Safe Reinforcement Learning

Clare Lyle

University of Oxford

Outline

What is safety?

Constrained RL

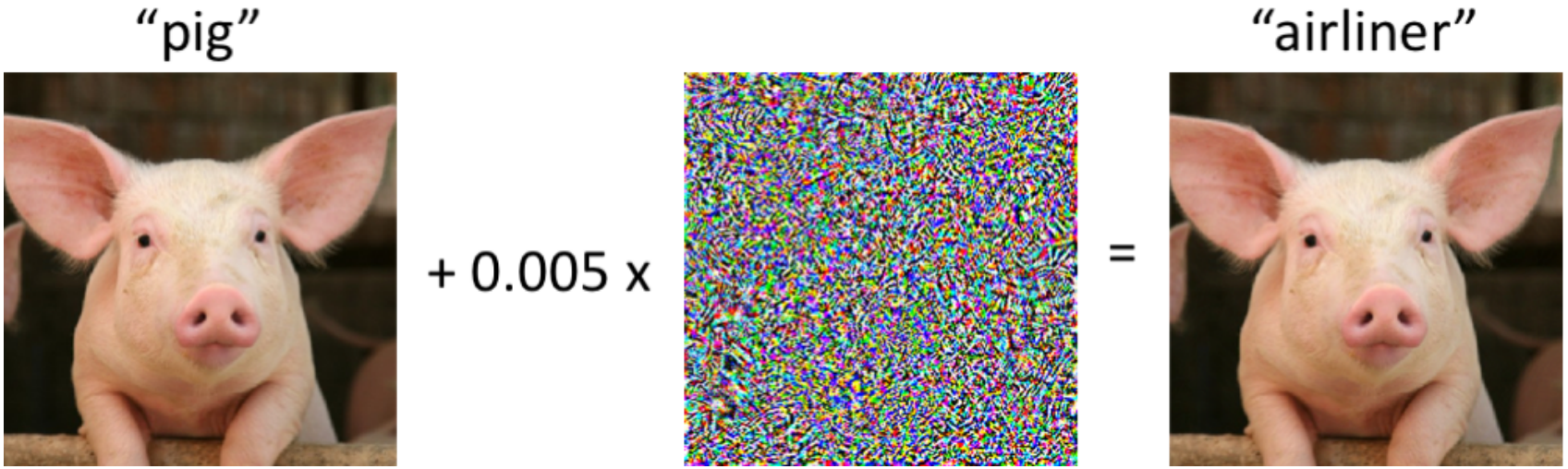
Safe Exploration

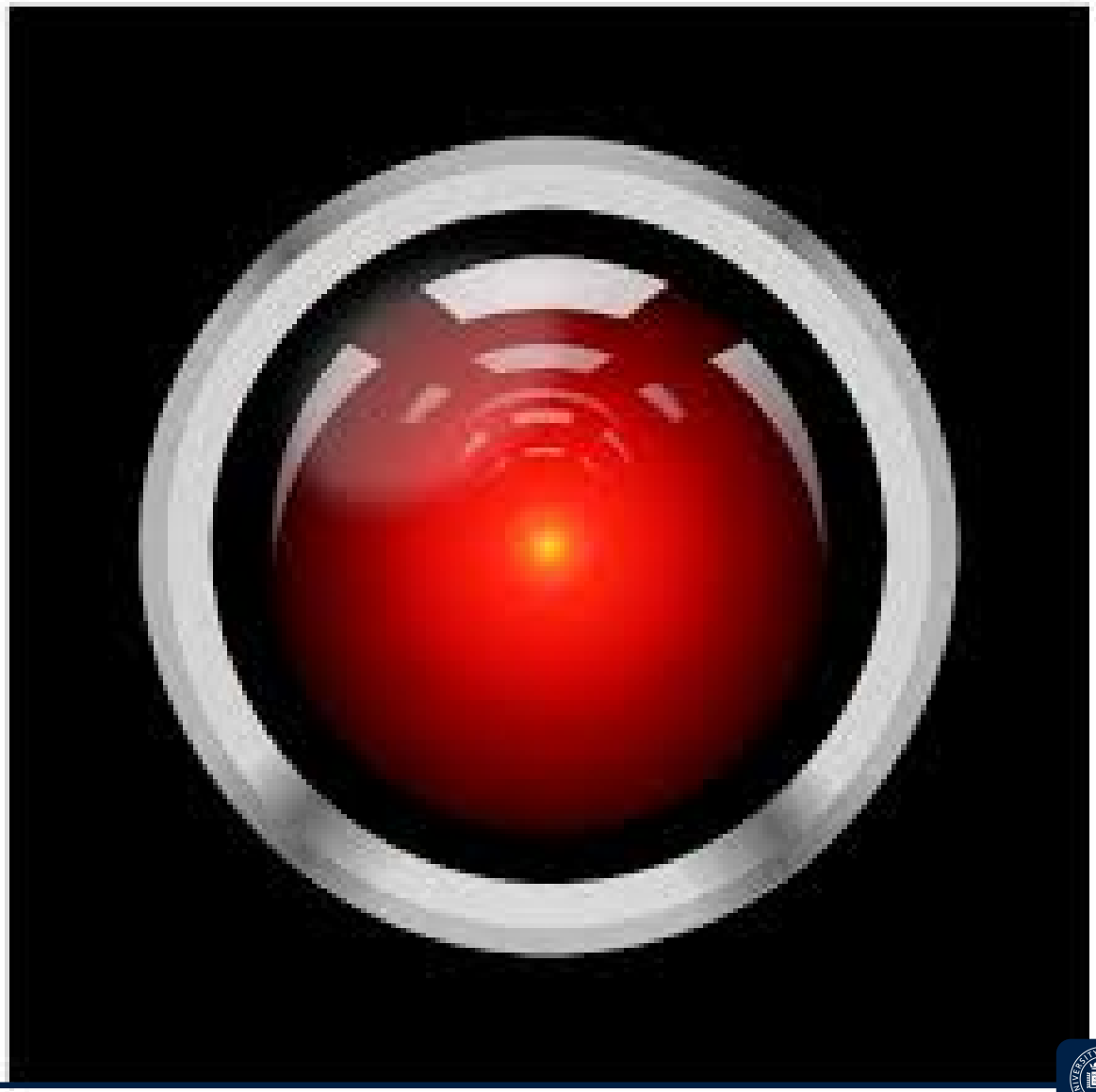
Buggy specifications

Figure: 737 MAX

Distributional shift

Robustness to perturbation





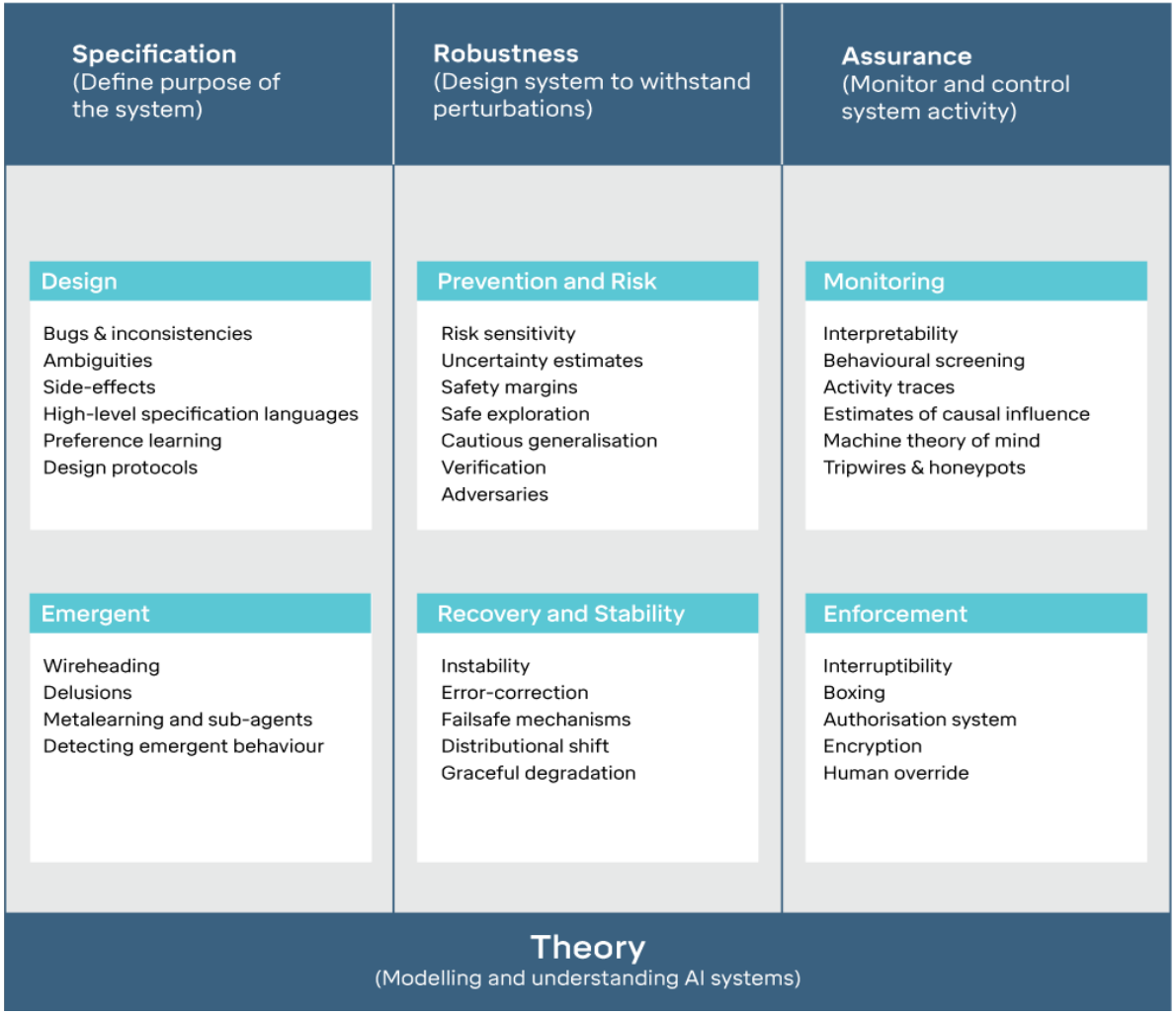
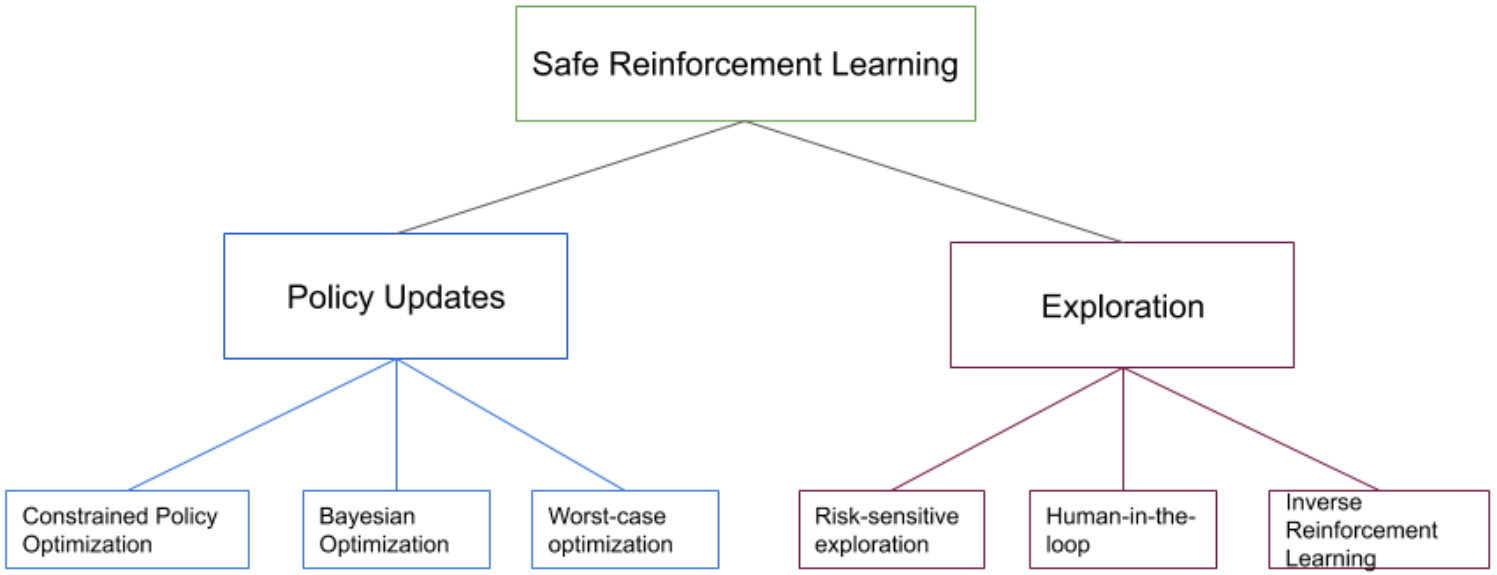


Figure: DeepMind's AI Safety Framework

Safe Reinforcement Learning: Definitions



Definition 1

A constrained Markov decision process (CMDP) is an MDP augmented with constraints $C_1, \ldots, C_n : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and limits $d_1, \ldots, d_n$ that restrict the set of allowable policies for that MDP. We denote by Π_C the set of policies satisfying $\mathbb{E}_{a_t, s_t \sim P^\pi} \sum \gamma^t C_i(s_t, a_t) \leq d_i$.

Definition 1

A policy is *safe* if it satisfies the constraints of our CMDP.

Constrained Policy Optimization is an algorithm intended to (with high probability) satisfy Definition 1 of safety. It performs policy optimization over the set of feasible policies.

$$\pi_{k+1} = \arg\max_{\pi \in \Pi_\theta} J(\pi) : D(\pi, \pi_k) \leq \delta, J_{C_i}(\pi) \leq d_i$$

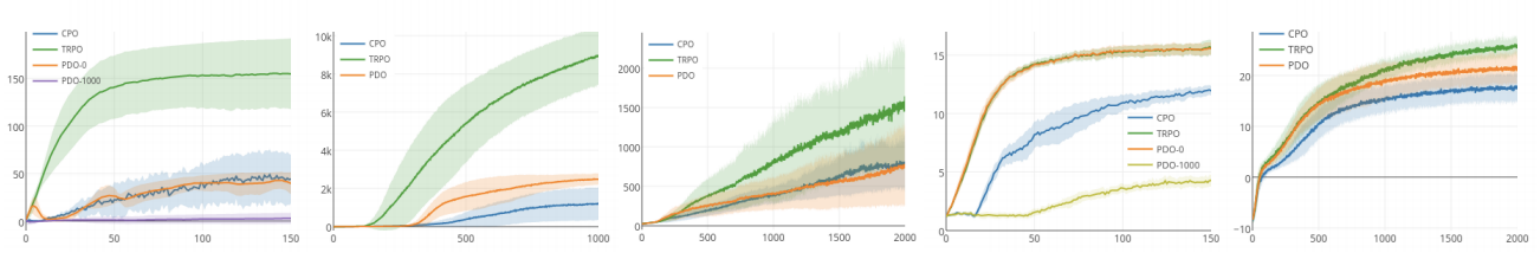
The above is intractable, and so practical algorithms use the following approximation:

$$\pi_{k+1} = \arg\max_{\pi \in \Pi_\theta} \mathbb{E}_{s,a \sim P^{\pi_k}} A^{\pi_k}(s,a)] - \alpha_k \sqrt{D_{KL}(\pi || \pi_k)}$$

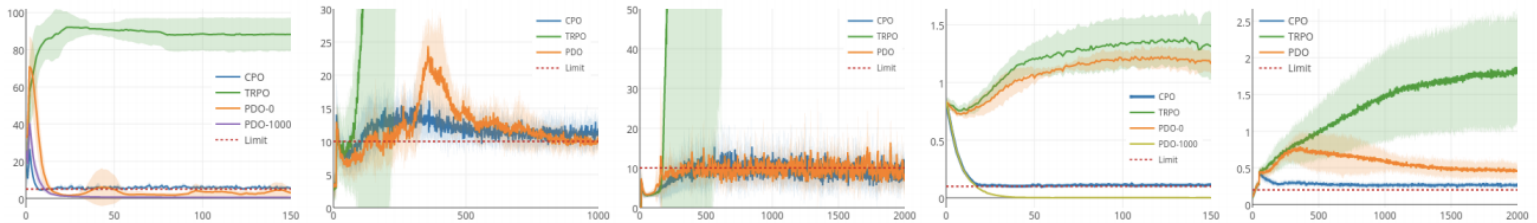
$$\text{s.t. } J_{C_i}(\pi_k) + \mathbb{E} \frac{A_{C_i}^{\pi_k}(s,a)}{1-\gamma} + \beta_k^i \sqrt{D_{KL}(\pi || \pi_k)} \leq d_i$$

CPO

Returns:



Constraint values: (closer to the limit is better)



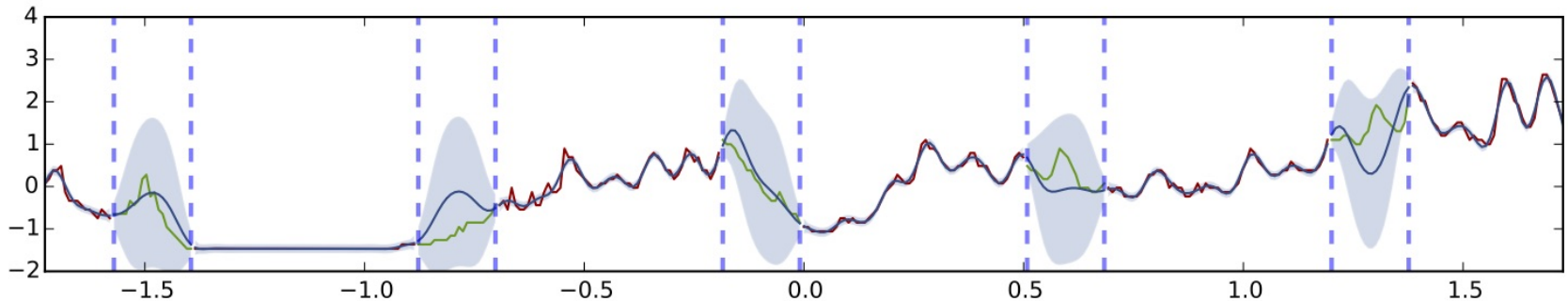
(a) Point-Circle (b) Ant-Circle (c) Humanoid-Circle (d) Point-Gather (e) Ant-Gather

For special classes of constraints, no need to approximate!
(To be continued soon...)

- Bayesians are experts in uncertainty
- Bayesian RL allows us to model how confident we are that we can predict the effects of actions.
- Bayesian methods can be model-free (we estimate uncertainty in the value function) or model-based (we estimate uncertainty in the reward and transition kernel separately)
- We'll look at a model-based approach in this talk

Safe RL with Gaussian Processes

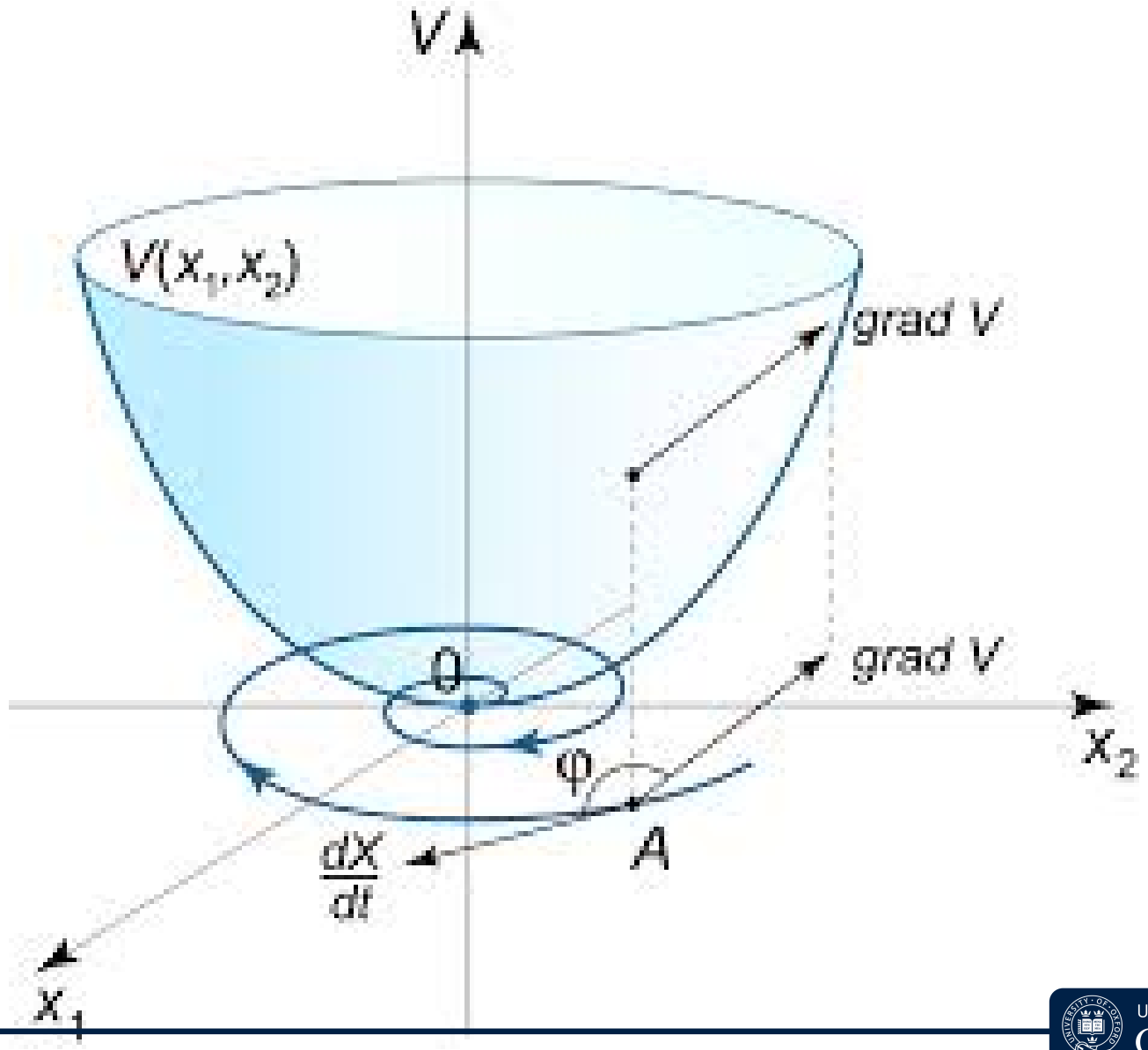
Use a Gaussian Process to get confidence intervals on the environment transition function, and use sampled data to gradually expand a set of 'safe' states (assumes Lipschitz dynamics!)



Lyapunov Function

Lyapunov functions come from the world of dynamical systems and ODEs. They're used to show stability of solutions.
Since an MDP can be thought of as a dynamical system and we're interested in finding stable policies, they can be a useful tool.

Lyapunov Function Illustration



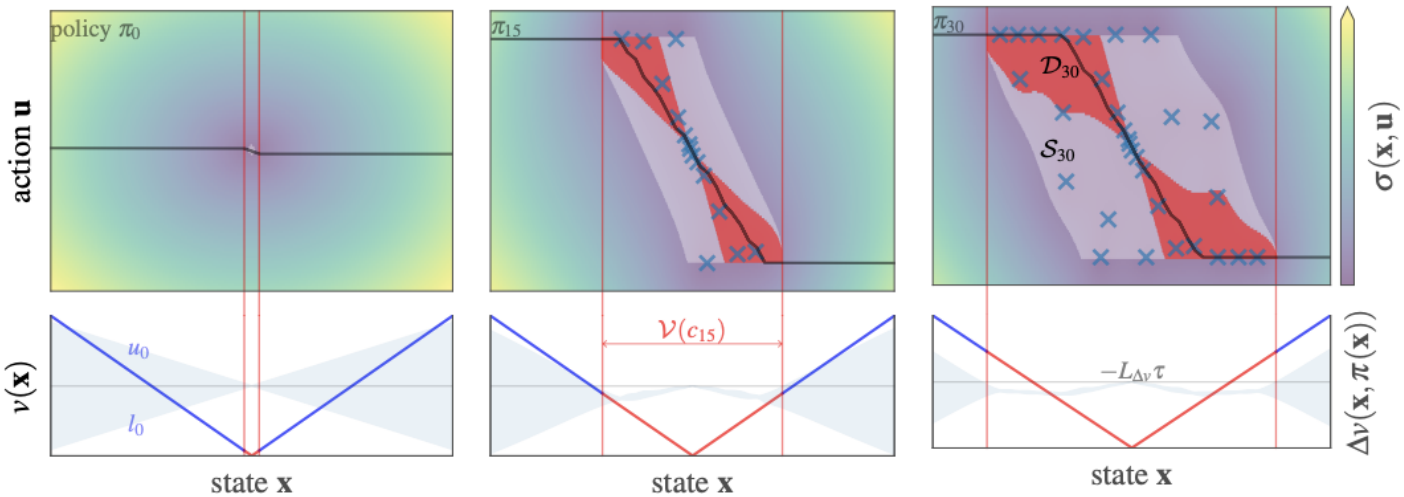
Assumptions for Safe RL with Stability Guarantees

- Model is Lipschitz continuous
- There exists a Lyapunov function for the dynamics of the MDP and policies π
- Initial safe policy

2 main steps:

1. Compute approximation of the region of attraction for policy π_n (find a level set of the Lyapunov function where the environment dynamics decrease the Lyapunov function with high probability)
2. Optimize Lagrangian of constrained policy optimization problem (use GP confidence intervals)

Visualizing the Lyapunov Level Sets



(a) Initial safe set (in red). (b) Exploration: 15 data points. (c) Final policy after 30 evaluations.

Definition 2

A reinforcement learning algorithm is *safe* if it never performs an action that brings its expected return below a specified value L.

Worst-case Criterion

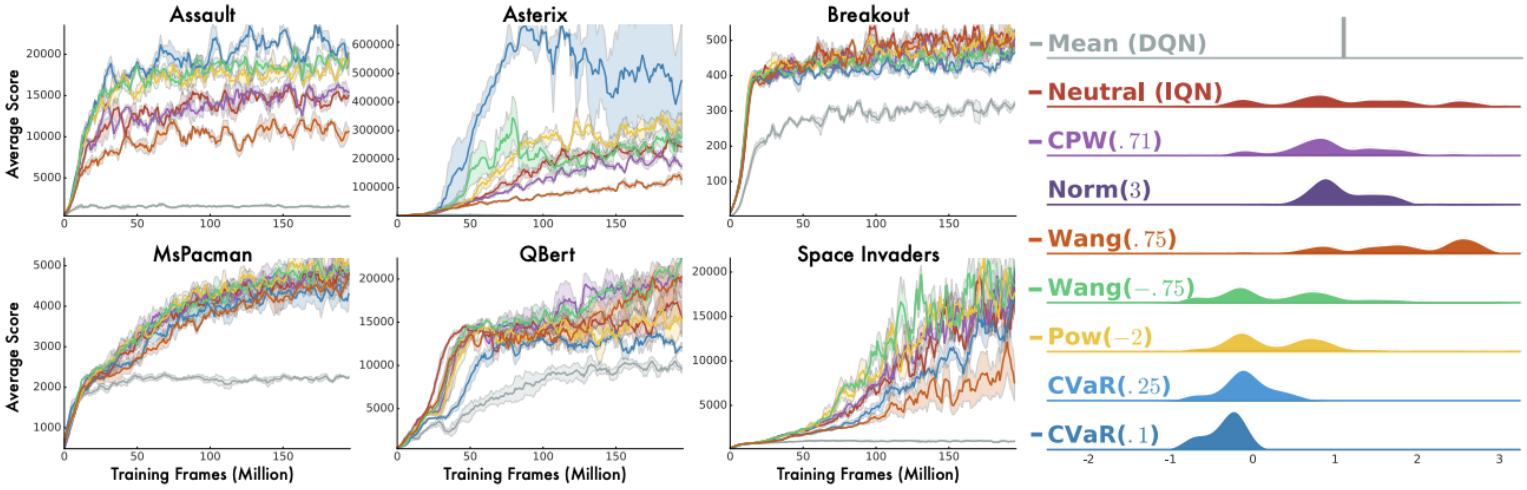
When the agent is uncertain over its model, it can attempt to maximize its *worst-case* reward.

$$\max_{\pi} \min_{p} \mathbb{E}_{\pi,p} \sum_{t=1}^{\infty} \gamma^t r_t$$

Risk-sensitive exploration

- Concerned with value function only
- Model a *distribution* of returns
- Specify how **risk-averse** your policy should be
- **Re-weight** your distribution according to this risk aversion when computing the value of each state

Risk-sensitive exploration

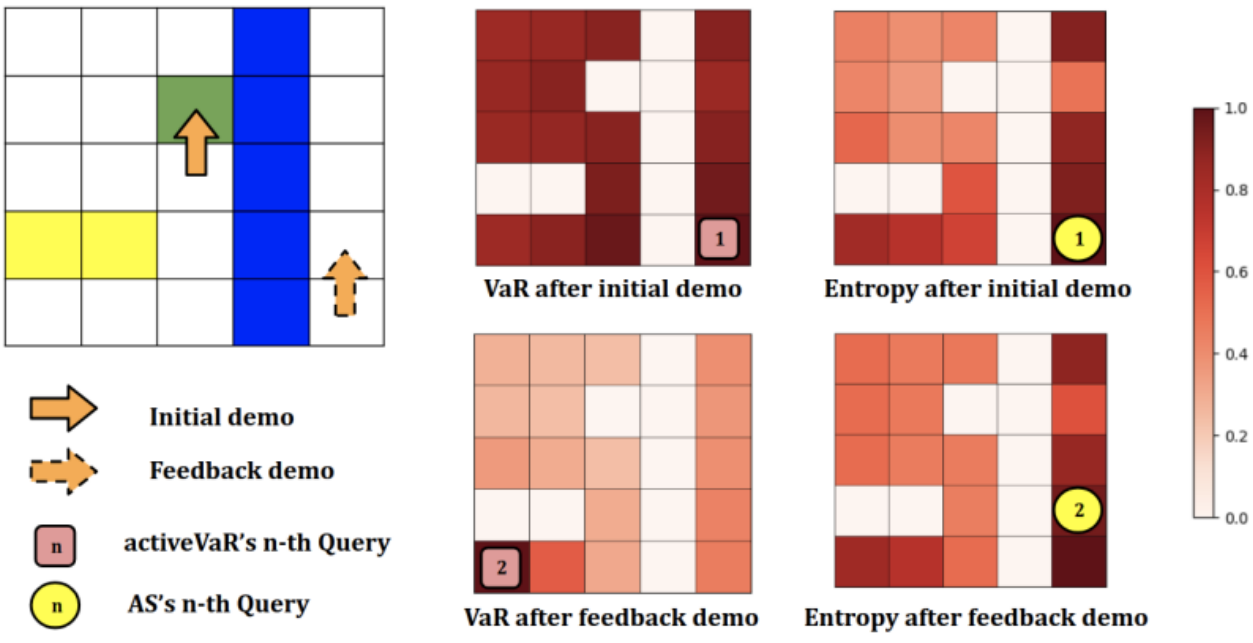


Inverse reinforcement learning (IRL): the agent is given a set of demonstrations, from which it infers a reward function which it then optimizes for.

Active inverse reinforcement learning

Active IRL permits the agent to query the demonstrator when it encounters states about which its reward model is highly uncertain.
For example, if a self-driving car encounters an unusual weather condition, it may ask the driver to take over.

Active inverse reinforcement learning



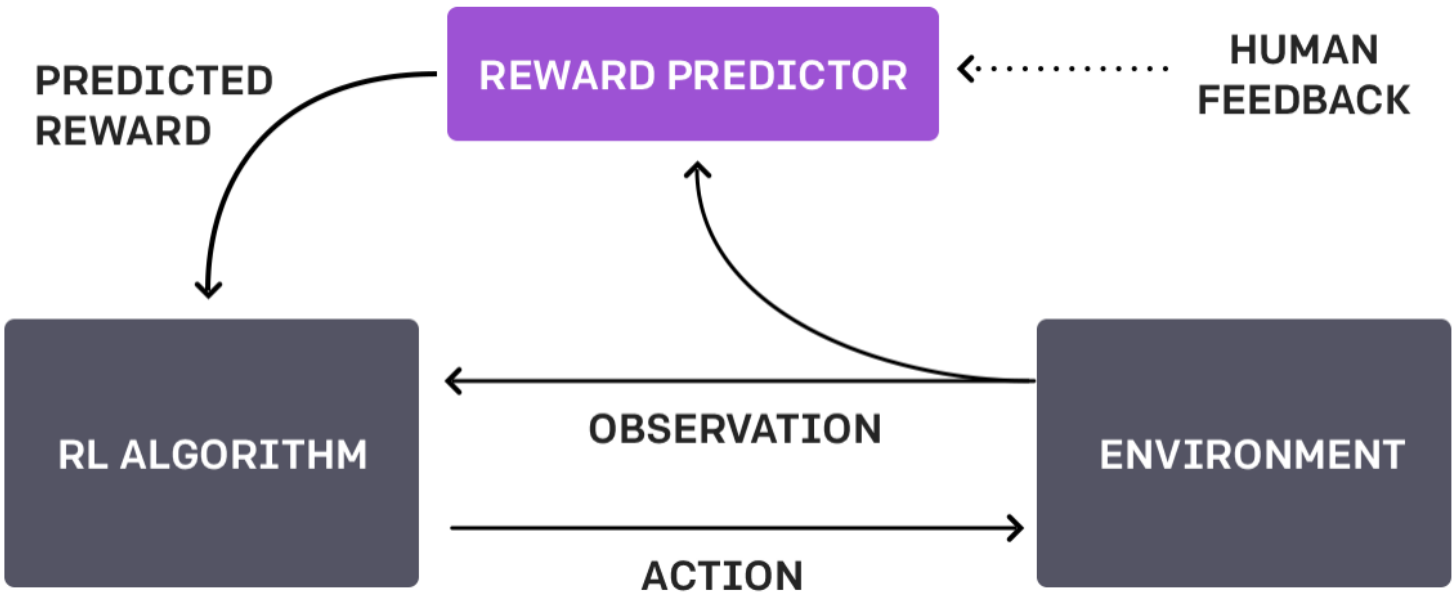
A Philosophical Note

The next paper I'll present is arguably not a 'safe RL' paper.
Is capturing human preference within the scope of RL?

Learning human preferences

We can train a model to predict human preferences, and then use that as our reward function instead of relying on the environment.

Learning human preferences



A video

https://www.youtube.com/watch?time$_c ontinue = 1v = oC7Cw3fu3gU$

- Safe RL can be formalized in many ways.
- Two main categories: finding safe policies, and avoiding catastrophic actions in early phases of training.
- We're still not very good at it.