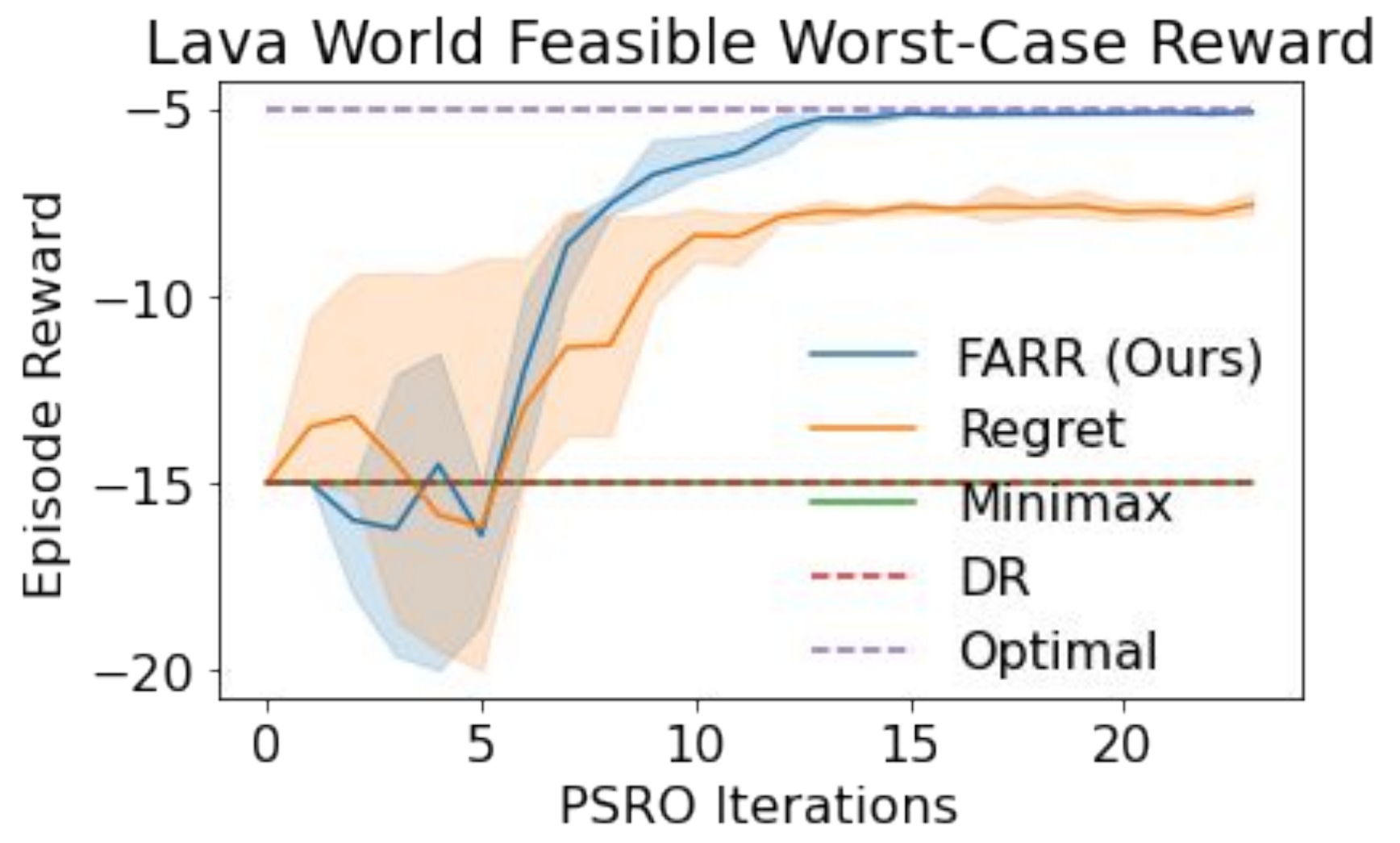
Middle **(Difficult)**

Right **(Infeasible)**

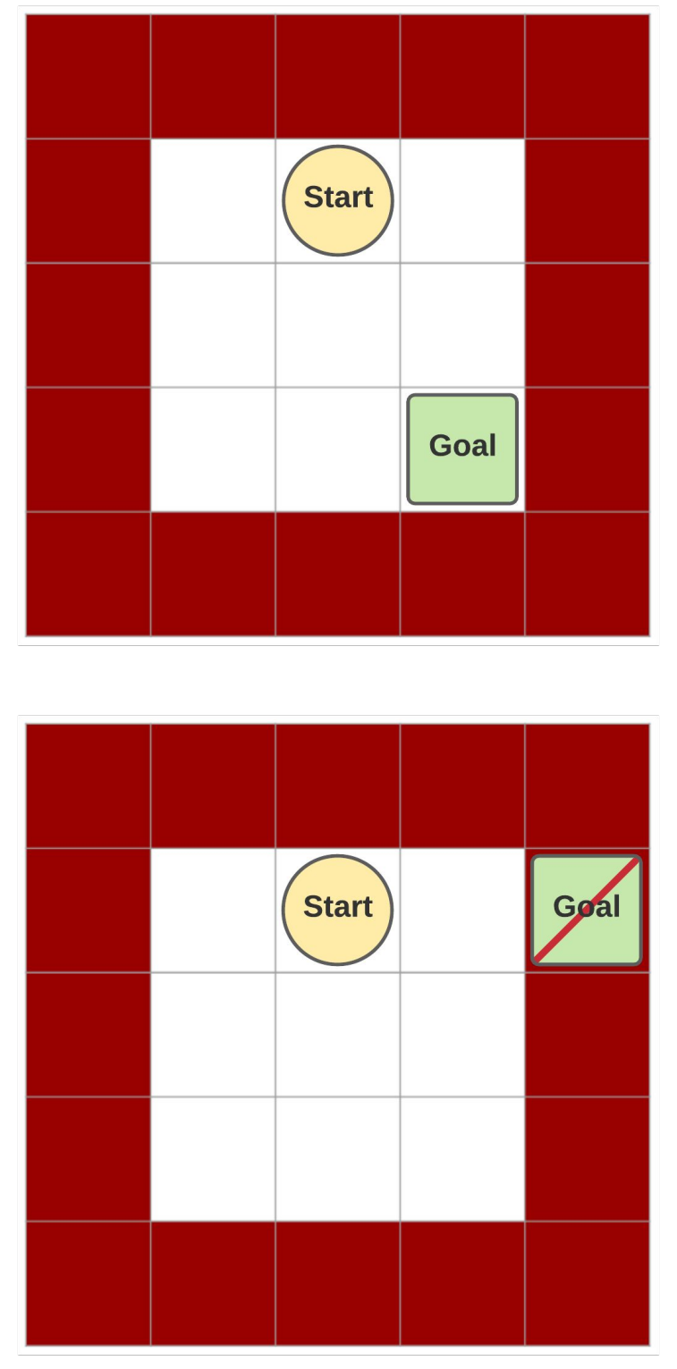
Protagonist **(Attempt to Grab Item or Not)**

Adversary **(Choose Item Location)**

Instead of tuning environment rules to prevent overly difficult tasks, FARR allows effective minimax robust RL learning only over "λ-feasible" tasks given a desired max level of difficulty:

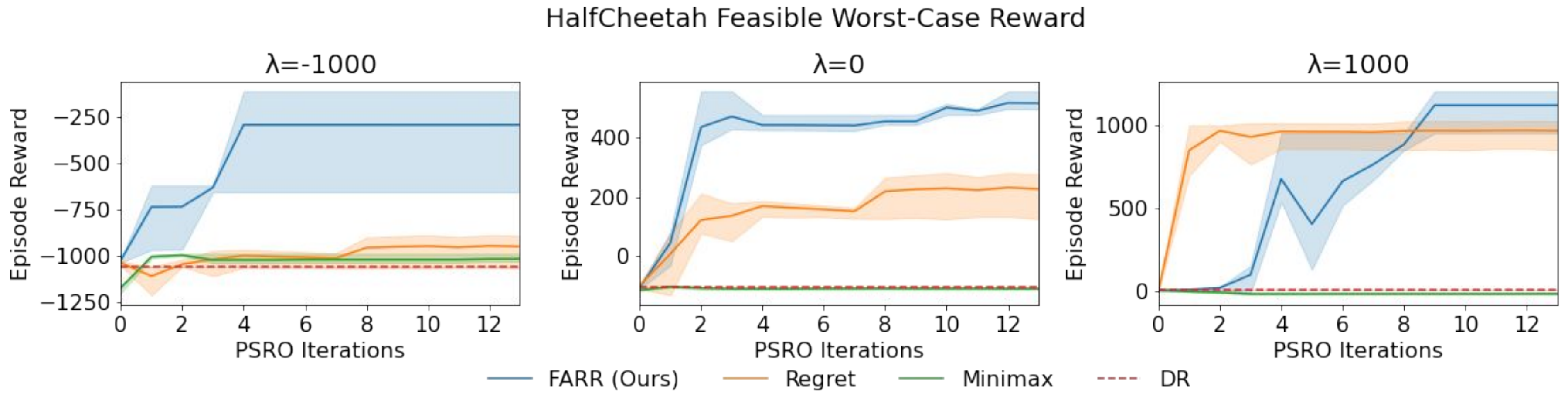


Lava World Feasible Worst-Case Reward

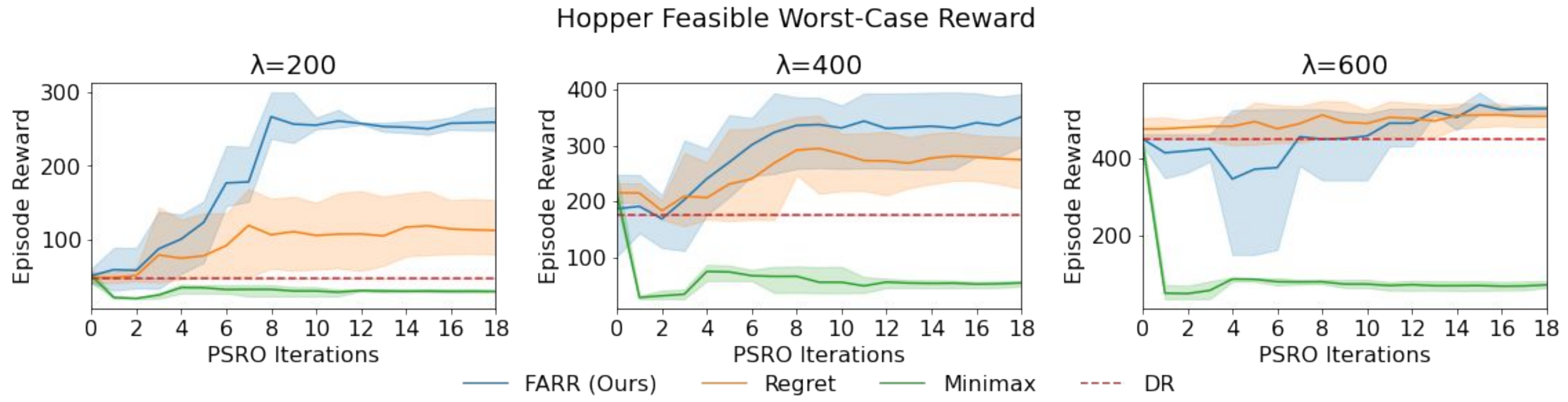


Adversary provides hidden goals. Goals in lava are impossible, and we want an agent robust to doable goals.

These perturbed MuJoCo environments can be made insurmountably difficult. For any target task difficulty λ, FARR avoids producing unwanted infeasible tasks and trains a protagonist agent robust to the feasible tasks for any given λ:

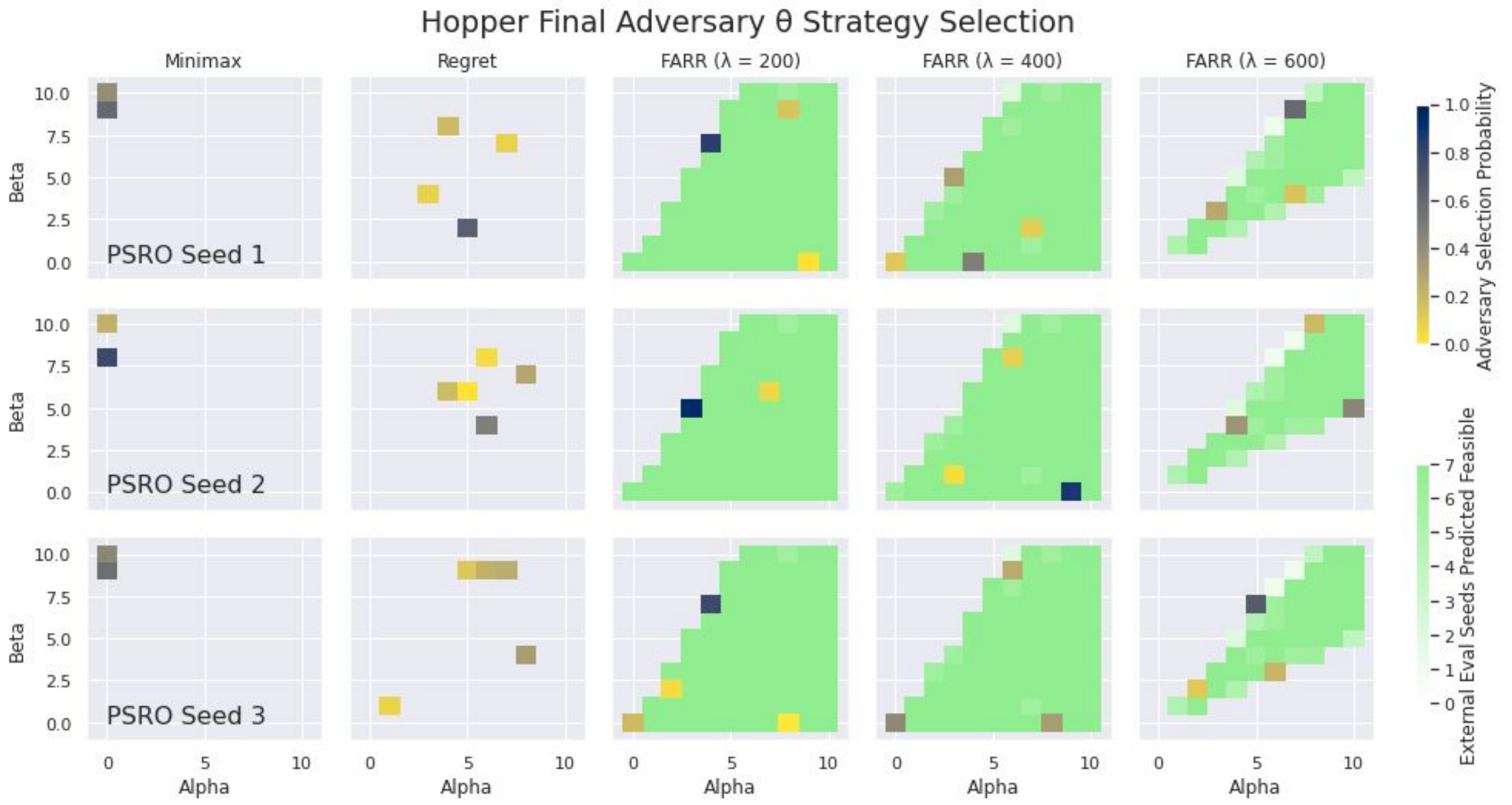


HalfCheetah Feasible Worst-Case Reward



Hopper Feasible Worst-Case Reward

Actual feasible tasks for a target minimum achievable reward λ are in **green**. Traditional robust RL minimax produces overly difficult task mixtures outside the feasible set of tasks. FARR produces a **worst-case distribution inside the target feasible set**:



Hopper Final Adversary θ Strategy Selection

[1] Dennis, Michael, et al. "Emergent Complexity and Zero-shot Transfer via Unsupervised Environment Design." NeurIPS 33 (2020).
[2] Lanctot, Marc, et al. "A Unified Game-Theoretic Approach to Multiagent Reinforcement Learning." NeurIPS 30 (2017).