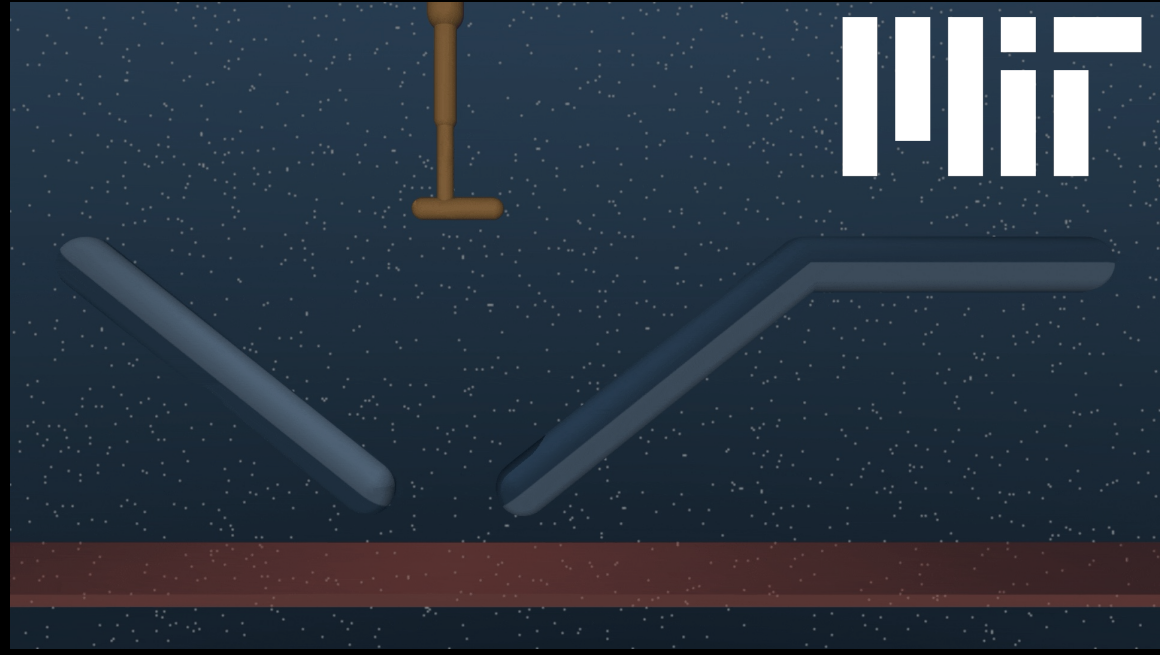


Solving Parameter-Robust Avoid Problems with Unknown Feasibility using Reinforcement Learning

Oswin So[*†], **Eric Yang Yu**[*†], Songyuan Zhang[†], Matthew Cleaveland[‡], Mitchell Black[‡], and Chuchu Fan[†]

[†]Department of Aeronautics and Astronautics, MIT

[‡]MIT Lincoln Laboratory

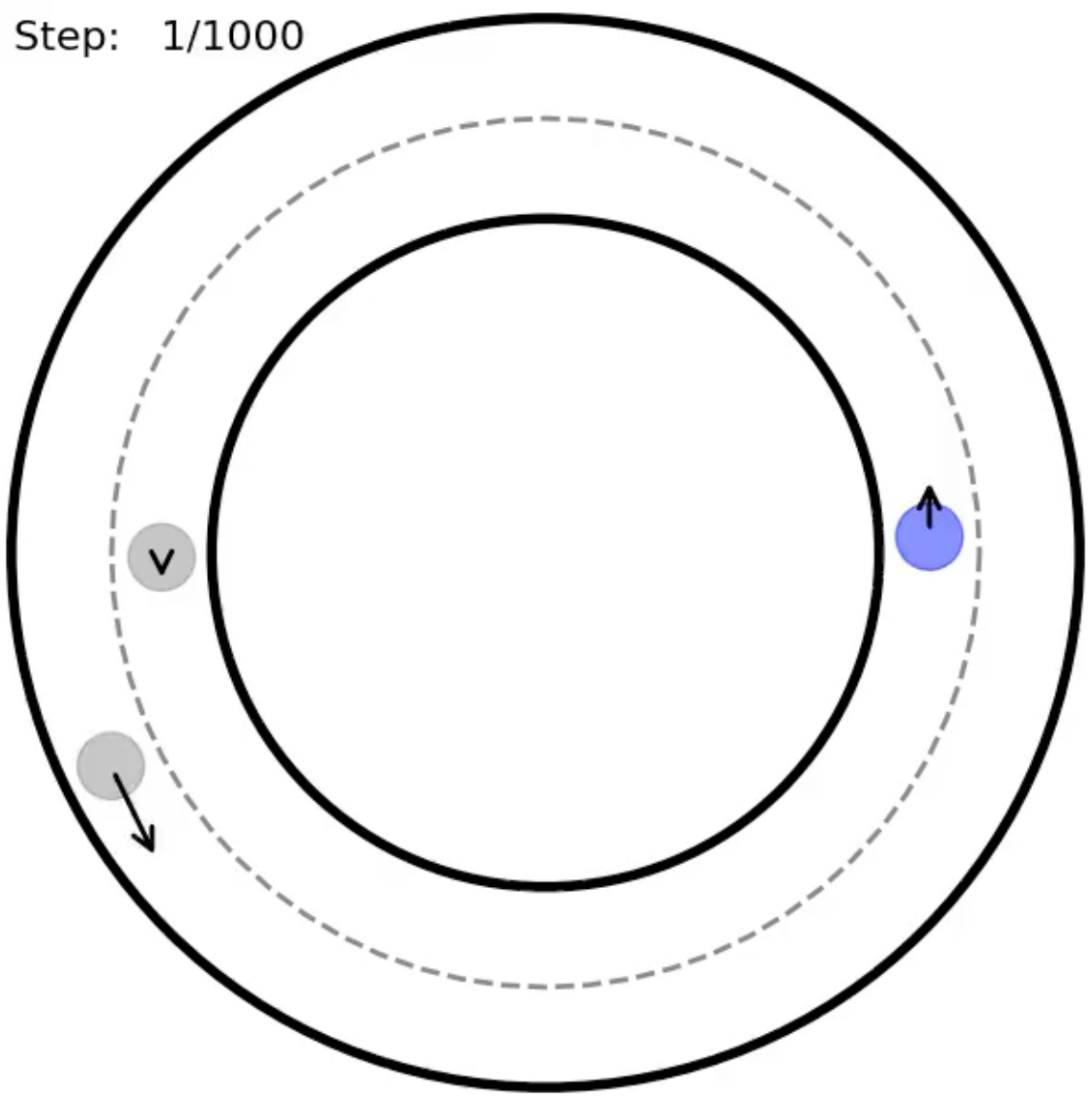
[*]Equal Contribution

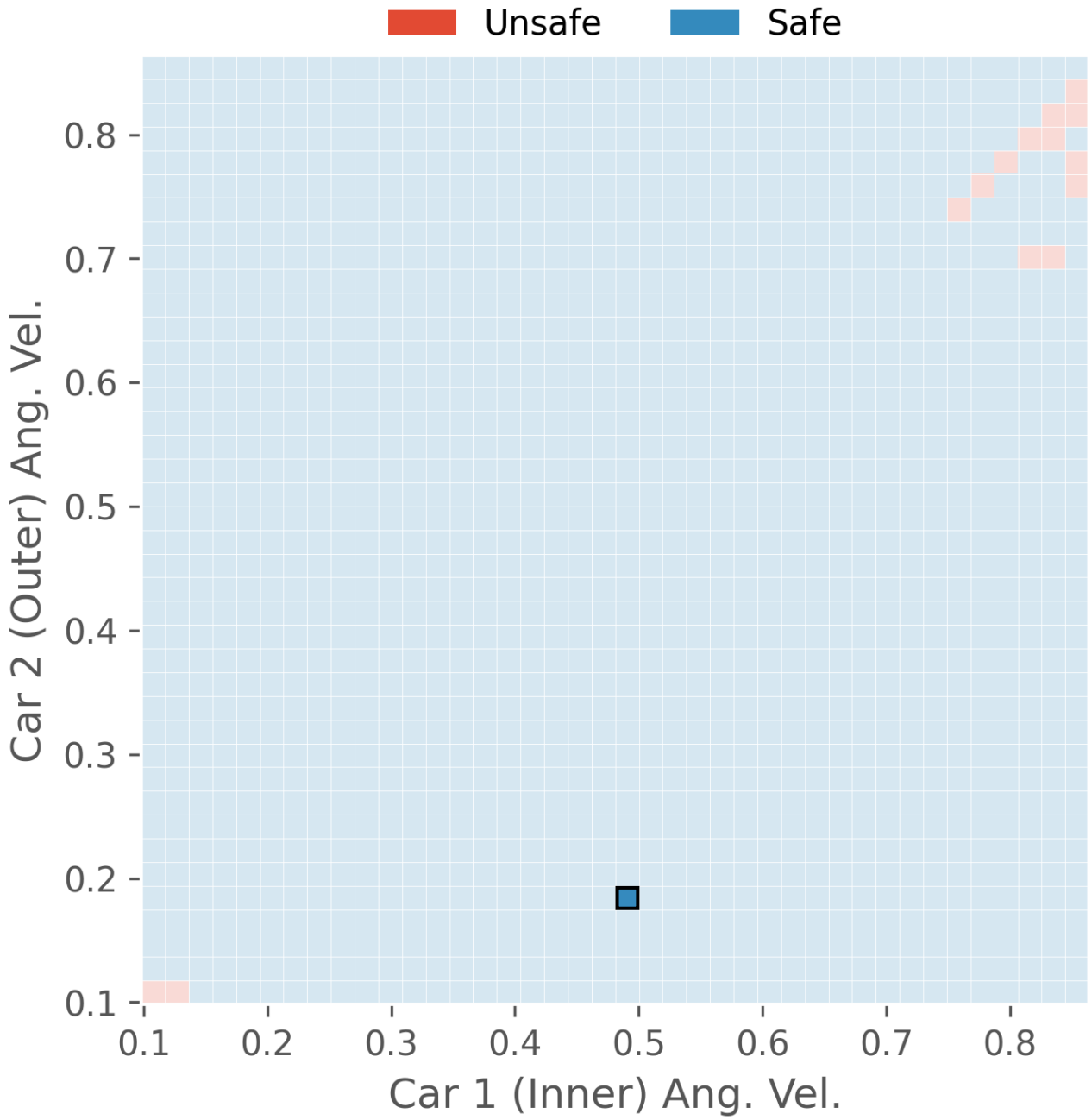
ICLR

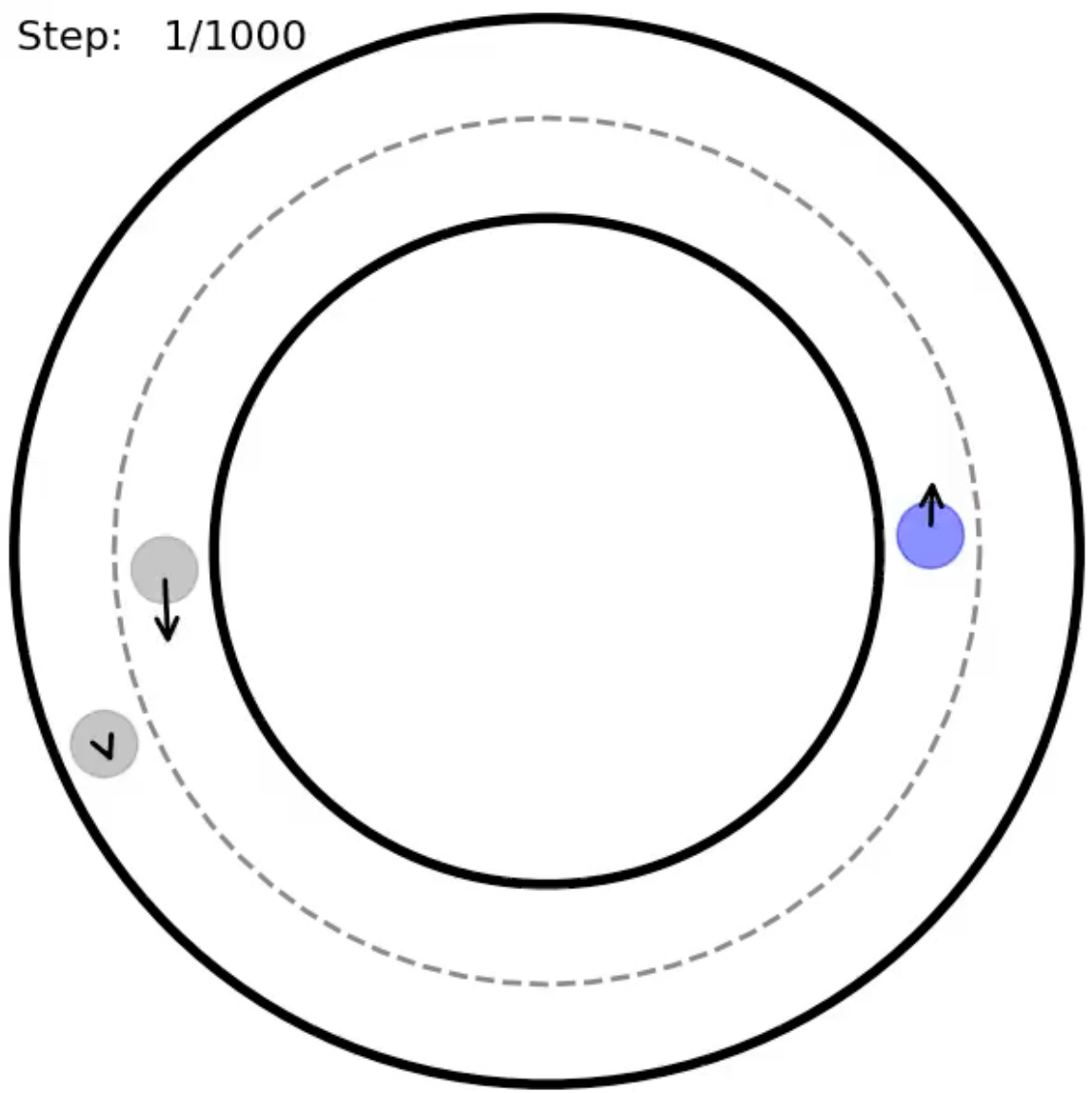
Safety Specification:

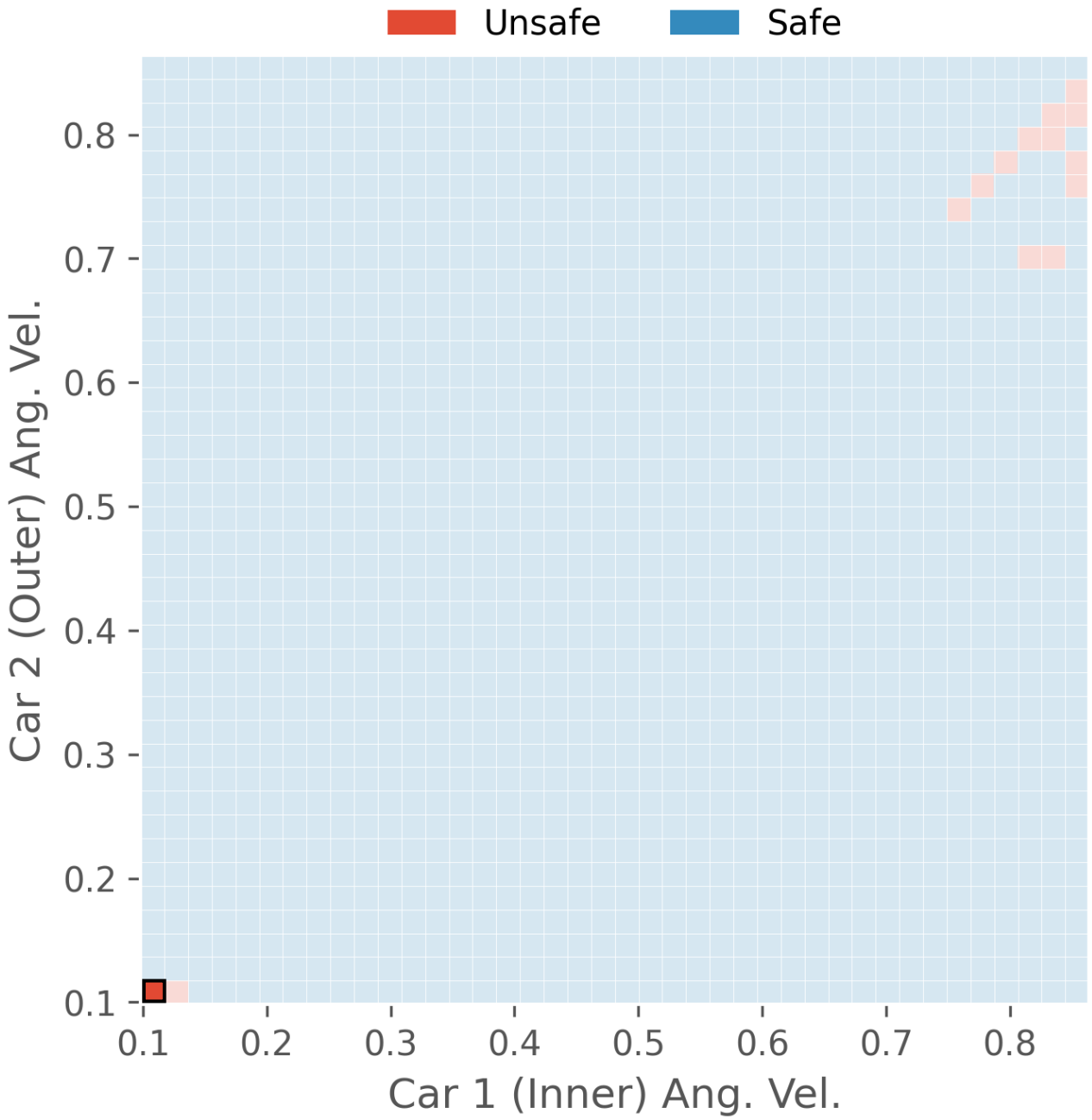
- Stay inside road
- Don’t collide
- Stay within velocity limits

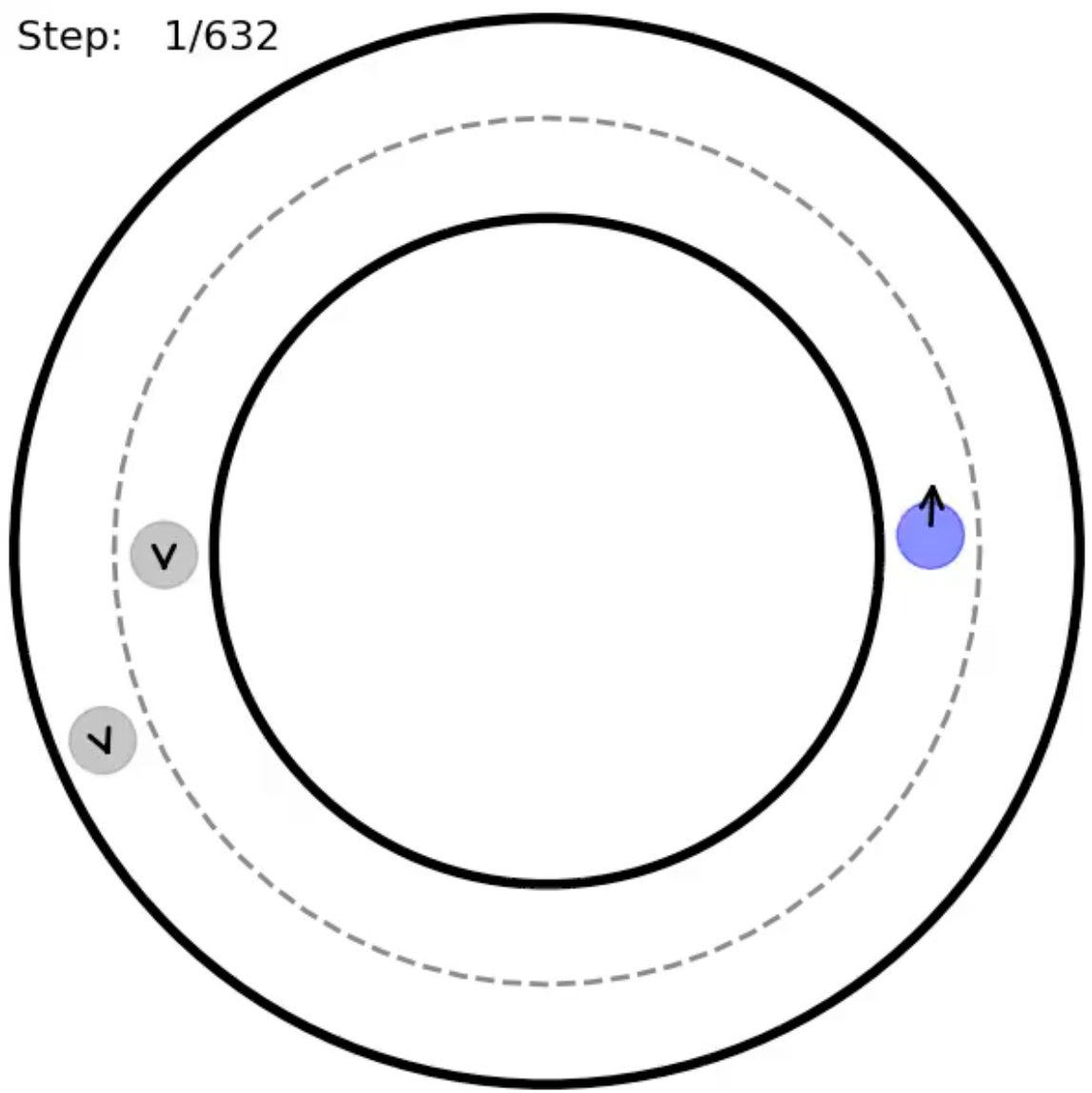
Be as robust as possible to the other cars’ velocities (**parameters**)

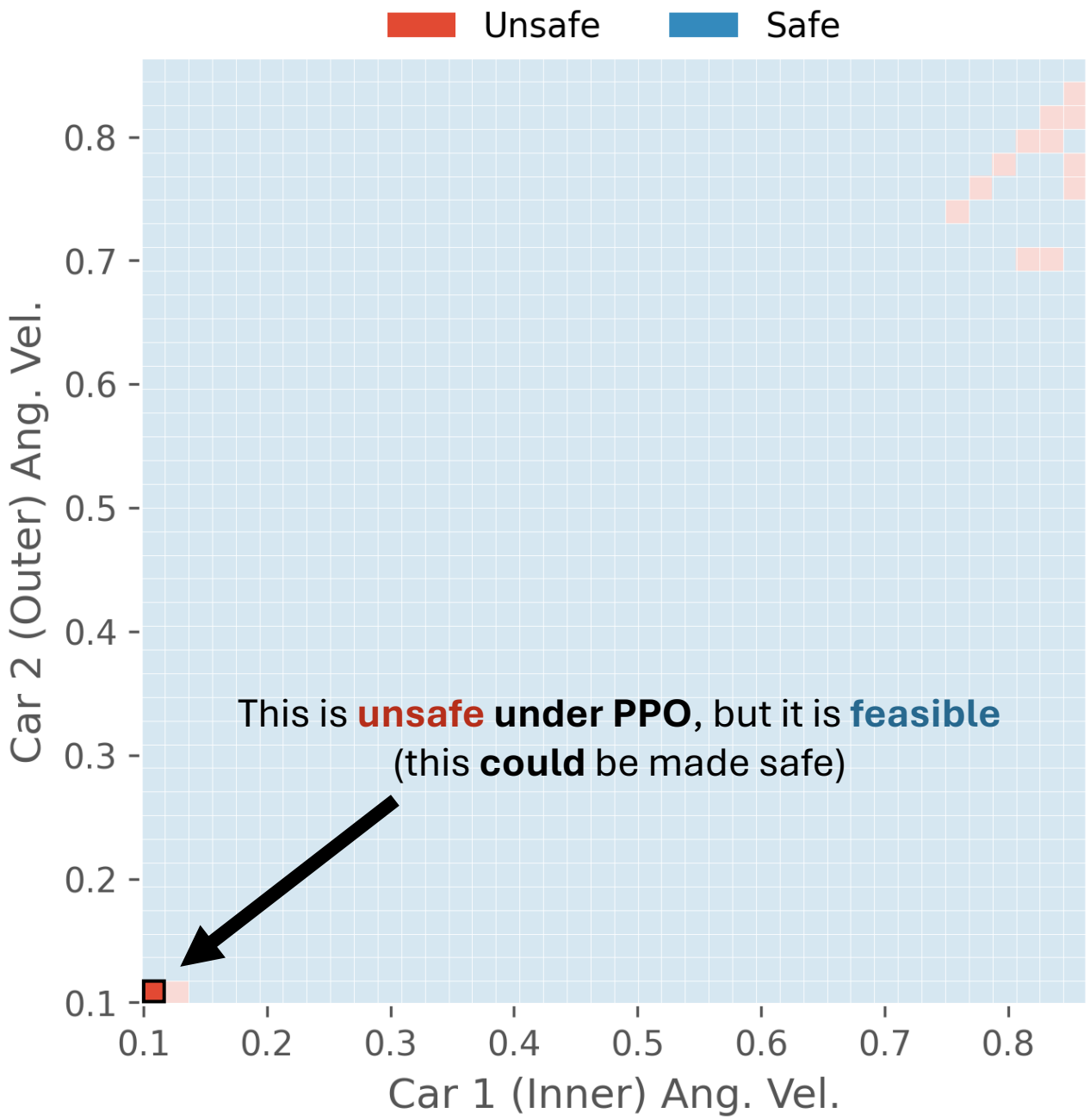


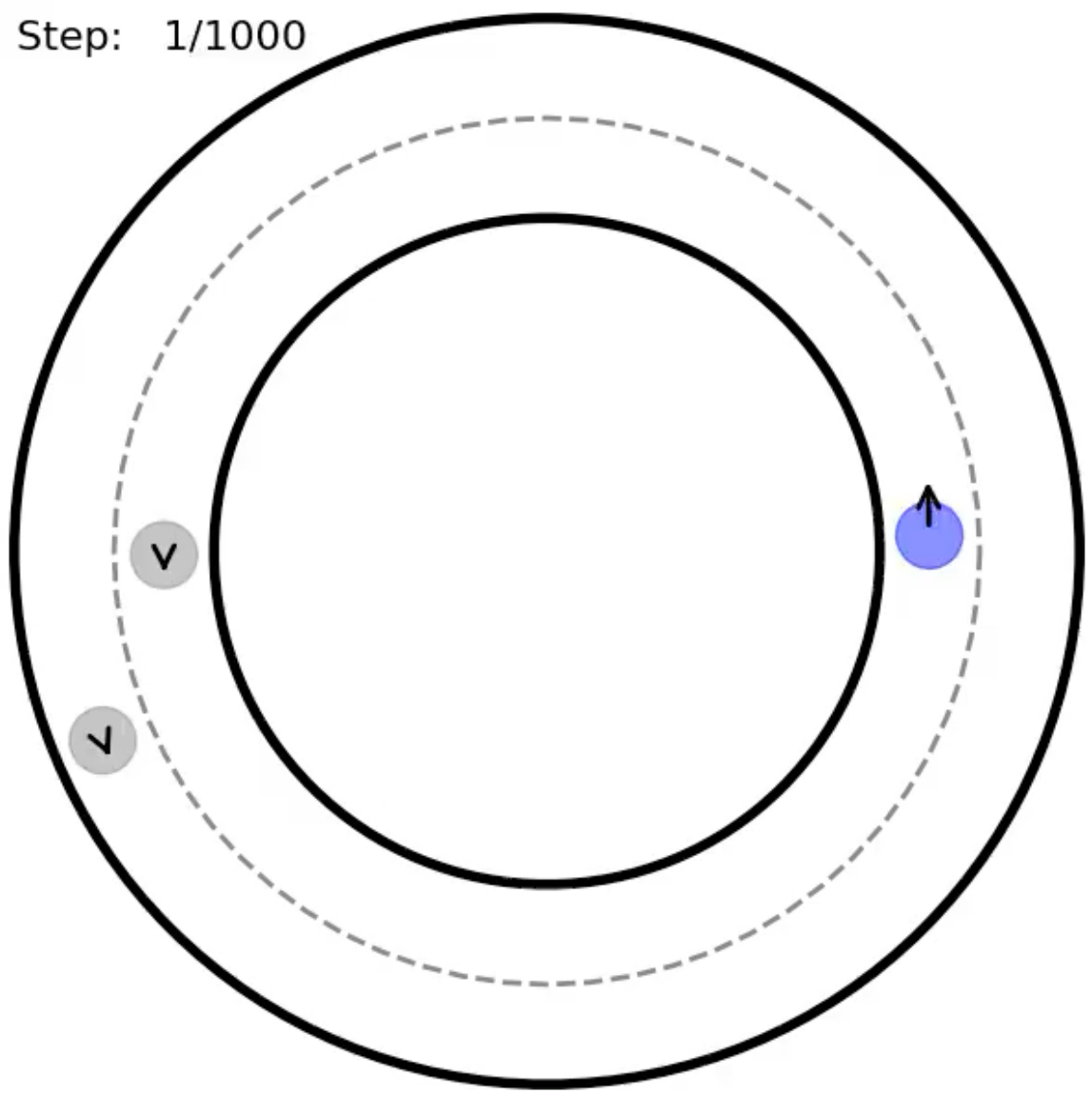

Unsafe
Safe
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
Car 2 (Outer) Ang. Vel.
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Car 1 (Inner) Ang. Vel.

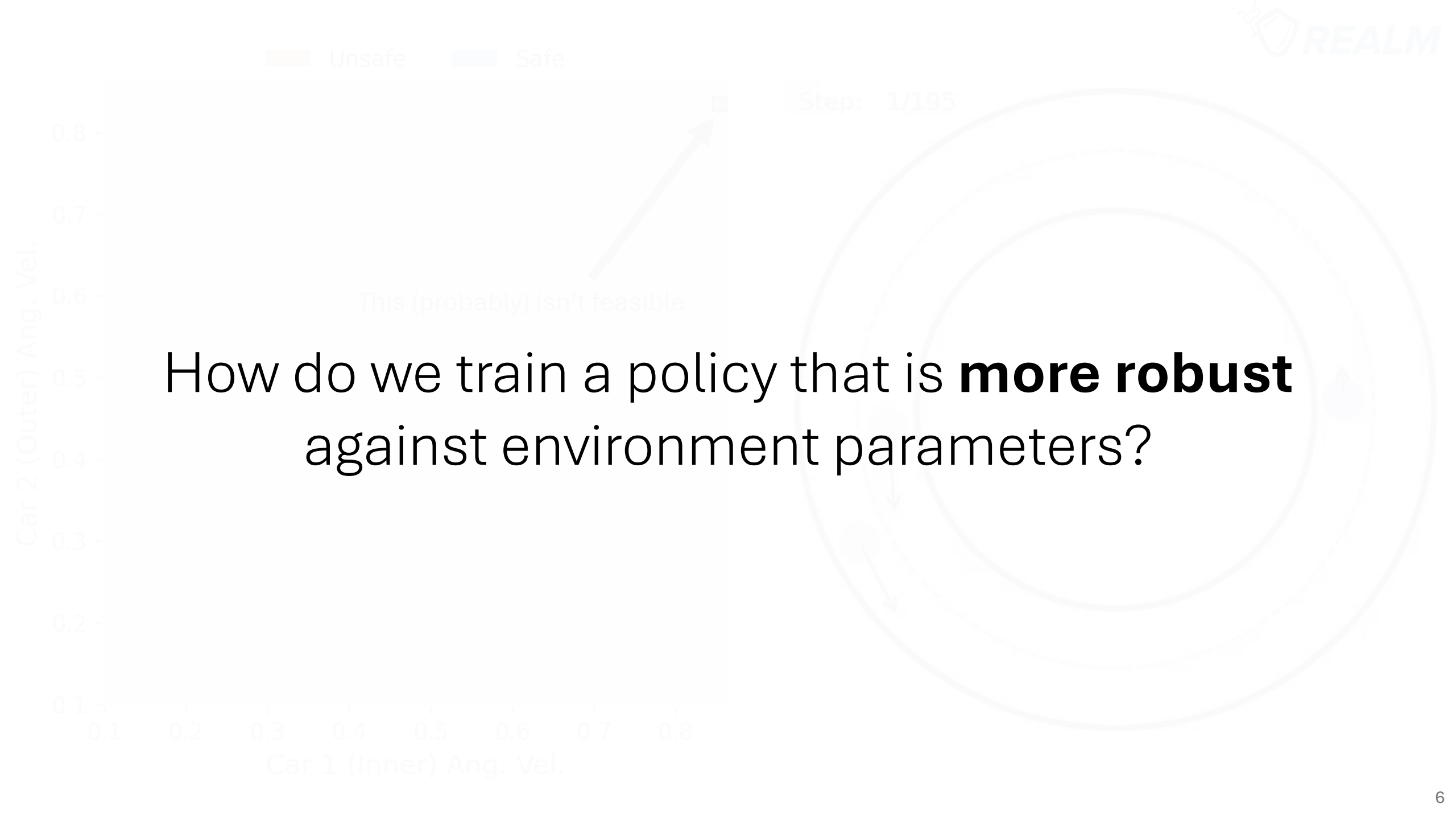

Step: 1/1000


Unsafe
Safe
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
Car 2 (Outer) Ang. Vel.
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Car 1 (Inner) Ang. Vel.


Step: 1/632

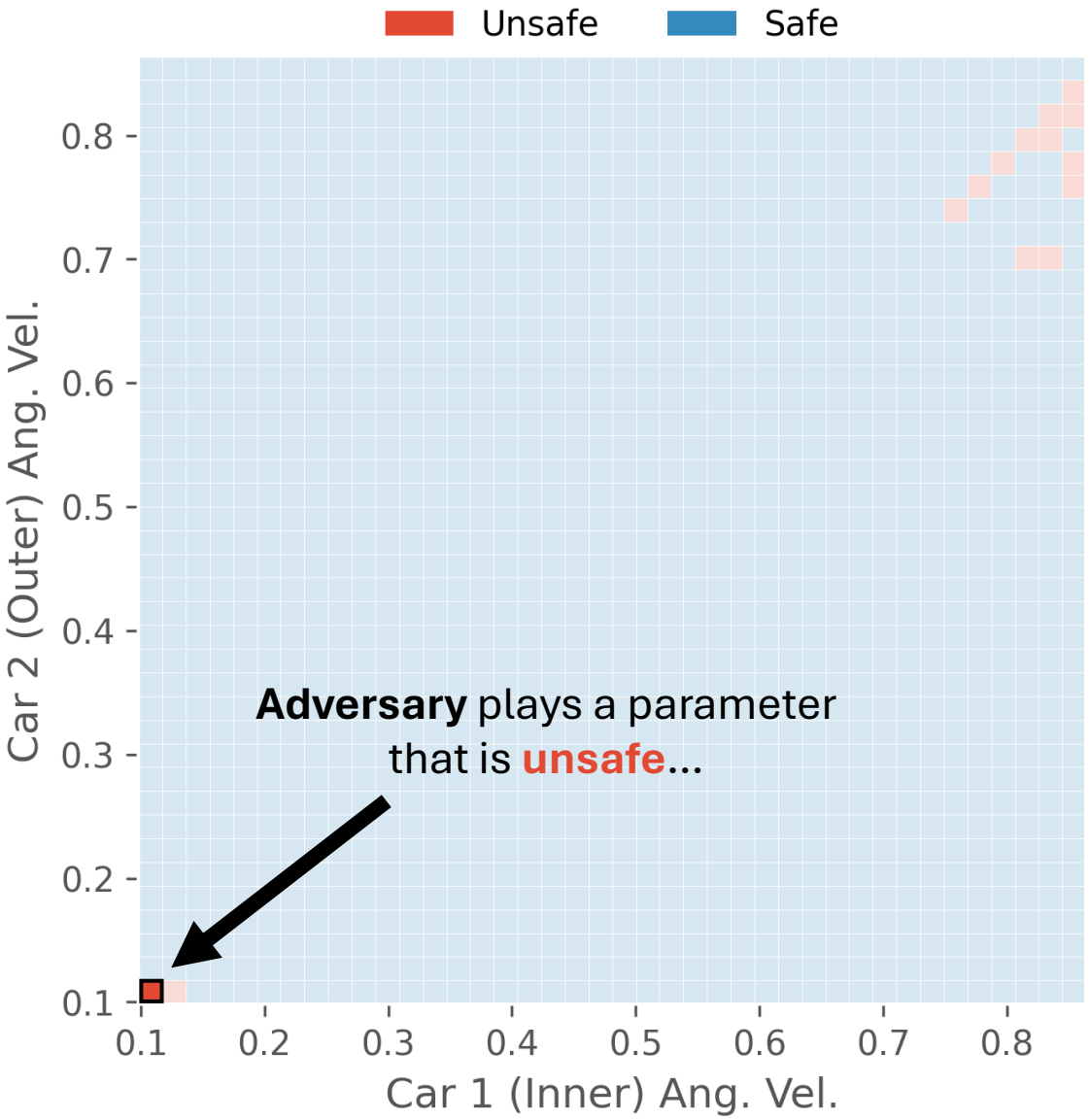

Unsafe
Safe
Car 2 (Outer) Ang. Vel.
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Car 1 (Inner) Ang. Vel.
This is unsafe under PPO, but it is feasible
(this could be made safe)


Step: 1/1000

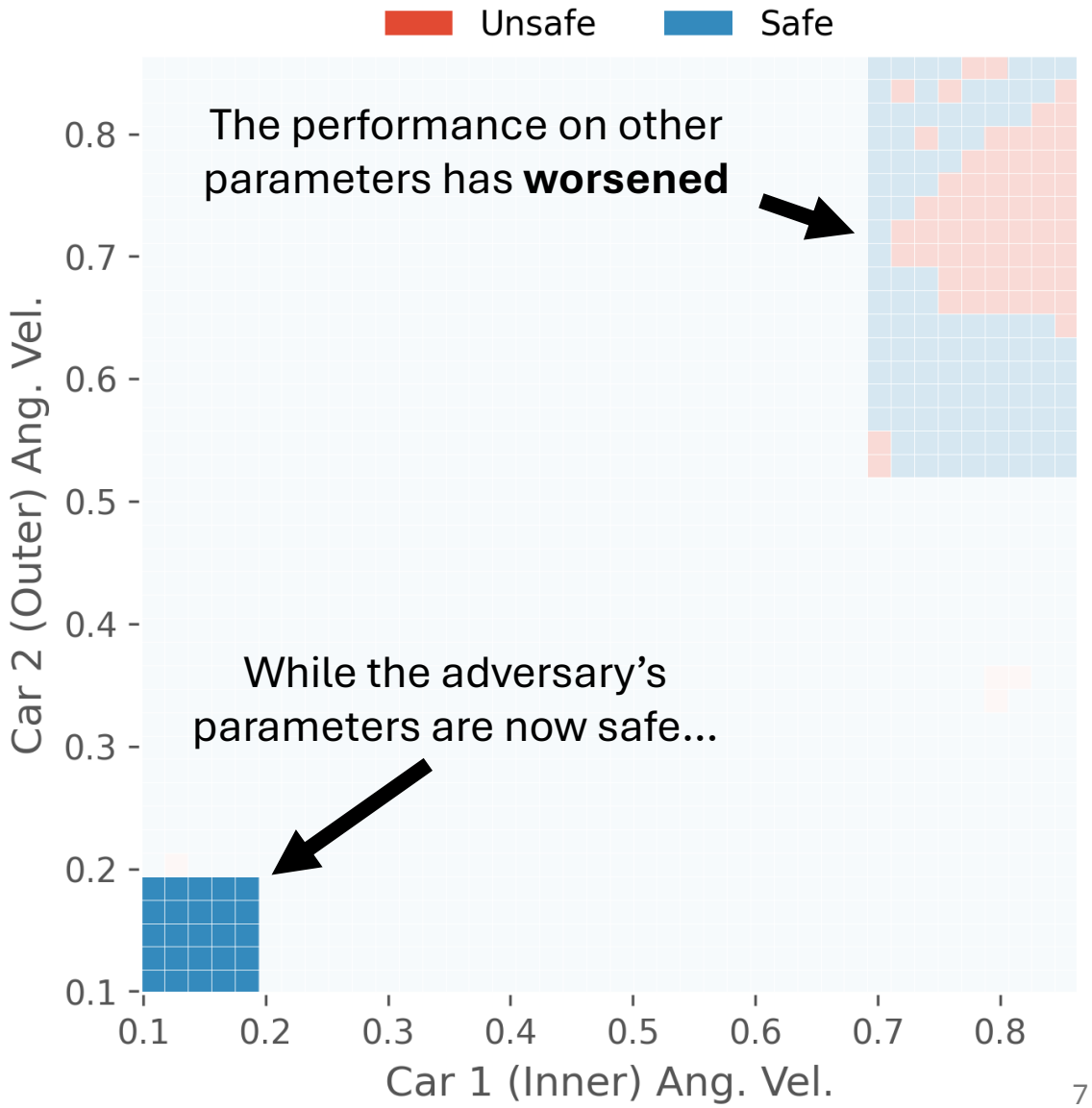


How do we train a policy that is **more robust** against environment parameters?

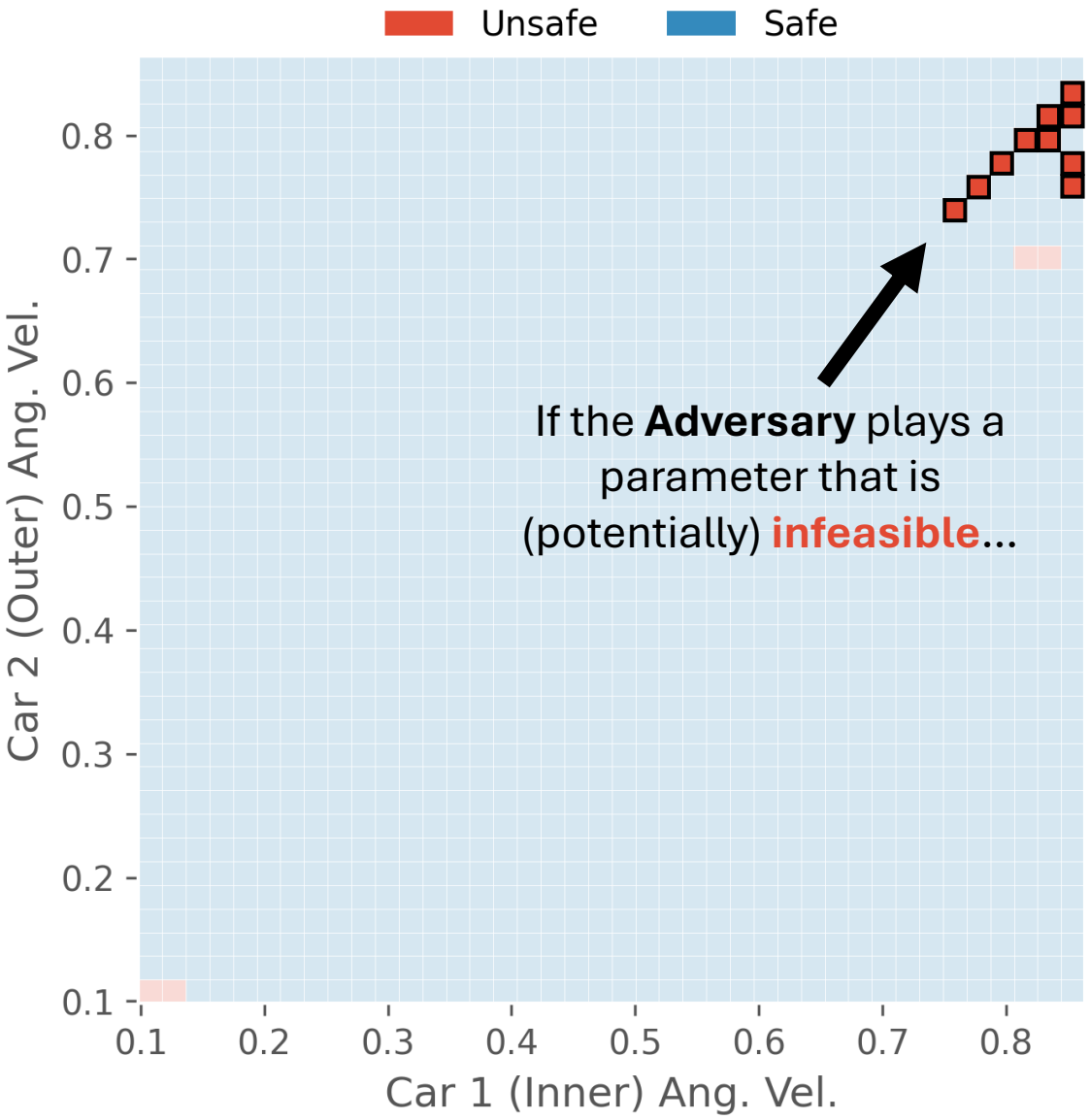
Adversarial RL helps but has flaws

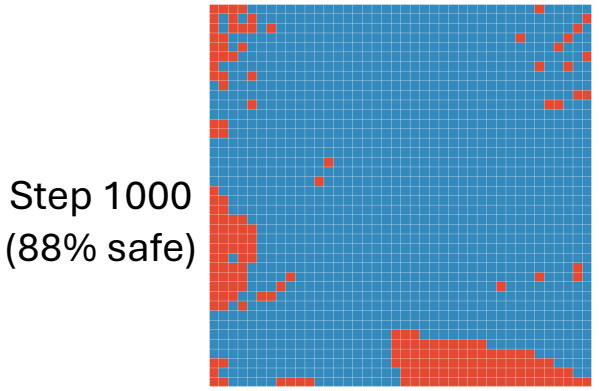


Finetune

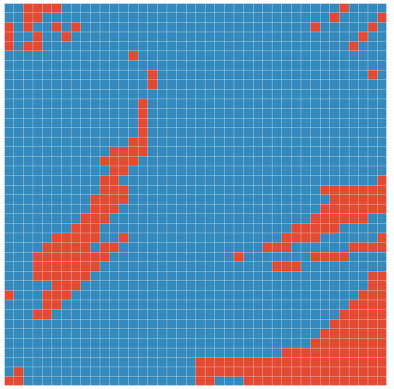


Adversarial RL helps but has flaws

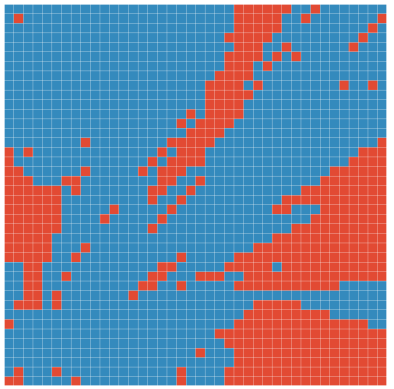




Step 1000
(88% safe)



Step 2000
(86% safe)

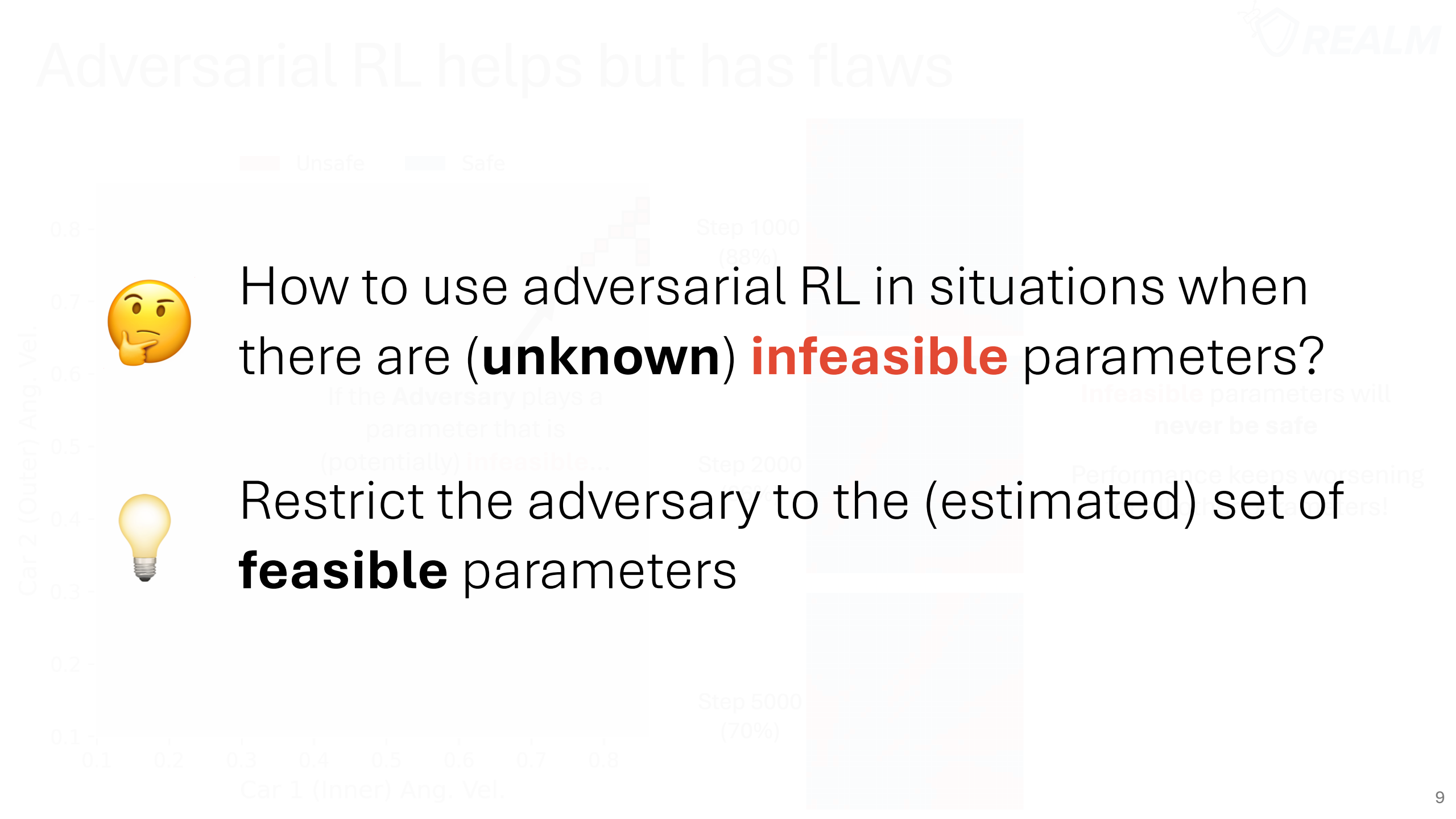


Step 5000
(70% safe)

Infeasible parameters will **never be safe**

Performance keeps worsening for all other parameters!

Adversarial RL helps but has flaws



🤔 How to use adversarial RL in situations when there are (**unknown**) **infeasible** parameters?

💡 Restrict the adversary to the (estimated) set of **feasible** parameters

Formal Problem Definition

$$\max_{\Theta' \subseteq \Theta} \quad |\Theta'|$$

Find the **largest** set of parameters...

$$\text{s.t.} \quad \exists \pi_{(.)} \in \mathcal{M}, \forall \theta \in \Theta',$$

$$\underbrace{\mathbb{1}\{\text{initial state } s_0(\theta) \text{ is safe under } \pi_\theta\} > 0}$$

... such that there exists a policy that makes every parameter safe ⊛

$$\circledast = \max_{\pi_{(.)} \in \mathcal{M}} \min_{\theta \in \Theta'} \quad \sum_{k=0}^{T_{\theta, \mathbf{s}_0}} \mathbb{1}\{\mathbf{s}_k \text{ is safe}\} > 0,$$

$T_{\theta, \mathbf{s}_0}$ ← Terminate on unsafe state

$$\text{s.t.} \quad \mathbf{s}_0 = s_0(\theta),$$

$$\mathbf{s}_{k+1} = f_\theta(\mathbf{s}_k, \pi_\theta(\mathbf{s}_k))$$

This is a **minimax** RL problem

Building up our method

1. How do we solve for a robust policy, assuming we know the feasible set (of parameters)?

2. How do we (conservatively) estimate the feasible set?

3. How do we improve our feasible set estimate?

Building up our method

1. How do we solve for a robust policy, assuming we know the feasible set (of parameters)?

   💡 Saddle-point finding using techniques from online learning

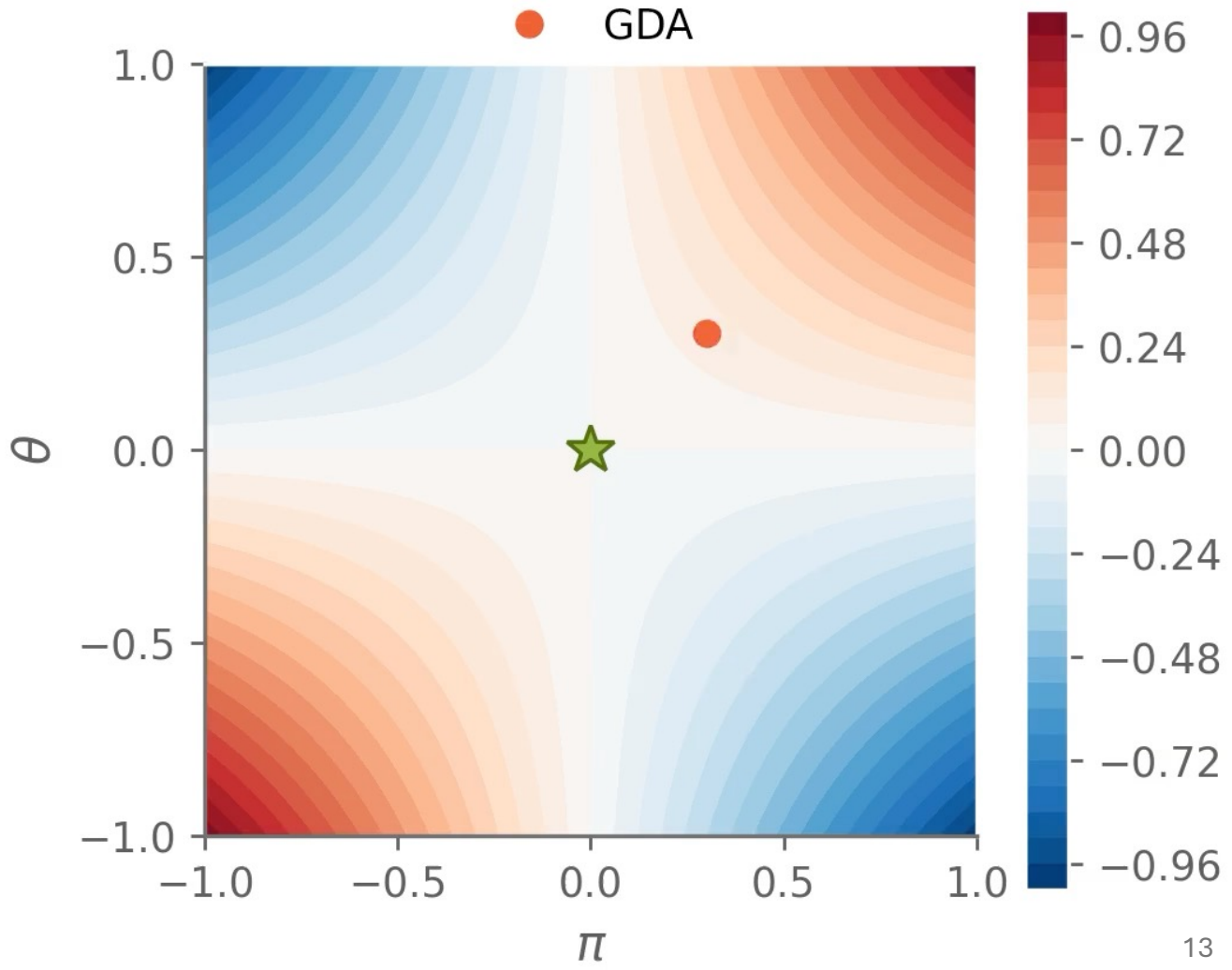
2. How do we (conservatively) estimate the feasible set?

3. How do we improve our feasible set estimate?

Saddle-point finding

$$\max_{\pi \in \mathcal{M}} \min_{\theta \in \Theta^*} J(\pi, \theta)$$

$$\max_{\pi \in [-1,1]} \min_{\theta \in [-1,1]} \pi \cdot \theta$$



Saddle-point finding

🤖 Protagonist (FTRL)

$$\pi_{t+1} := \underbrace{\arg\max_{\pi} \boxed{-\psi_t(\pi)} + \boxed{\sum_{i=1}^{t} J(\pi, \theta_i)}}_{:= \tilde{J}_t(\pi)}$$

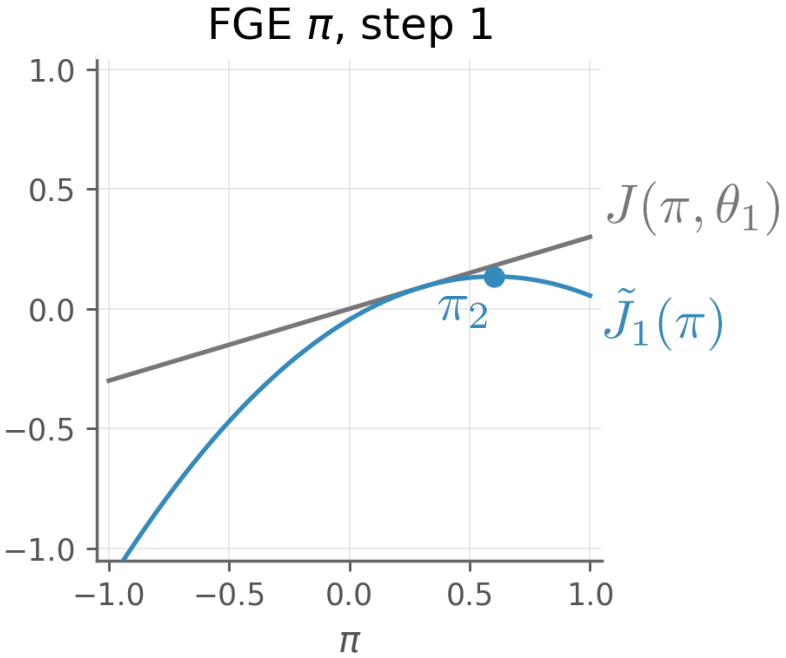
"Trust-region" regularization

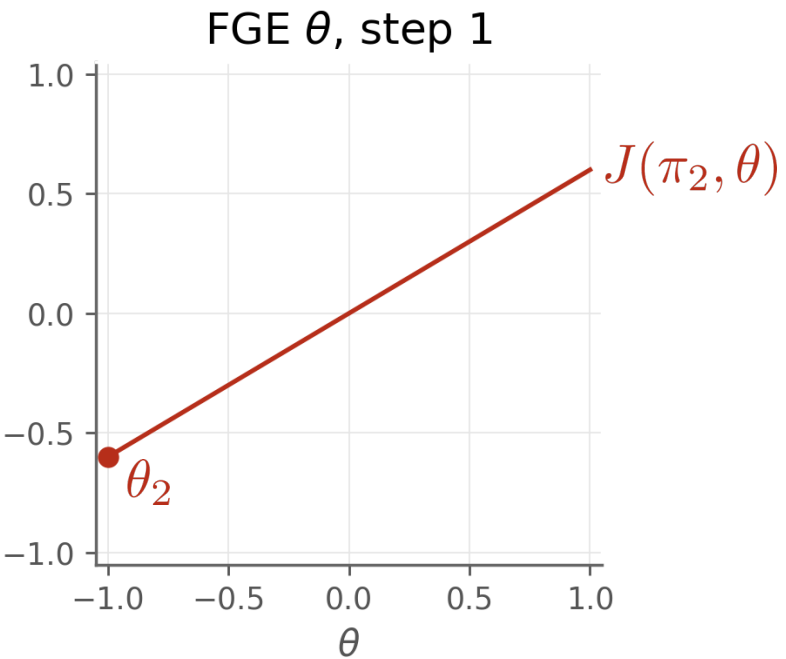
Consider **all** previous θ_i

θ_{t+1} ↶ ↷ π_{t+1}

😈 Adversary (Best-Response)

$$\theta_{t+1} := \arg\min_{\theta \in \Theta^*} J(\pi_{t+1}, \theta)$$





Saddle-point finding

🤖 Protagonist (FTRL)

$$\pi_{t+1} := \underbrace{\arg\max_{\pi} \boxed{-\psi_t(\pi)} + \sum_{i=1}^{t} J(\pi, \theta_i)}_{:= \tilde{J}_t(\pi)}$$

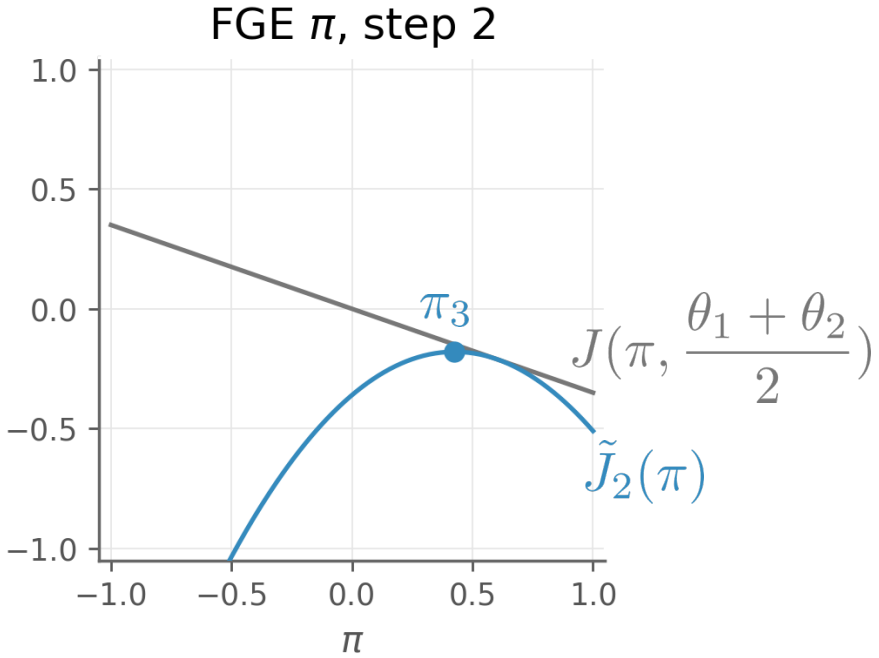
"Trust-region" regularization

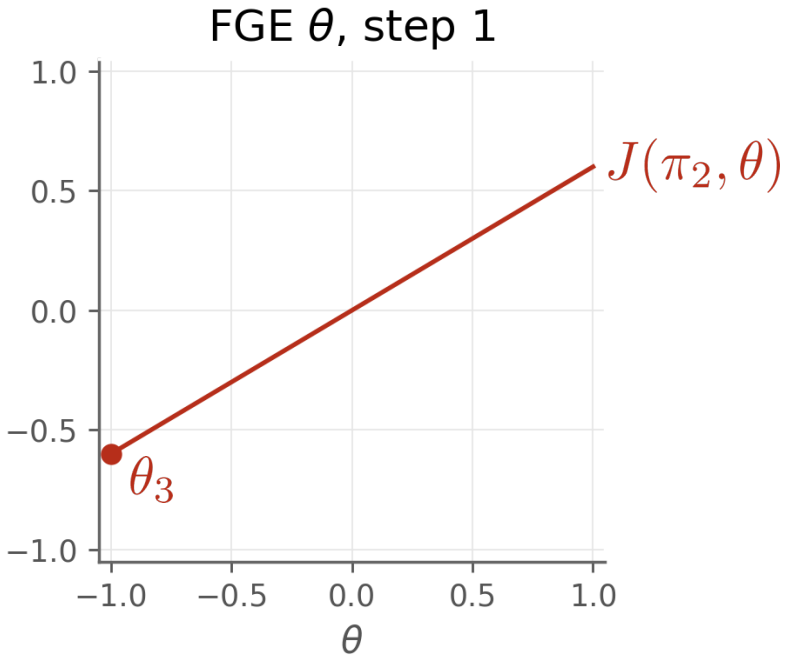
Consider **all** previous θ_i

θ_{t+1} ↶ ↷ π_{t+1}

😈 Adversary (Best-Response)

$$\theta_{t+1} := \arg\min_{\theta \in \Theta^*} J(\pi_{t+1}, \theta)$$





Saddle-point finding

🤖 Protagonist (FTRL)

$$\pi_{t+1} := \underbrace{\arg\max_{\pi} \boxed{-\psi_t(\pi)} + \sum_{i=1}^{t} J(\pi, \theta_i)}_{:= \tilde{J}_t(\pi)}$$

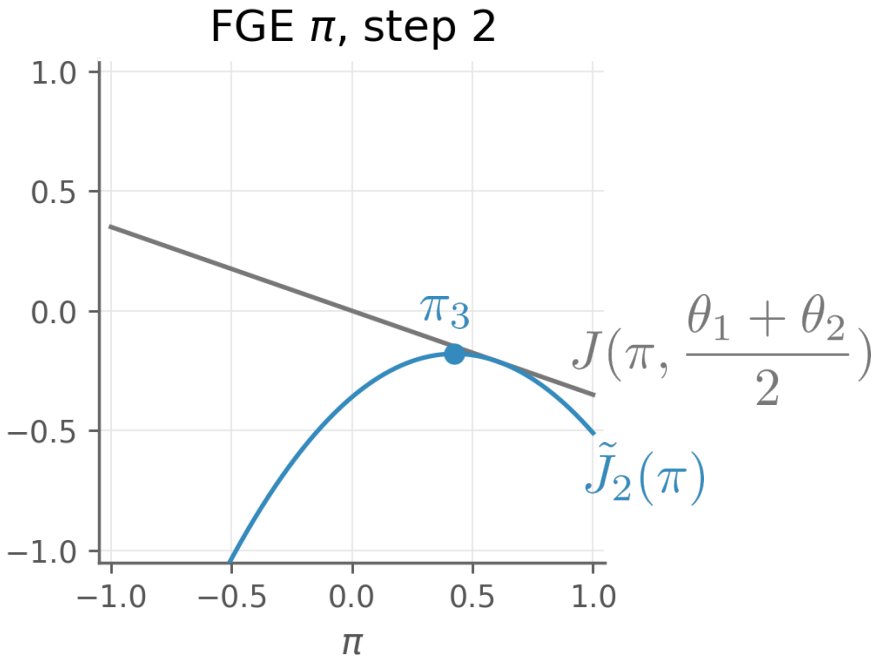
"Trust-region" regularization

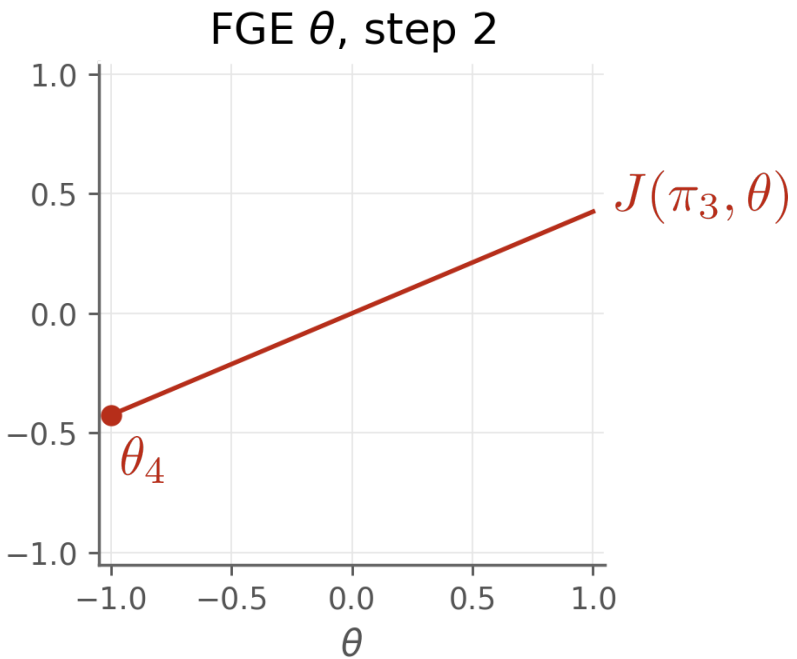
Consider **all** previous θ_i

θ_{t+1} ↶ ↷ π_{t+1}

😈 Adversary (Best-Response)

$$\theta_{t+1} := \arg\min_{\theta \in \Theta^*} J(\pi_{t+1}, \theta)$$





Saddle-point finding

🤖 Protagonist (FTRL)

$$\pi_{t+1} := \underbrace{\arg\max_{\pi} \boxed{-\psi_t(\pi)} + \boxed{\sum_{i=1}^{t} J(\pi, \theta_i)}}_{:= \tilde{J}_t(\pi)}$$

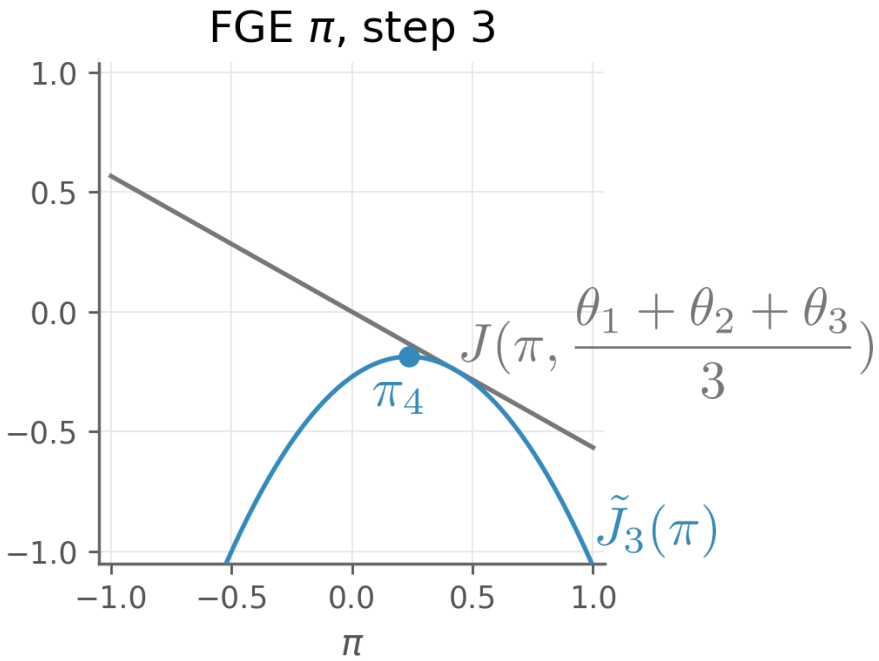
"Trust-region" regularization

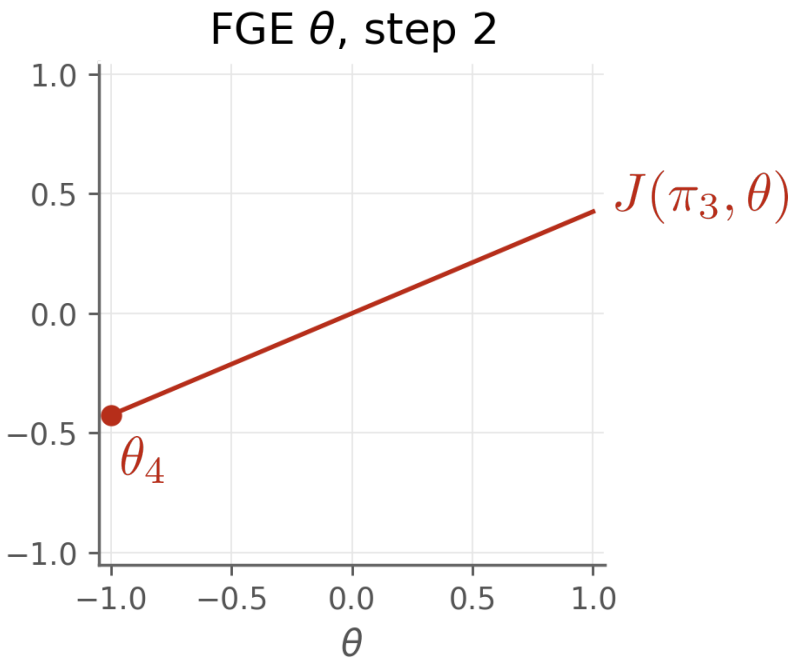
Consider **all** previous θ_i

θ_{t+1} ↶ ↷ π_{t+1}

😈 Adversary (Best-Response)

$$\theta_{t+1} := \arg\min_{\theta \in \Theta^*} J(\pi_{t+1}, \theta)$$





Saddle-point finding

🤖 Protagonist (FTRL)

$$\pi_{t+1} := \underbrace{\arg\max_{\pi} \boxed{-\psi_t(\pi)} + \boxed{\sum_{i=1}^{t} J(\pi, \theta_i)}}_{:= \tilde{J}_t(\pi)}$$

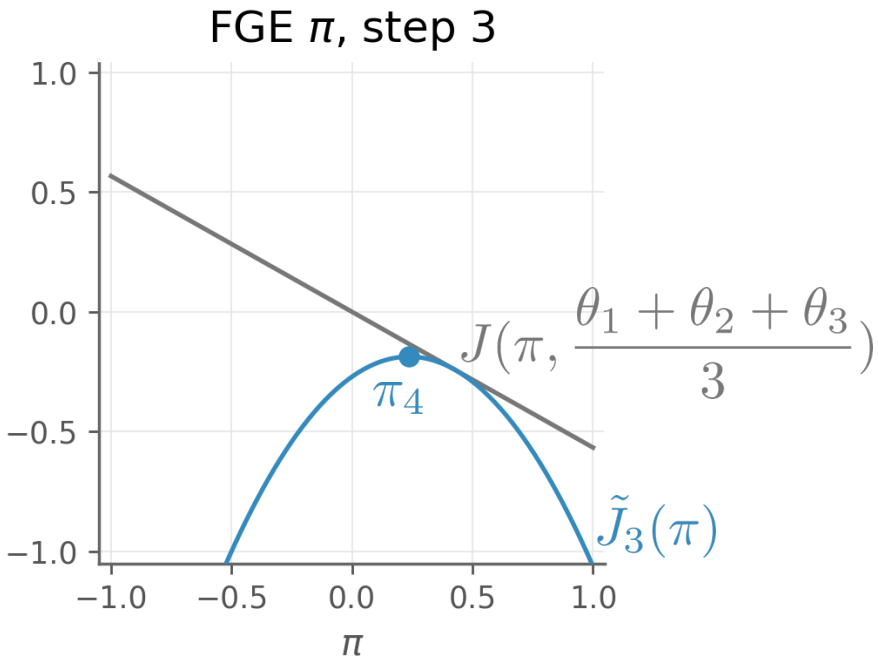
"Trust-region" regularization

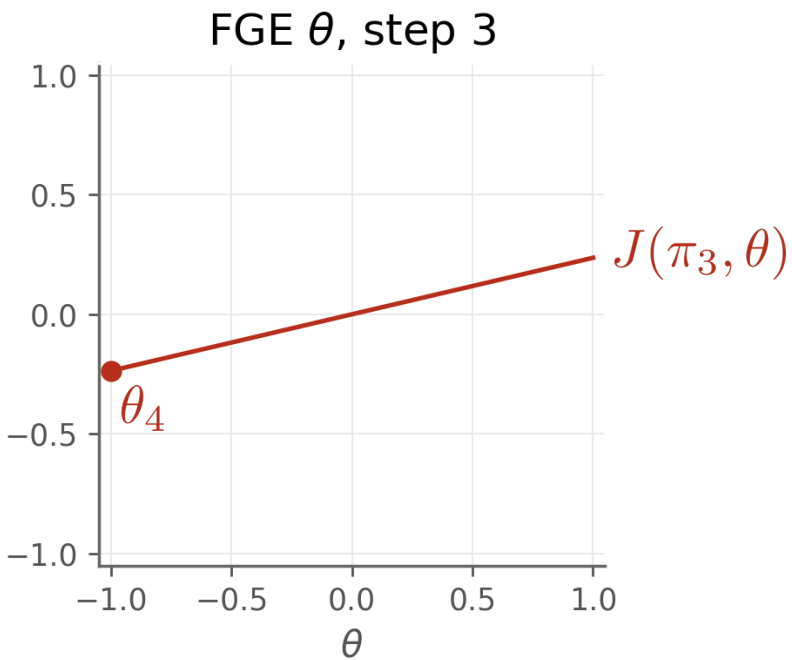
Consider **all** previous θ_i

θ_{t+1} ↶ ↷ π_{t+1}

😈 Adversary (Best-Response)

$$\theta_{t+1} := \arg\min_{\theta \in \Theta^*} J(\pi_{t+1}, \theta)$$





Saddle-point finding

🤖 Protagonist (FTRL)

$$\pi_{t+1} := \underbrace{\arg\max_{\pi} \boxed{-\psi_t(\pi)} + \sum_{i=1}^{t} J(\pi, \theta_i)}_{:= \tilde{J}_t(\pi)}$$

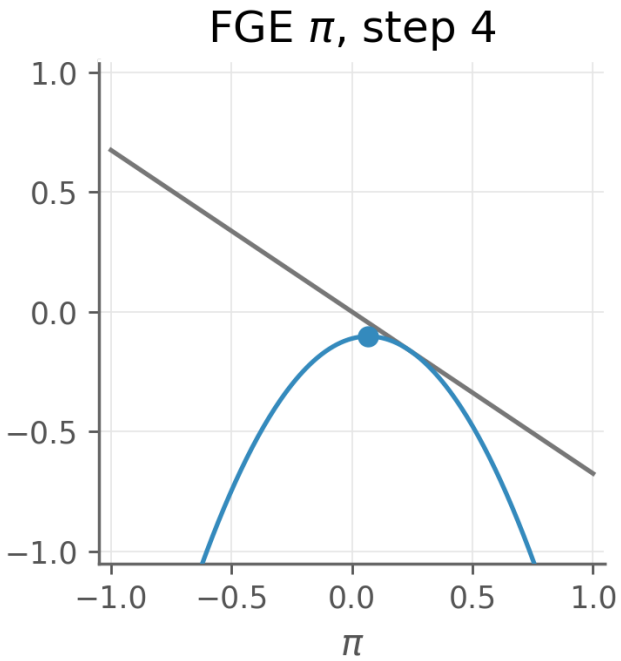
"Trust-region" regularization

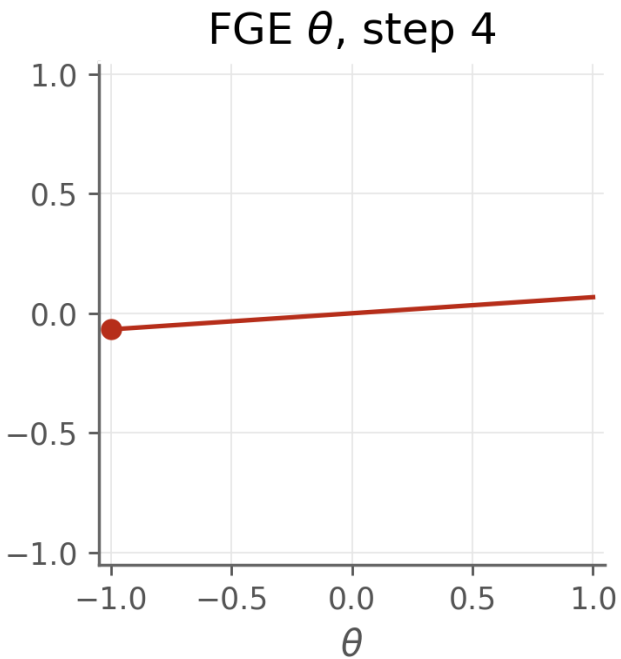
Consider **all** previous θ_i

θ_{t+1} ↶ ↷ π_{t+1}

😈 Adversary (Best-Response)

$$\theta_{t+1} := \arg\min_{\theta \in \Theta^*} J(\pi_{t+1}, \theta)$$





Saddle-point finding

🤖 Protagonist (FTRL)

$$\pi_{t+1} := \underbrace{\arg\max_{\pi} \boxed{-\psi_t(\pi)} + \sum_{i=1}^{t} J(\pi, \theta_i)}_{:=\tilde{J}_t(\pi)}$$

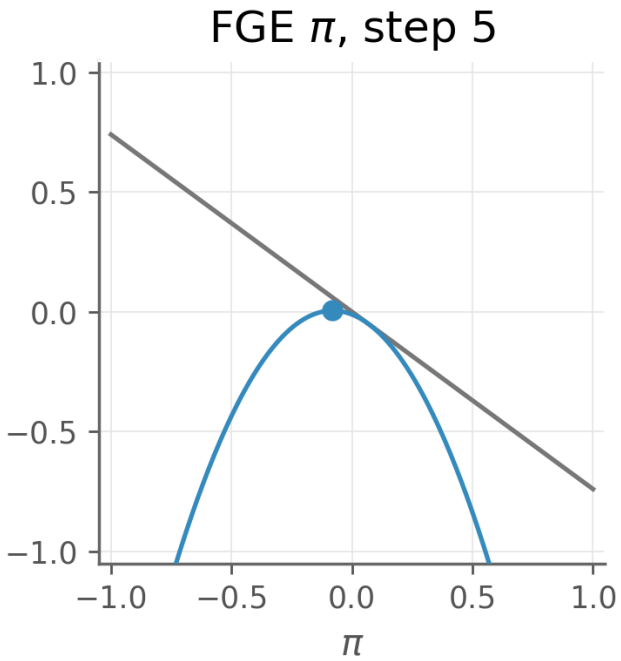
"Trust-region" regularization

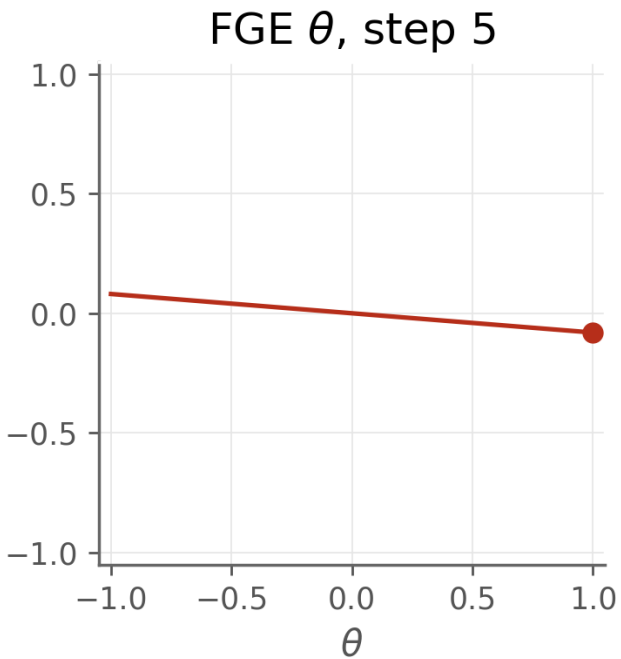
Consider **all** previous θ_i

θ_{t+1} ↶ ↷ π_{t+1}

😈 Adversary (Best-Response)

$$\theta_{t+1} := \arg\min_{\theta \in \Theta^*} J(\pi_{t+1}, \theta)$$





Saddle-point finding

🤖 Protagonist (FTRL)

$$\pi_{t+1} := \underbrace{\arg\max_{\pi} \boxed{-\psi_t(\pi)} + \boxed{\sum_{i=1}^{t} J(\pi, \theta_i)}}_{:= \tilde{J}_t(\pi)}$$

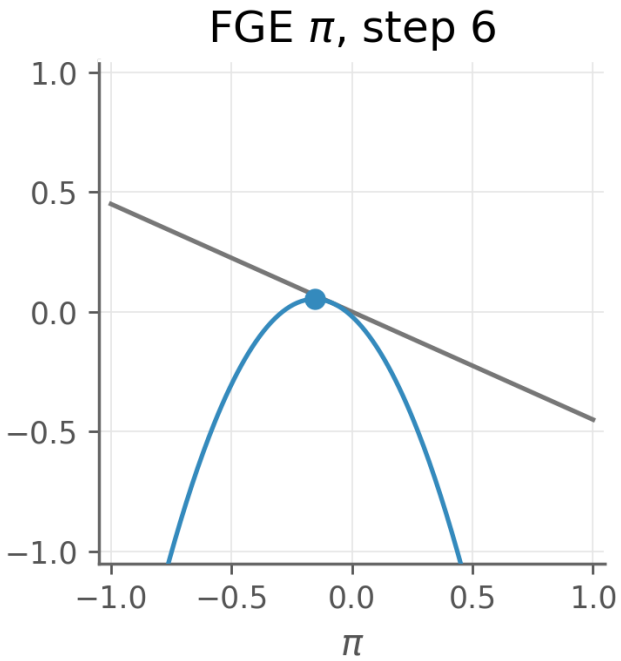
"Trust-region" regularization

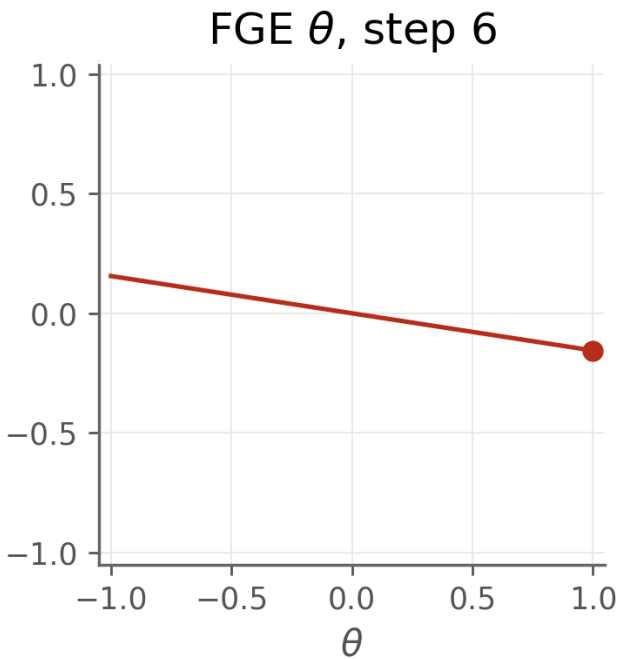
Consider **all** previous θ_i

θ_{t+1} ↶ ↷ π_{t+1}

😈 Adversary (Best-Response)

$$\theta_{t+1} := \arg\min_{\theta \in \Theta^*} J(\pi_{t+1}, \theta)$$

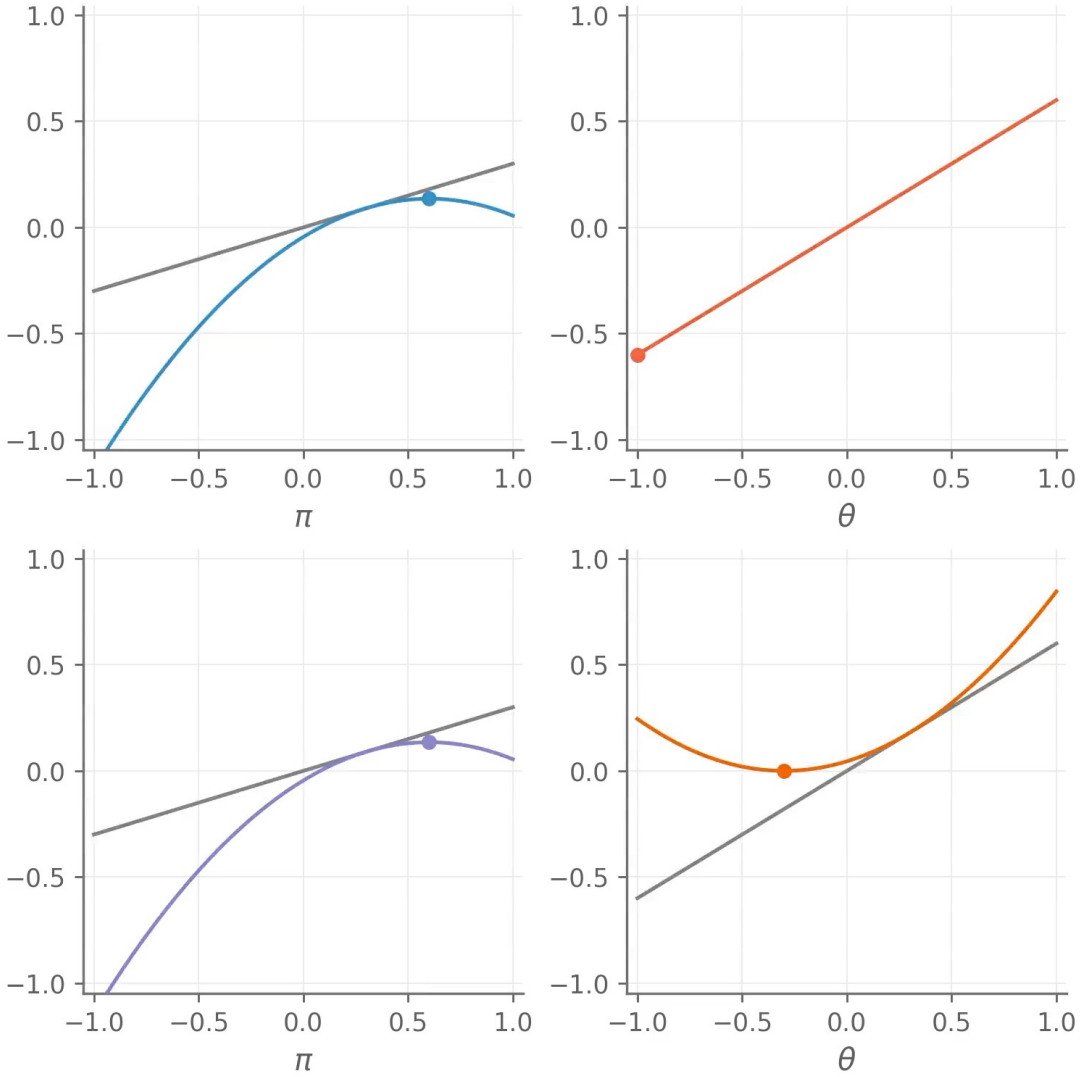


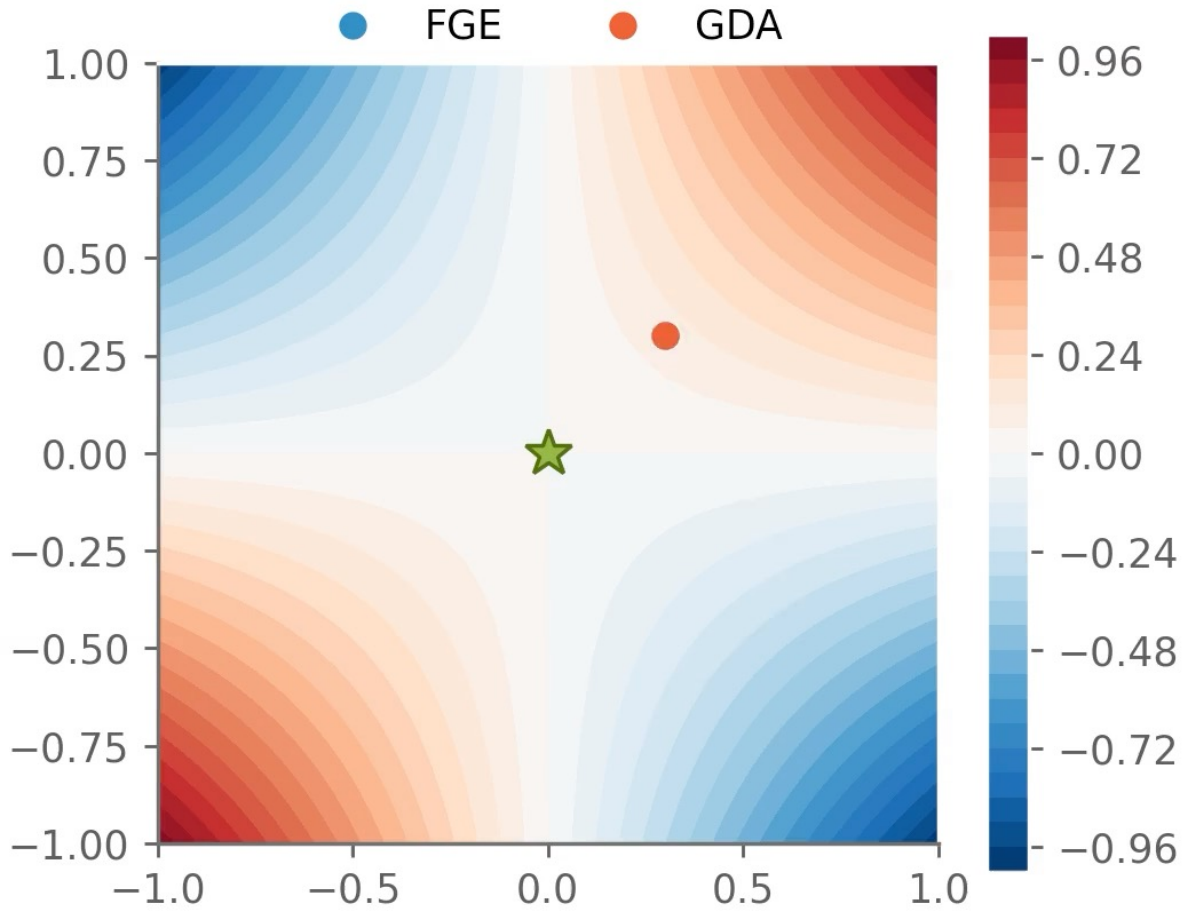


Saddle-point finding

FGE (Ours)

GDA





Saddle-point finding | Implementation

REALM

🤖 Protagonist (FTRL)

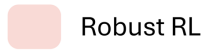
$$\pi_{t+1} := \underbrace{\arg\max_{\pi} -\psi_t(\pi) + \sum_{i=1}^{t} J(\pi, \theta_i)}_{:=\tilde{J}_t(\pi)}$$

$$\pi_{t+1} = \pi_t + \eta_t \nabla_\pi \frac{1}{t} \sum_{i=1}^{t} J(\pi, \theta_i)$$

This is just **gradient ascent**!
→ PPO update with **domain randomization** over θ

$$= \pi_t + \eta_t \nabla_\pi \mathbb{E}_{\theta \sim \mathcal{D}_{\theta,t}} \left[J(\pi, \theta) \right], \quad \mathcal{D}_{\theta,t} = \mathrm{Uniform}(\{\theta_i\}_{i=1}^{t})$$

Saddle-point finding | Implementation



Building up our method

1. How do we solve for a robust policy, assuming we know the feasible set (of parameters)?

   Saddle-point finding using techniques from online learning

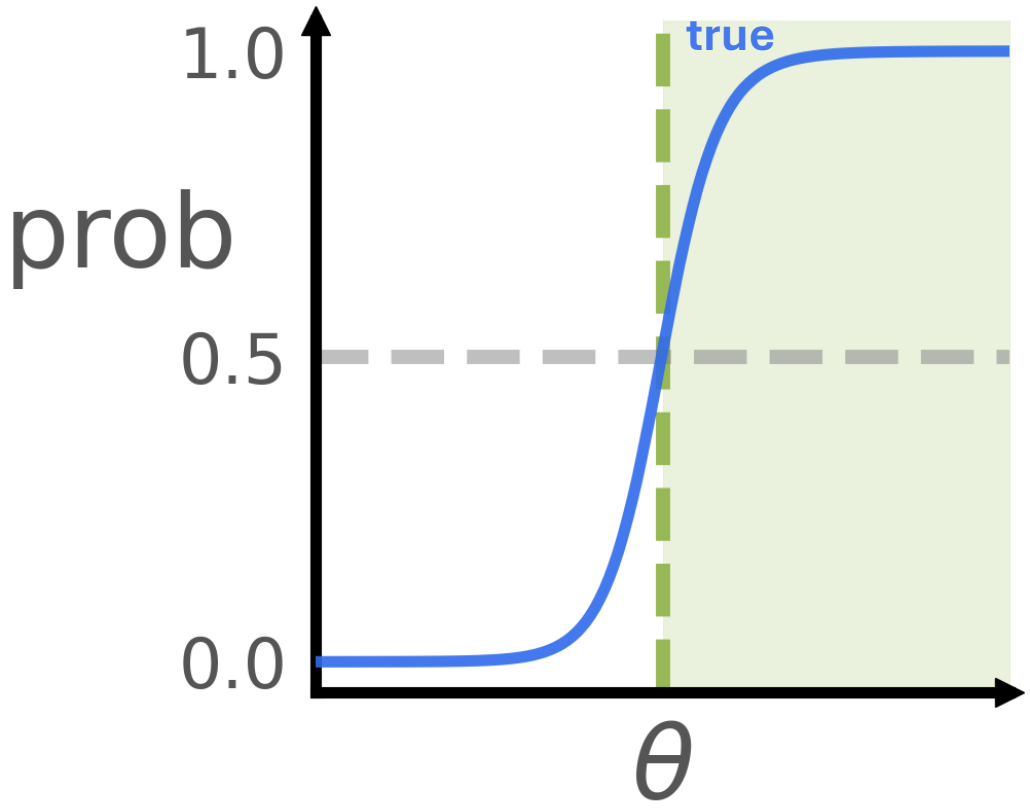
2. How do we (conservatively) estimate the feasible set?

   Use policy rollouts in a smart way

3. How do we improve our feasible set estimate?

Feasible Set Estimation

Want to learn $p(\textbf{safe} \mid \theta)$.

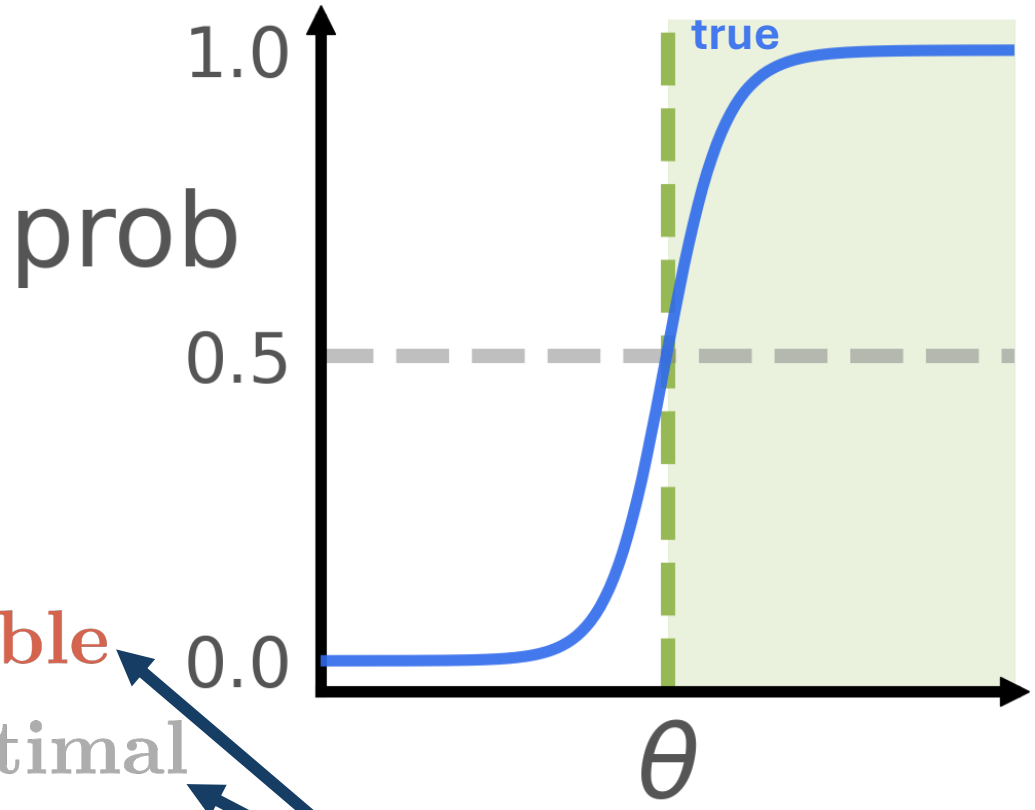


Feasible Set Estimation

Want to learn $p(\textbf{safe} \mid \theta)$.

π renders θ **safe** $\implies$ θ is **feasible**

unsafe $\implies$ $\begin{cases} \theta \text{ is } \textbf{infeasible} \\ \pi \text{ is } \textbf{suboptimal} \end{cases}$



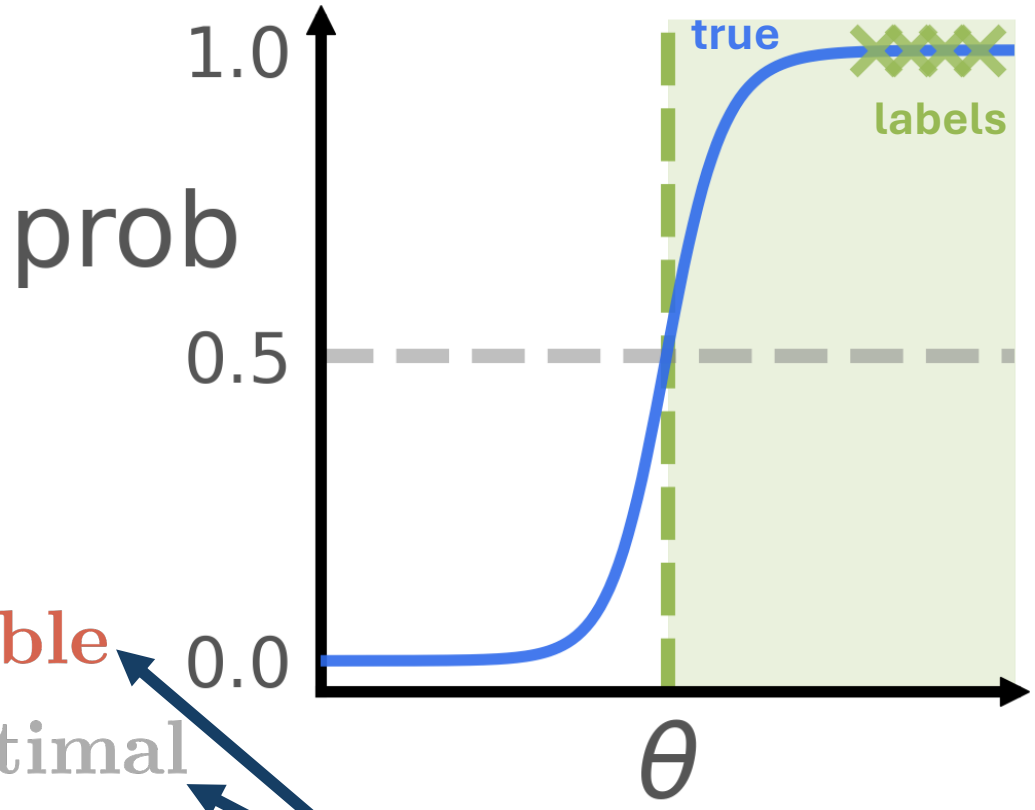
Impossible to distinguish between these two cases!

Feasible Set Estimation

Want to learn $p(\text{safe} \mid \theta)$.

π renders θ safe $\implies$ θ is feasible

unsafe $\implies$ $\begin{cases} \theta \text{ is infeasible} \\ \pi \text{ is suboptimal} \end{cases}$



Impossible to distinguish between these two cases!

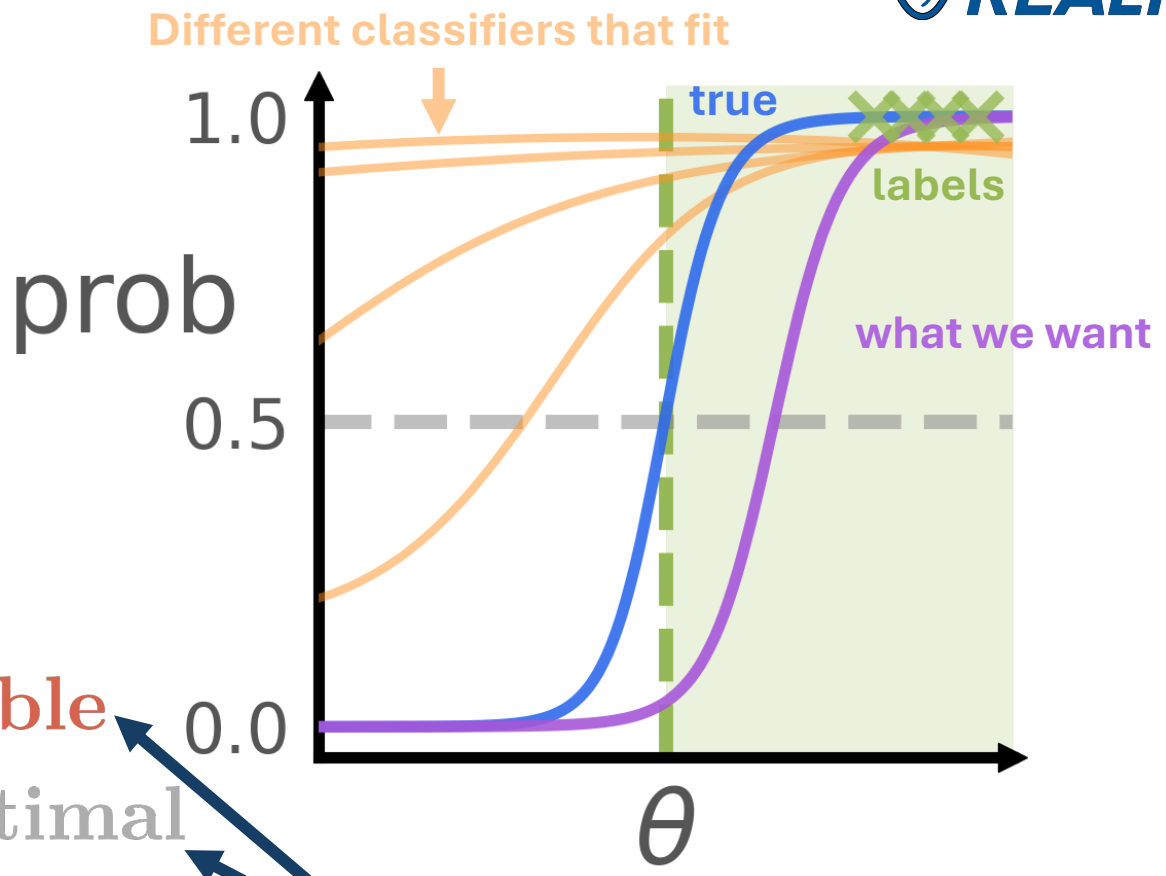
We have (θ, safe), but no (θ, unsafe).

Feasible Set Estimation

Want to learn $p(\text{safe} \mid \theta)$.

π renders θ safe $\implies$ θ is feasible

unsafe $\implies$ $\begin{cases} \theta \text{ is infeasible} \\ \pi \text{ is suboptimal} \end{cases}$



Impossible to distinguish between these two cases!

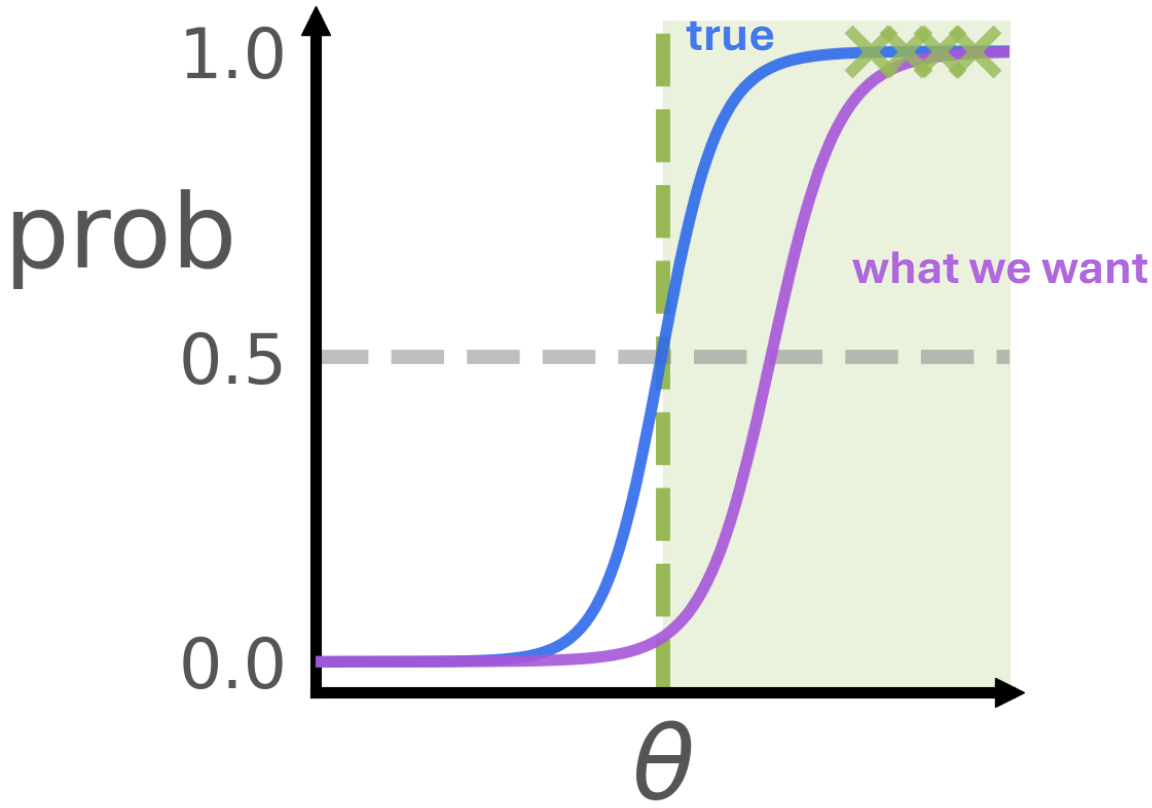
We have (θ, safe), but no (θ, unsafe).

😵‍💫 Cannot learn a classifier with only positive labels!

Feasible Set Estimation

We have $(\theta, \textbf{safe})$, but no $(\theta, \textbf{unsafe})$.

- 😵‍💫 Cannot learn a classifier with only positive labels!



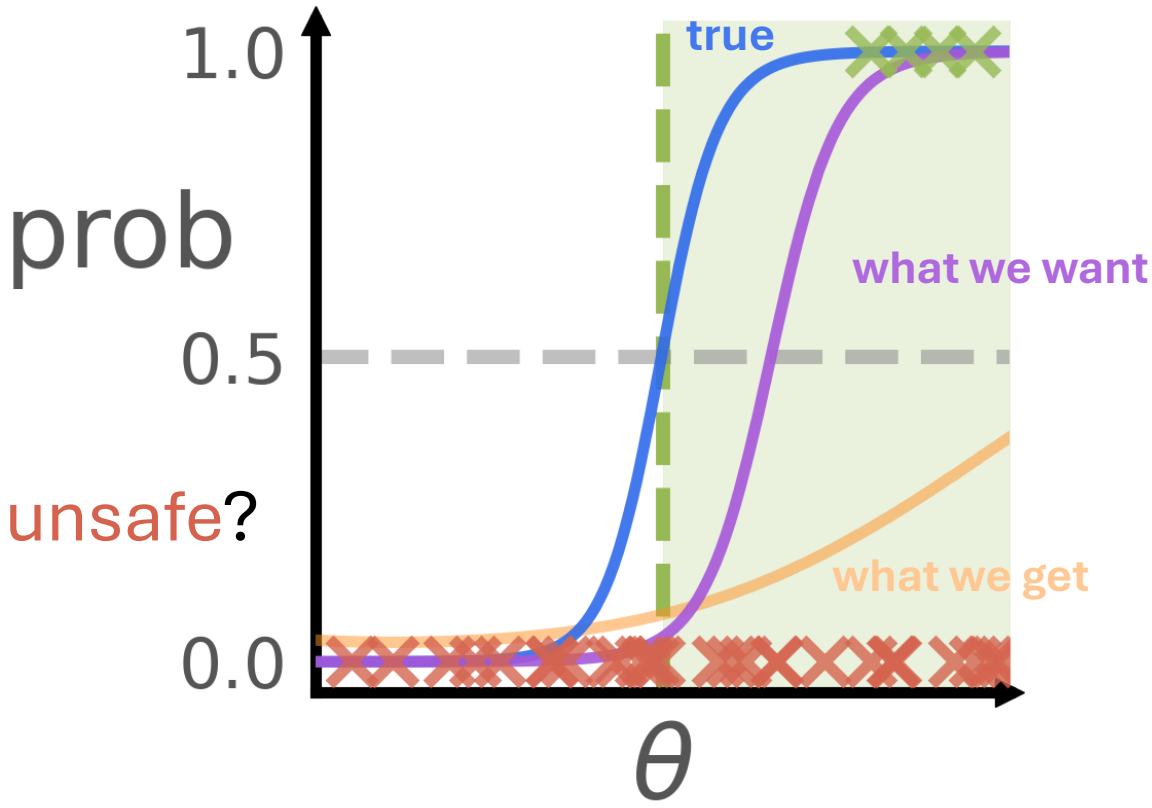
Feasible Set Estimation

We have $(\theta, \textbf{safe})$, but no $(\theta, \textbf{unsafe})$.

- 😵‍💫 Cannot learn a classifier with only positive labels!

🤔 What if we use $(\theta, \textbf{unsafe})$ when π is unsafe?

- 😵‍💫 Poor fit if π degrades!



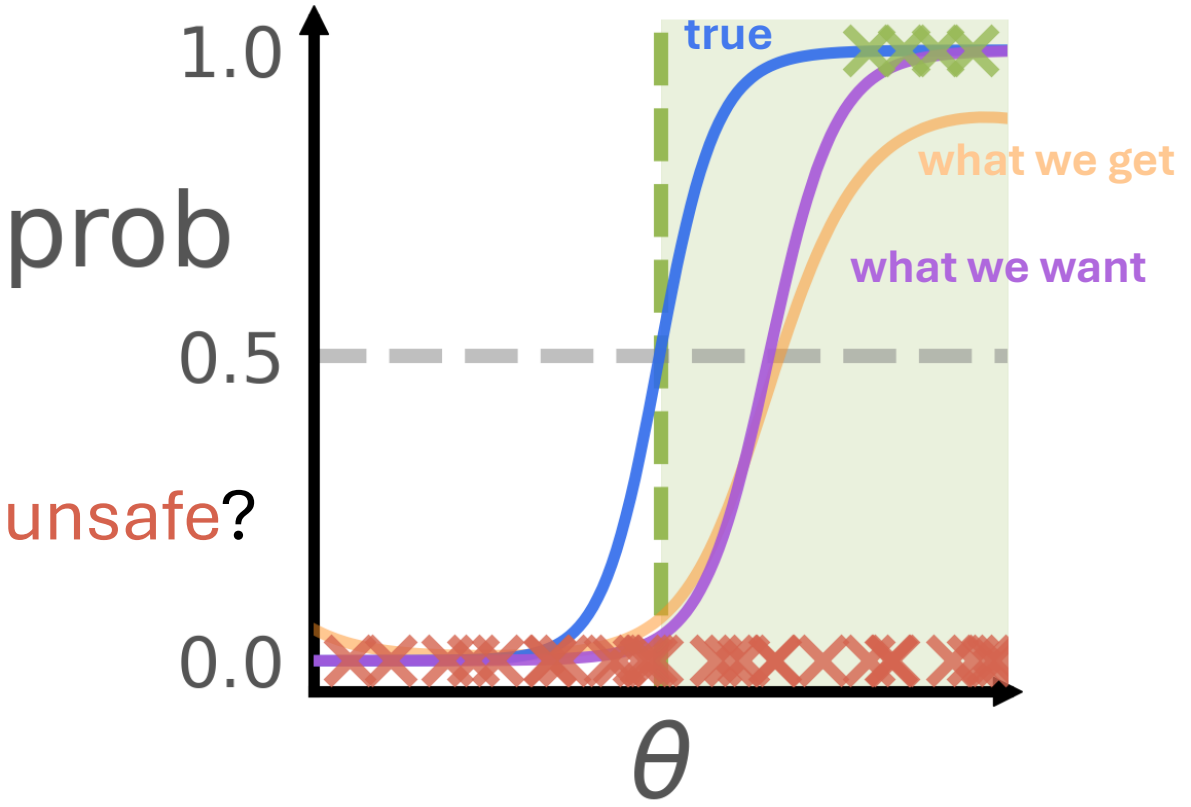
Feasible Set Estimation

We have $(\theta, \textbf{safe})$, but no $(\theta, \textbf{unsafe})$.

😵‍💫 Cannot learn a classifier with only positive labels!

🤔 What if we use $(\theta, \textbf{unsafe})$ when π is unsafe?

😵‍💫 Poor fit if π degrades!



💡 **Solution:** *Weight* the safe samples *more* than the unsafe samples!

$$P_{\text{mix}}(\textbf{safe}, \theta) = \alpha \underbrace{p^*(\textbf{safe} \mid \theta) p_{\mathcal{D}_{\textbf{feasible}}}(\theta)}_{\text{observed } \textbf{feasible } \theta} + (1-\alpha) \underbrace{p^{\pi}(\textbf{safe} \mid \theta)\rho(\theta)}_{\text{mixed } \textbf{safe}/\textbf{unsafe } \theta}$$

α lets us control the false negative rate!

Feasible Set Estimation | Guarantees

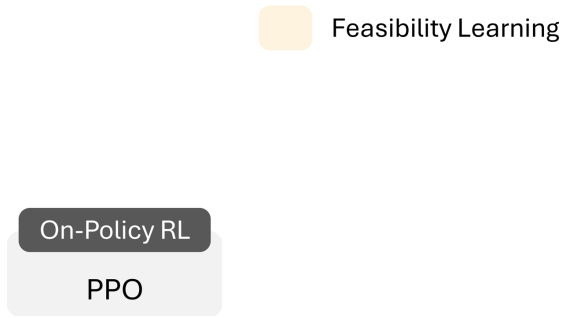
Theorem 1. Let $q_{\psi^*} : \Theta \to [0, 1]$ be the optimal variational approximation to P_{mix} wrt. variational parameters ψ.

For a **feasible** $\theta \in \mathcal{D}_{\text{feasible}}$, q_{ψ^*} classifies θ as **safe** with probability β where

$$p^{\pi}(\text{safe} \mid \theta) \geq \beta - (1 - \beta)\frac{\alpha}{1 - \alpha}\frac{p_{\mathcal{D}_{\text{feasible}}}(\theta)}{\rho(\theta)}$$

For an **infeasible** $\theta \in \Theta^{*\complement}$, q_{ψ^*} always correctly classifies θ as **unsafe**.

Feasible Set Estimation | Implementation



Building up our method

1. How do we solve for a robust policy, assuming we know the feasible set (of parameters)?

   Saddle-point finding using techniques from online learning

2. How do we (conservatively) estimate the feasible set?

   Use policy rollouts in a smart way

3. How do we improve our feasible set estimate?

   Focus exploration on parameters currently not feasible

Expanding the Feasible Set Estimation

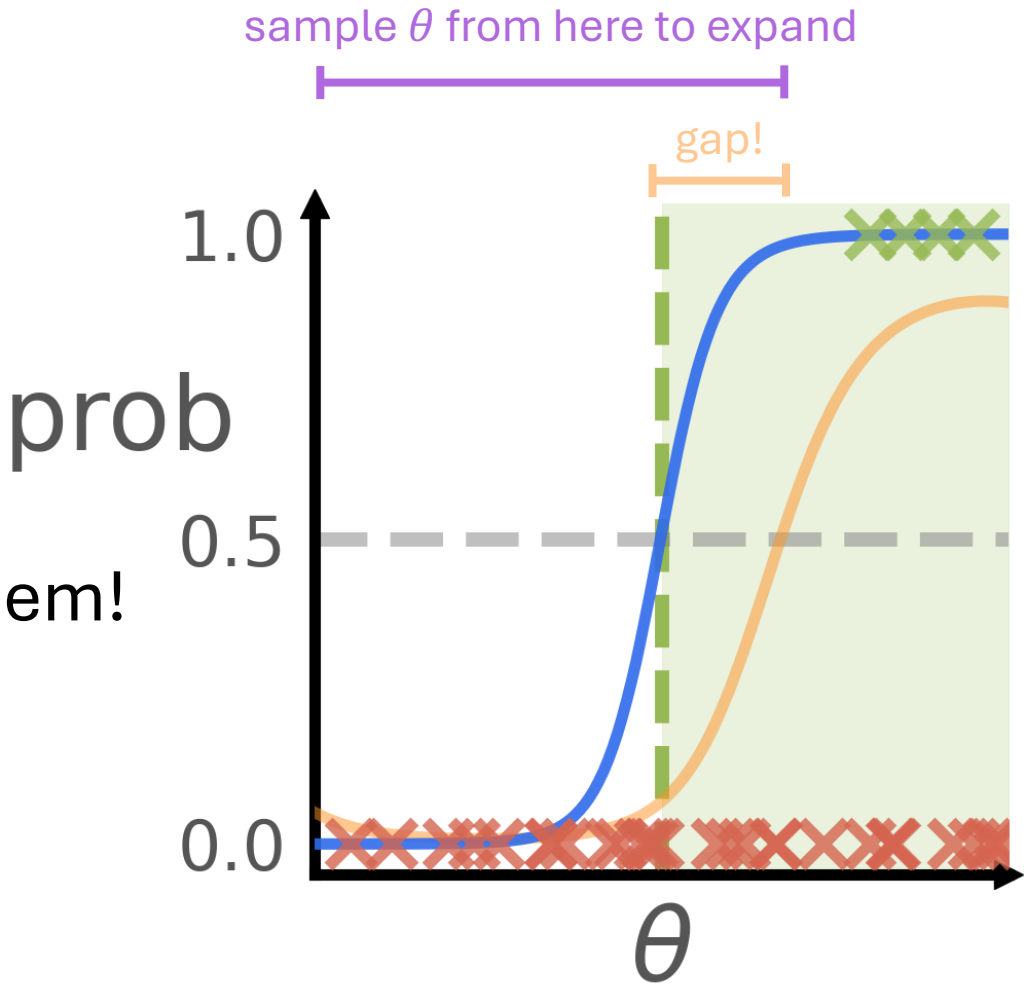
🤔 Estimated feasible set size depends on the quality of the policy.

To expand, need to render feasible but currently unsafe θ as safe.

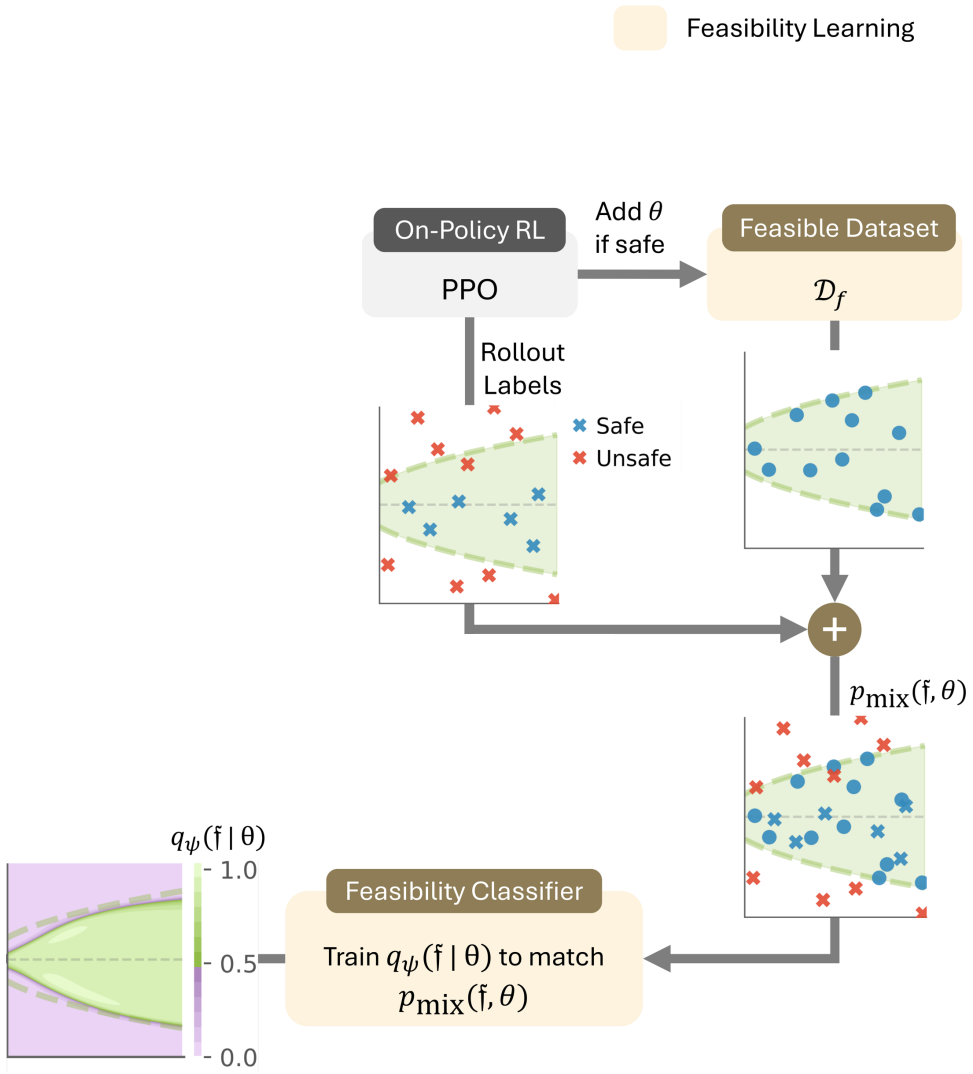
- 😵‍💫 Which θ are actually feasible?

💡 **Solution**: Rejection sample on all of them!

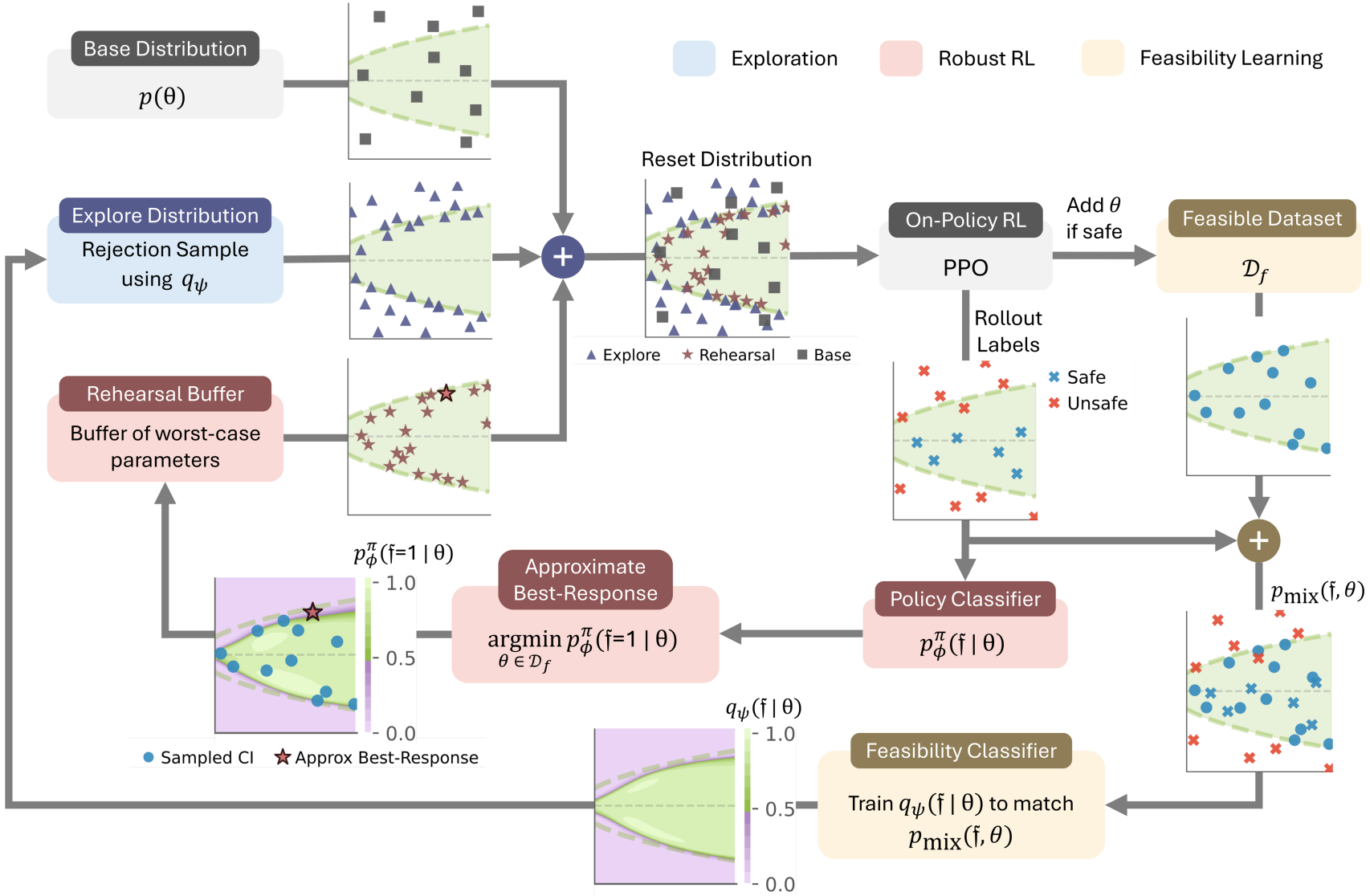
$$\theta \sim p(\cdot \mid \text{predict } \textbf{infeasible})$$



Expanding the Feasible Set Estimation | Implementation

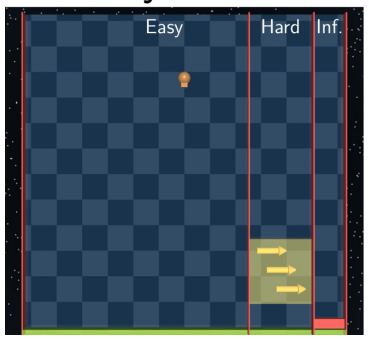


Putting FGE Together

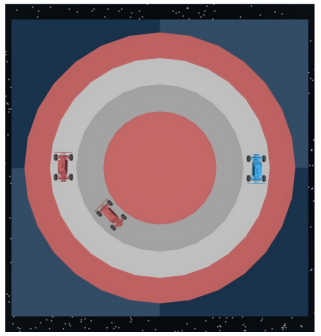


Results | Tasks

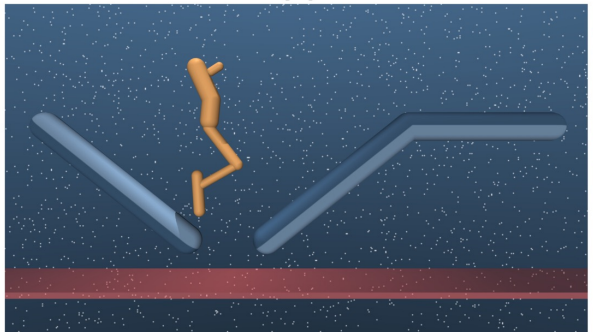
ToyLevels



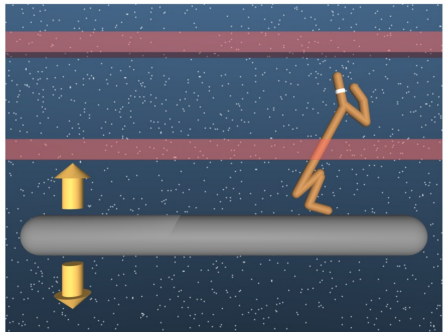
Dubins



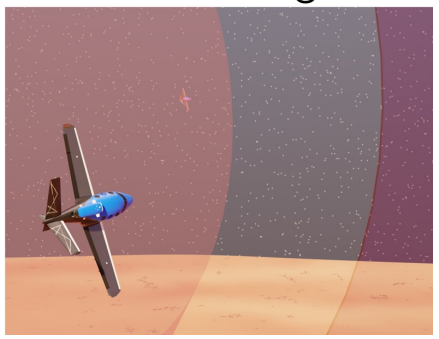
Hopper



HalfCheetah

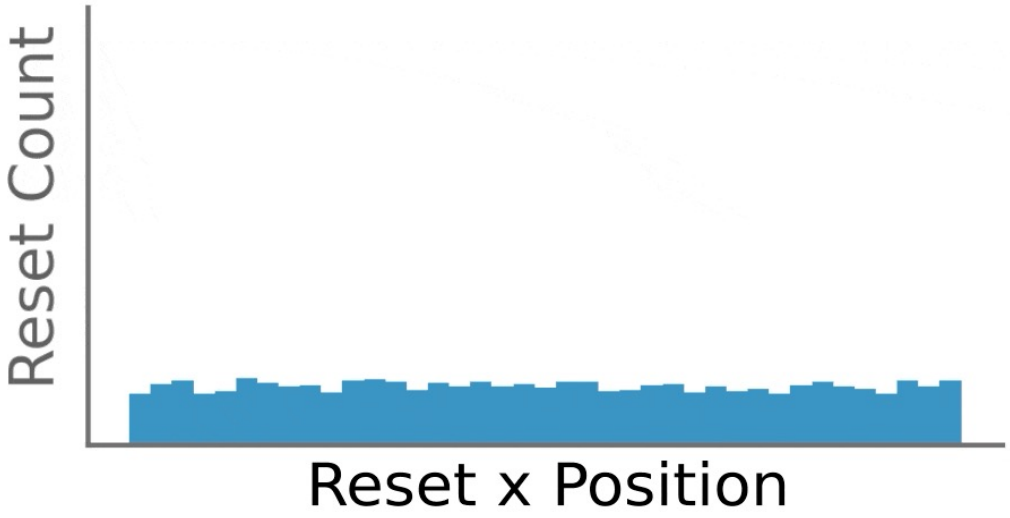


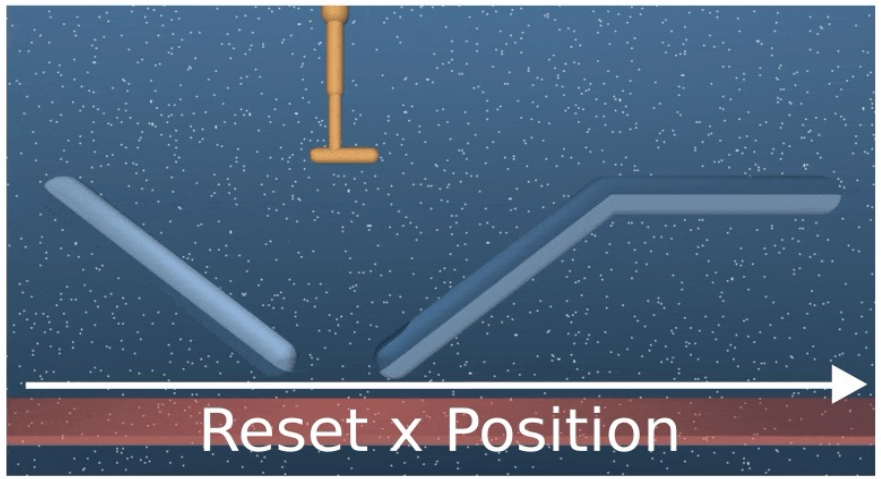
FixedWing



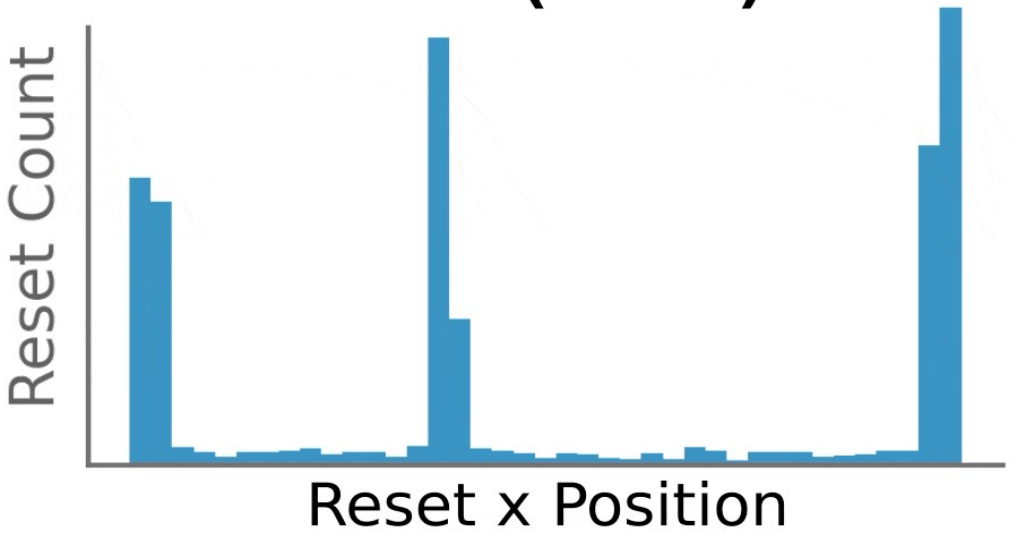
Results | Hopper

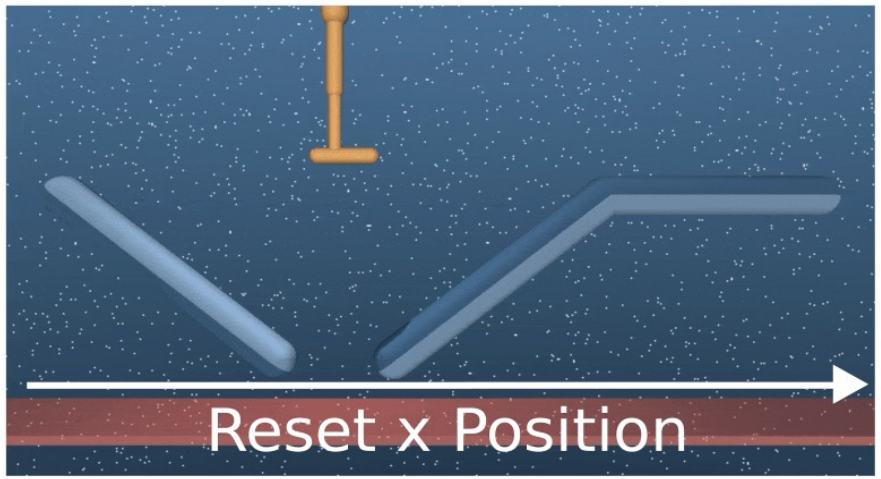
Domain Randomization



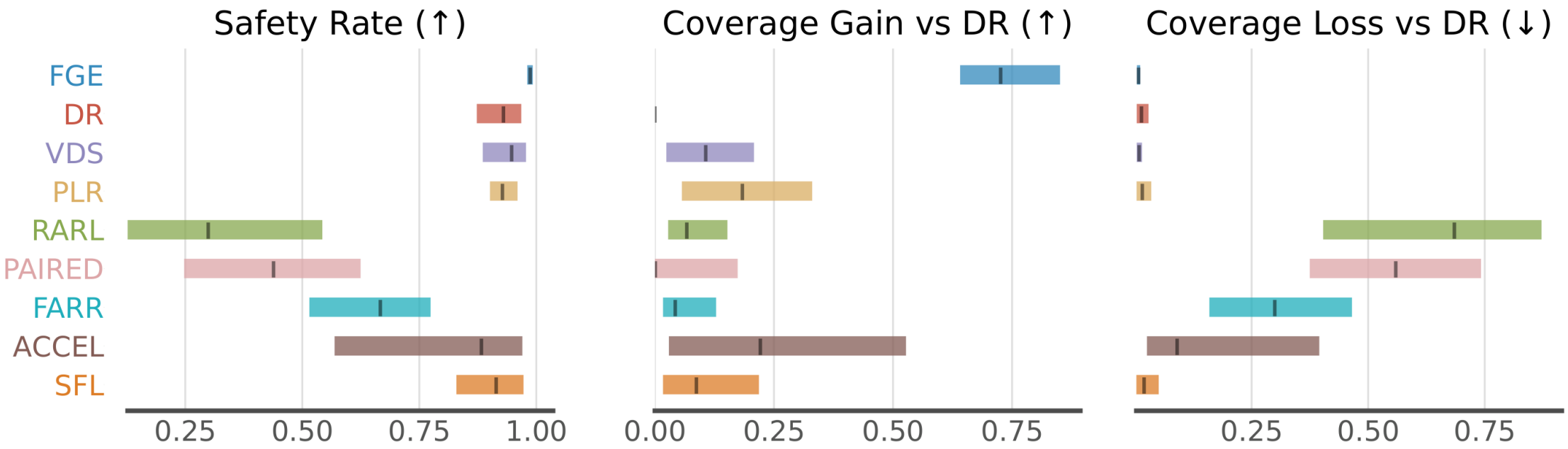


FGE (Ours)





Results | Overall Performance



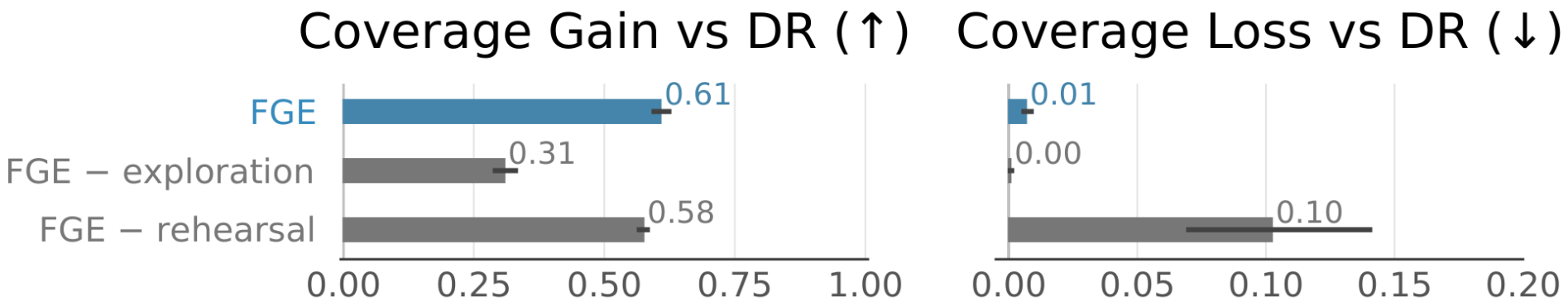
Ablation Studies

(Q1) Is the explore distribution important?

(Q2) Is the rehearsal buffer important?

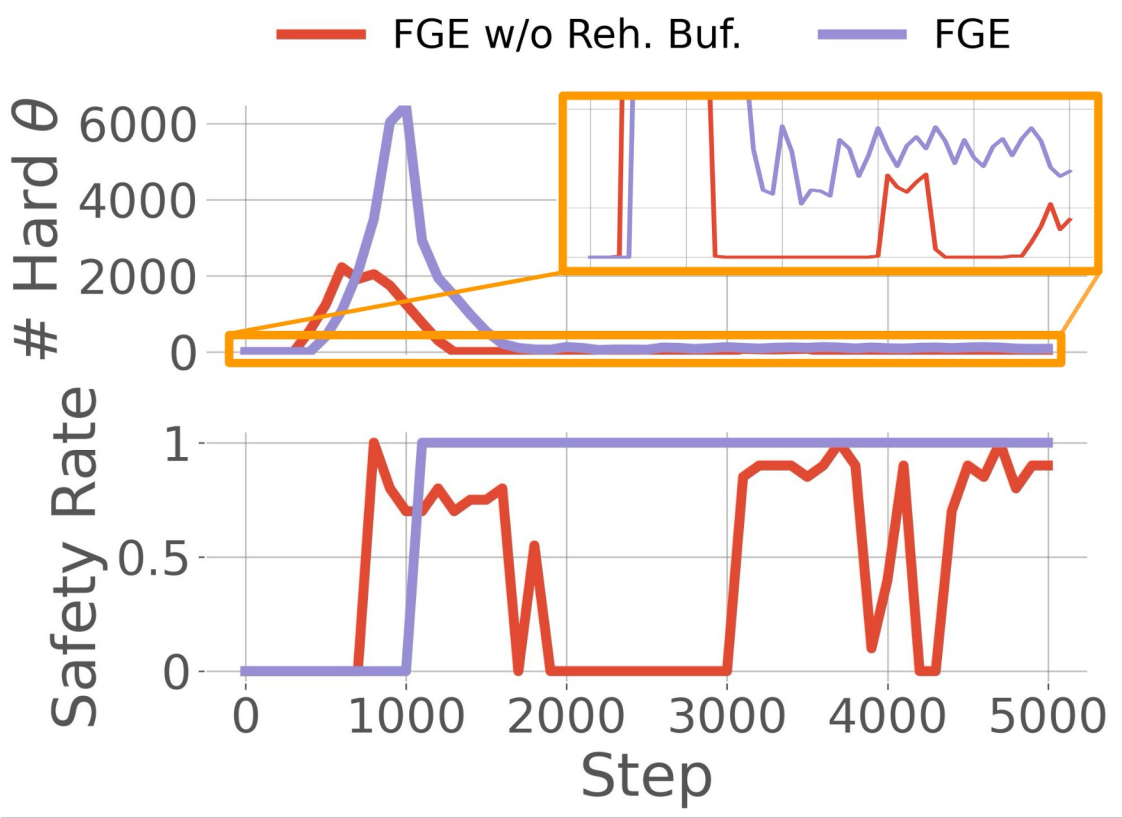
(Q3) Is the feasibility classifier important?

(Q1) Is the explore distribution important?



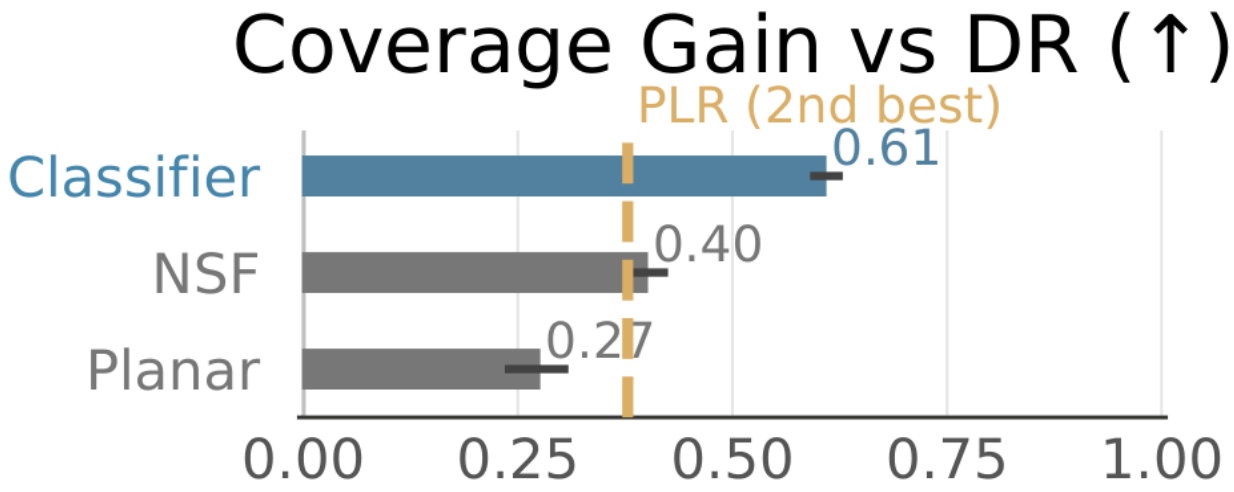
(A) Yes. Removing it leads to smaller feasible sets, and smaller coverage gain vs DR.

(Q2) Is the rehearsal buffer important?



(A) Yes. Removing it leads to catastrophic forgetting on the feasible set.

(Q3) Is the feasibility classifier important?



(A) Yes. Using a density model as a classifier instead degrades performance, largely due to thresholding issues.

Summary

- We propose **Feasibility Guided Exploration (FGE)** to improve safety under unknown feasibility using RL.