On Optimizing Interventions in Shared Autonomy

UMass Amherst CICS
Weihao Tan
Siddhant Pradhan
David Koleczek
Yash Chandak (PhD)
Nicholas Perello (PhD)

Microsoft
H M Sajjad Hossain
Vivek Chettiar
Vishal Rohra
Nan Ma
Aaslesha Rajaram
Soundar Srinivasan

Introduction

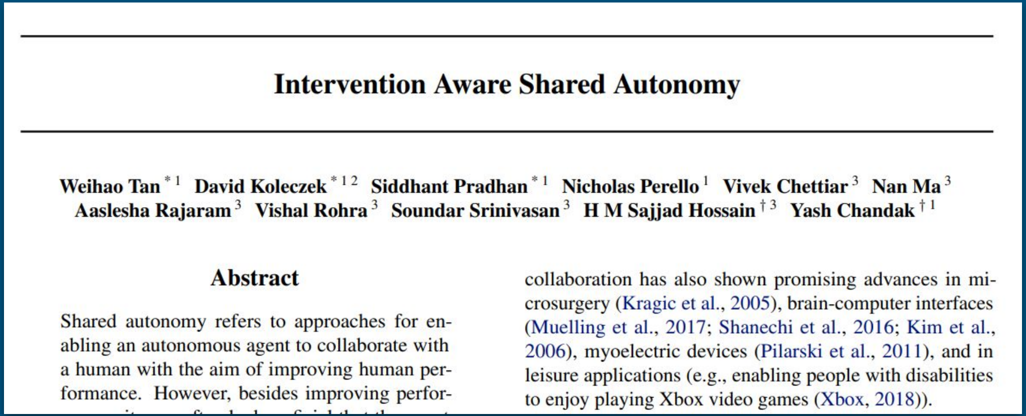
UMass CS 696DS

Primary Contributors (UMass)
- Weihao Tan (M.S.)
- Siddhant Pradhan (M.S.)
- Nicholas Perello (PhD candidate)

Advisors
- Yash Chandak (PhD candidate)
- H M Sajjad Hossain (Data Scientist, MAIDAP)

Microsoft's AI Development Acceleration Program (MAIDAP)
- Vivek, Nan, Vishal, Soundar, Aaslesha

Feedback from Microsoft Research
- Sam Devlin (Senior Researcher)



Intervention Aware Shared Autonomy

Weihao Tan [*1] **David Koleczek** [*12] **Siddhant Pradhan** [*1] **Nicholas Perello** [1] **Vivek Chettiar** [3] **Nan Ma** [3] **Aaslesha Rajaram** [3] **Vishal Rohra** [3] **Soundar Srinivasan** [3] **H M Sajjad Hossain** [†3] **Yash Chandak** [†1]

Abstract

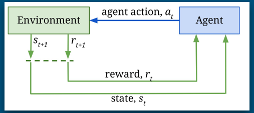
Shared autonomy refers to approaches for enabling an autonomous agent to collaborate with a human with the aim of improving human performance. However, besides improving perfor-

collaboration has also shown promising advances in microsurgery (Kragic et al., 2005), brain-computer interfaces (Muelling et al., 2017; Shanechi et al., 2016; Kim et al., 2006), myoelectric devices (Pilarski et al., 2011), and in leisure applications (e.g., enabling people with disabilities to enjoy playing Xbox video games (Xbox, 2018)).

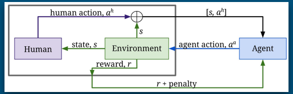
Outline

- The Problem

- Reinforcement Learning



- Our Method



- Evaluation

- Next Steps & Challenges

The Problem

Motivation

Xbox Adaptive Controller

- Helps people with disabilities enjoy games

Copilot Feature

- Two users can share “one controller” by combining the input from two controllers

The Problem

Can we work towards a general *autonomous agent* that assists someone with special needs in playing a game without the need for a second person?

Shared Autonomy

A broader area of research about **Human-AI collaboration**

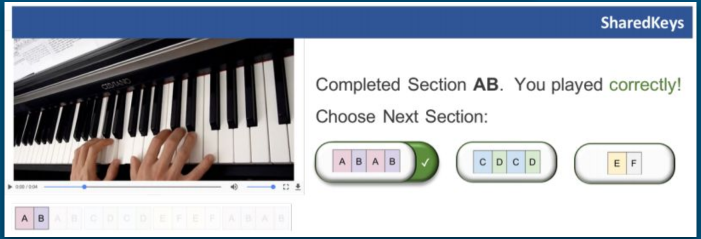
Human input is combined with semi-autonomous control to achieve a **common goal**.

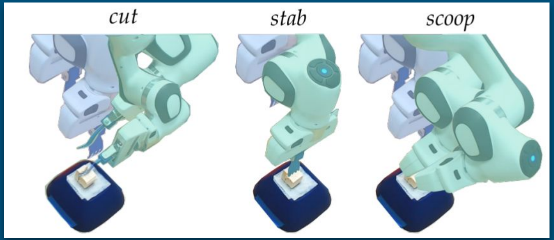
- AI inference of human goals
- Learning efficiently from limited human data
- Brain-computer interfaces
- **User experience**
- And much more...

Optimizing Interventions in Shared Autonomy

Our focus is on games, but so many other applications

Shared Control





Optimizing Interventions in Shared Autonomy

Imagine an agent assisting you play *Super Mario Bros.*

- If you are an **expert**...
- If it is your **first time** playing...
- If you are only able to use a **subset of available controls**...

Optimizing Interventions in Shared Autonomy

Goal

- Maximize **performance** at the task
- Minimize **perceived impact** of the agent (maximize user satisfaction)

How do we quantify the **experience** of a human?

- Subjective and no single, correct answer

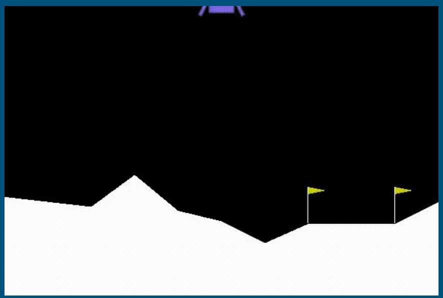
Interventions

- Consider the discrete number of **interventions** made by the agent
- **Intervention rate**: *interventions / time*

Experimental Setup - Simulated Humans

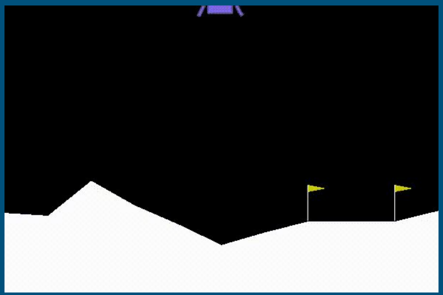
OpenAI Gym: Lunar Lander

No-op



Sensor

Laggy



Noisy

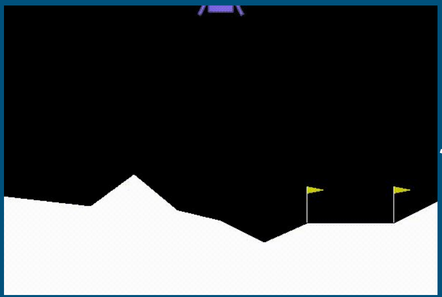
Super Mario Bros.

Noisy

David Koleczek

Problem Setup

User who needs assistance

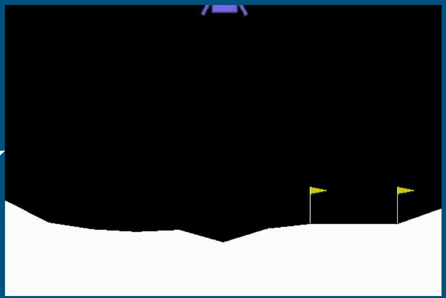


and

Level of assistance

Autonomous Agent

More success at the task



and

User Satisfaction = Interventions

Reinforcement Learning

Intro to Reinforcement Learning

How do you go from **real world interactions**, like playing a game...

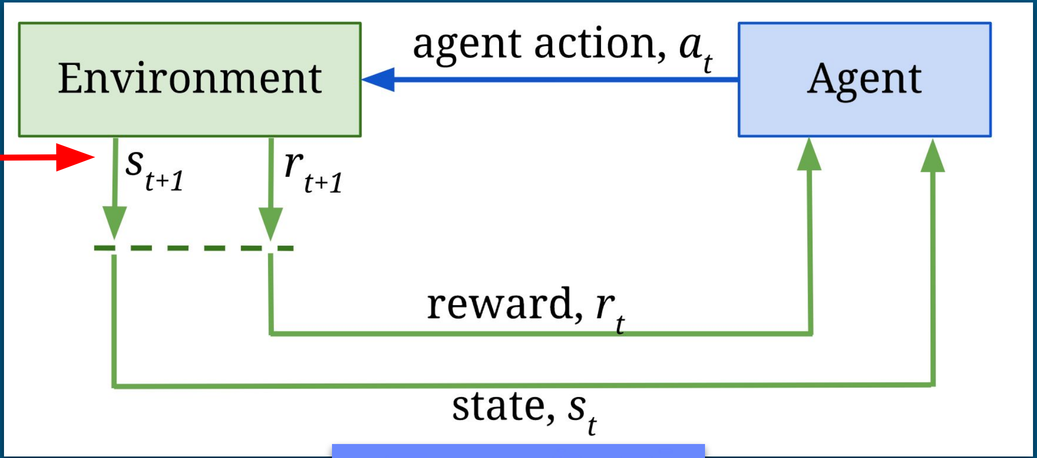
To a mathematically precise **decision making process** that you can optimize.

- How to build an automated system to play the game?

Reinforcement Learning!

Agent-Environment Interaction

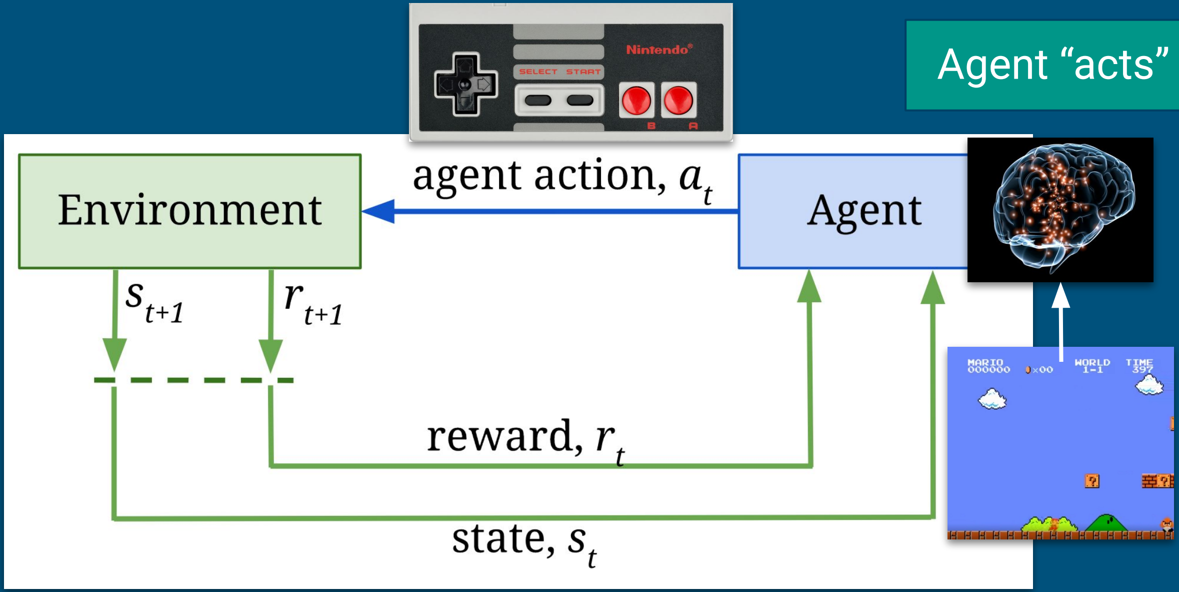
Start here



State the agent observes

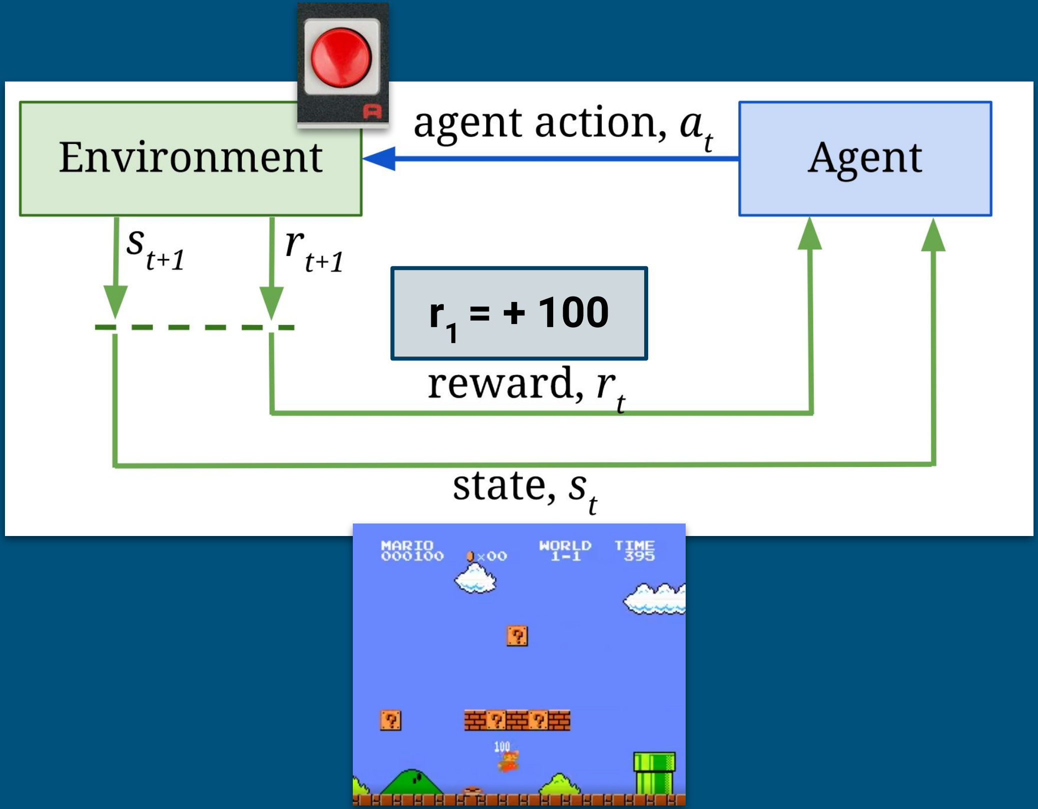
Agent-Environment Interaction

Agent "acts"



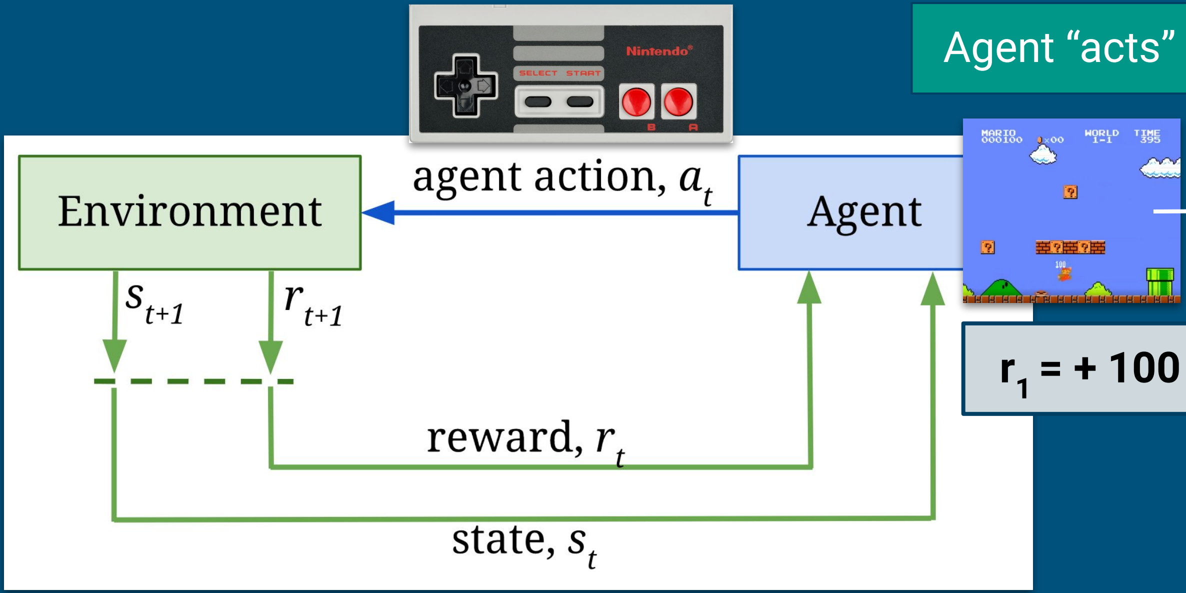
Agent-Environment Interaction

Environment "transitions"



Agent-Environment Interaction

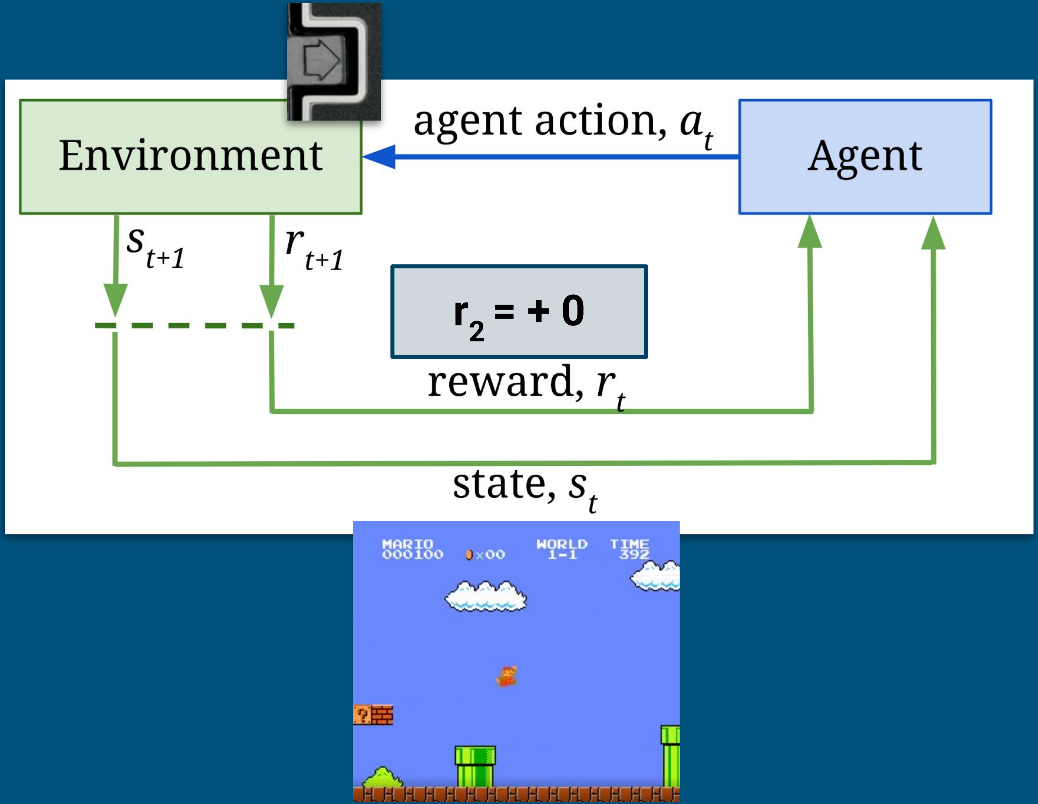
Agent "acts"



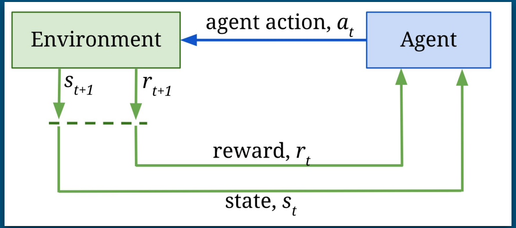
$r_1 = +100$

Agent-Environment Interaction

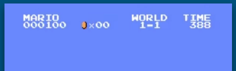
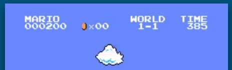
Environment “transitions”



David Koleczek

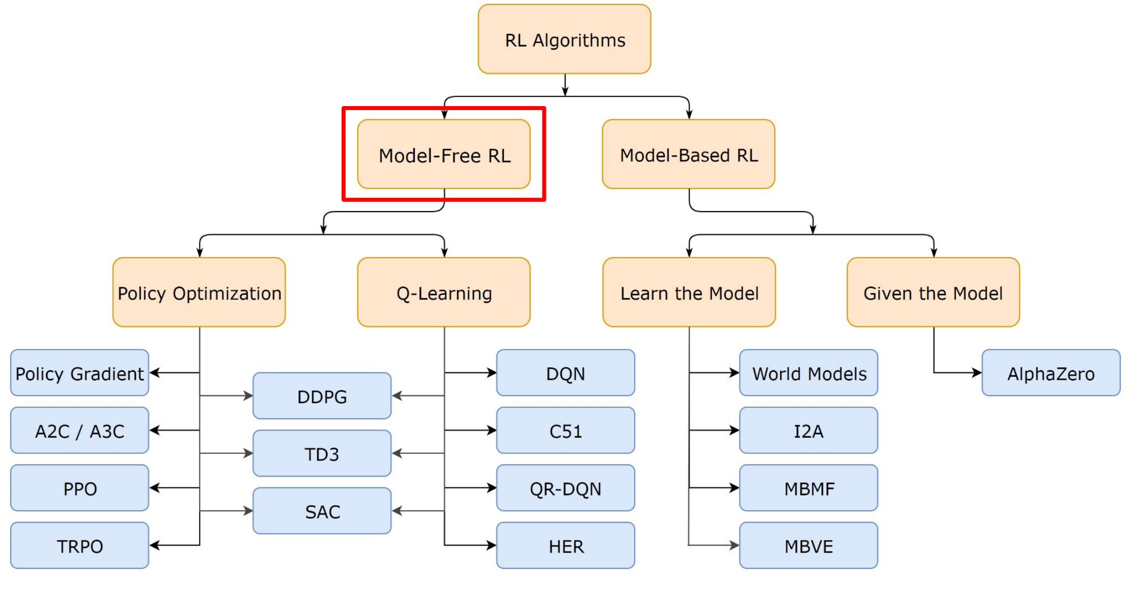


Dataset View of RL

State, *s*	Action, *a*	Reward, *r*	Next state, *s'*
		+ 100	
		0	
		0	
			

Learning Algorithms

How to train an agent to take optimal actions?

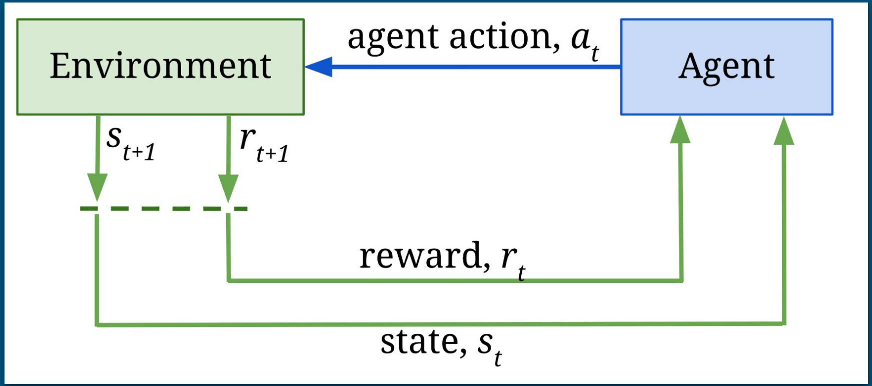


[Taxonomy of RL Algorithms - OpenAI]

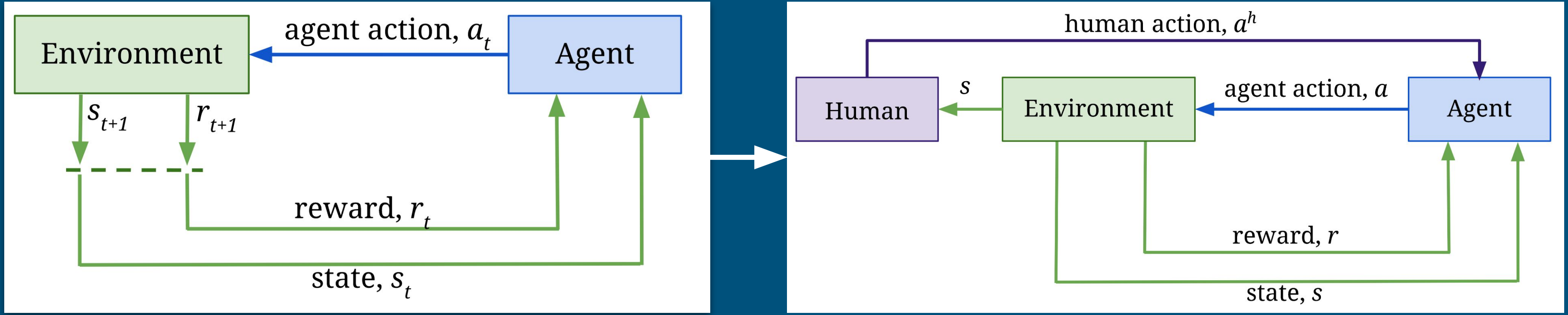
Our Method

Towards a Human in the Loop

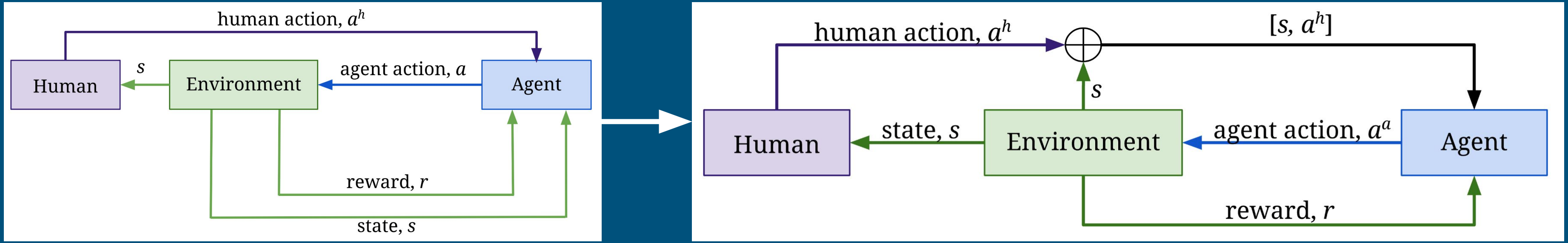
Goal: Use RL to train the agent to **modify** human actions to **balance increasing reward** and **minimizing interventions** (a proxy for satisfaction)



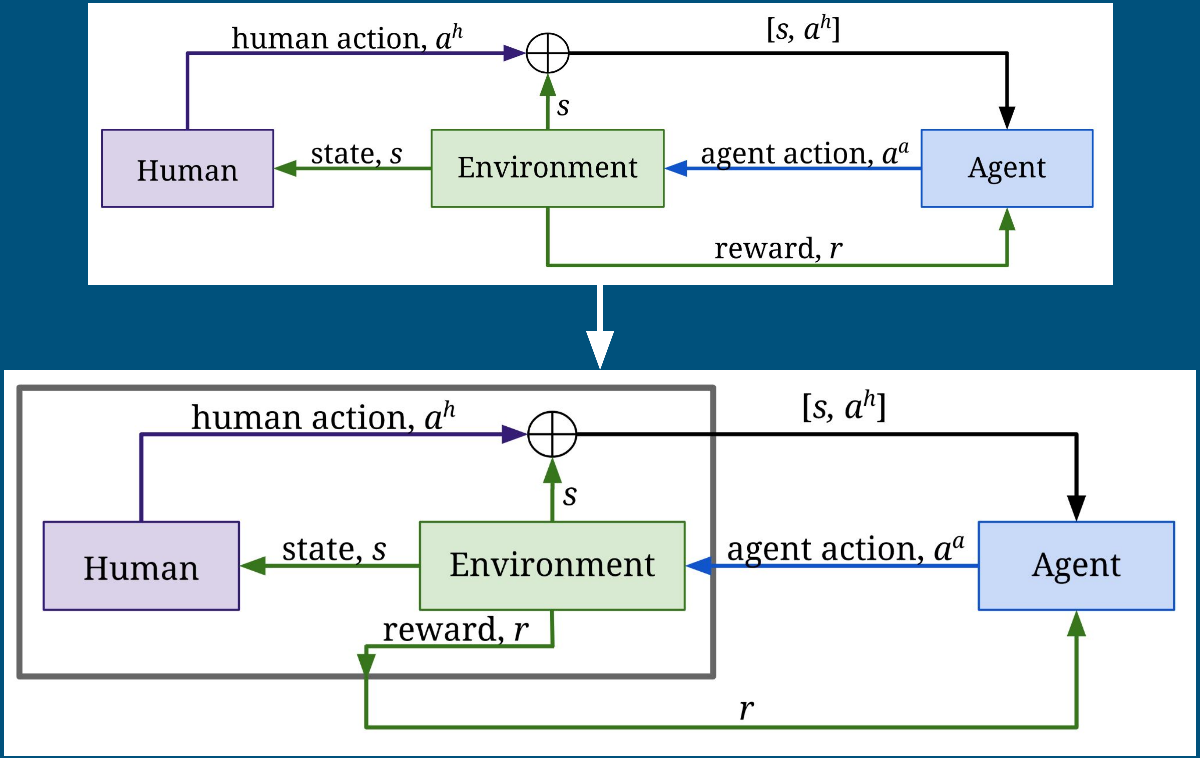
Towards a Human in the Loop



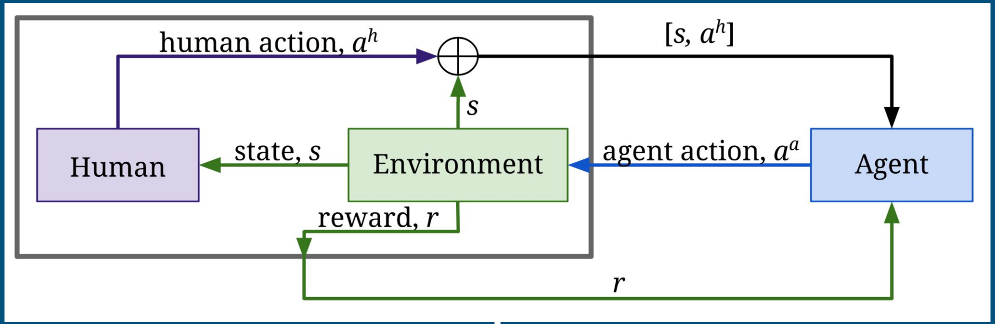
Towards a Human in the Loop

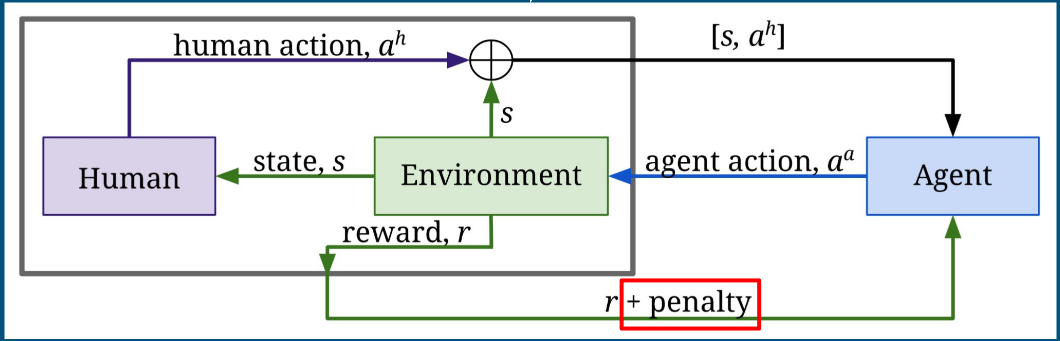


Towards a Human in the Loop



Soft Constrained Shared Autonomy





Soft Constrained Shared Autonomy - Penalty Adapting

How do you set the penalty? 🤷

- Propose Penalty Adapting to tune penalty during learning

Our objective

$$\pi_a^* = \max_{\pi_a} \mathbb{E}_{\pi_a, \pi_h} \underbrace{\left[\sum_{t=0}^{T} r(s_t, a_t) \right]}_{\text{Maximize total return}} \text{ s.t. } \underbrace{\mathbb{E}_{\pi_a, \pi_h} \left[\sum_{t=0}^{T} I(a_t^a, a_t^h) \right] \leq c}_{\text{Satisfy a constraint on interventions}}$$

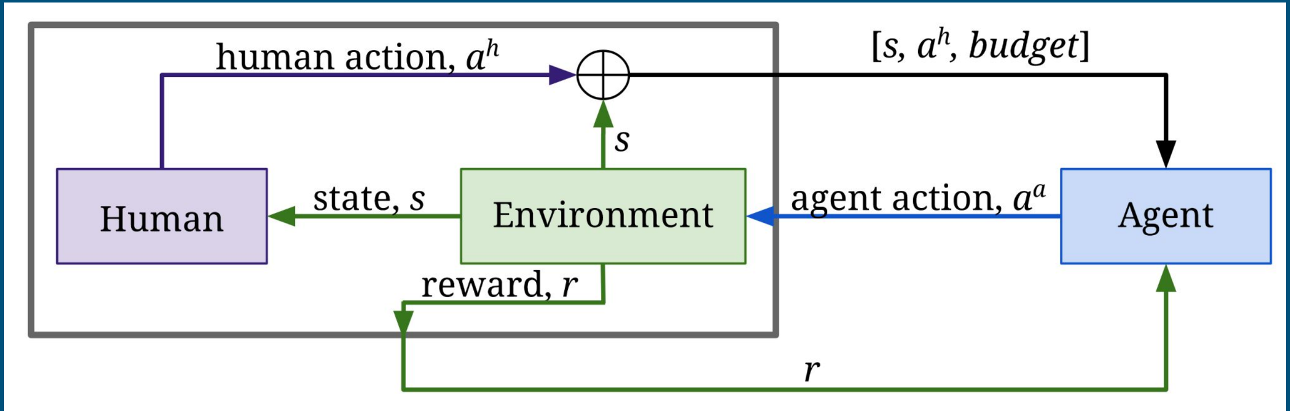
Written as a Lagrangian

$$\max_{\pi_a} \min_{\lambda \geq 0} \mathbb{E}_{\pi_a, \pi_h} \left(\underbrace{\sum_{t=0}^{T} r(s_t, a_t)}_{\text{Maximize using RL}} + \lambda \underbrace{\left(c - \sum_{t=0}^{T} I(a_t^a, a_t^h) \right)}_{\text{Minimize using gradient descent}} \right)$$

Hard Constrained Shared Autonomy

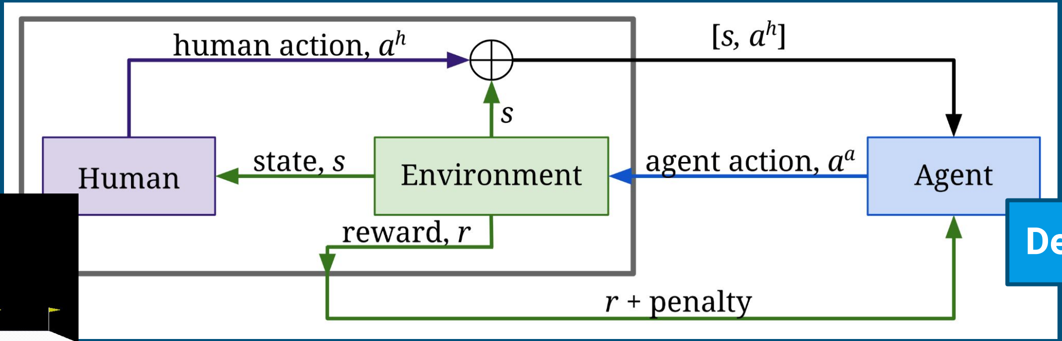
Set a ***budget***; the maximum number of times an agent can intervene

- budget > 0, the agent can intervene whenever it desires
- budget = 0, the agent can no longer intervene



Evaluation

Experimental Setup



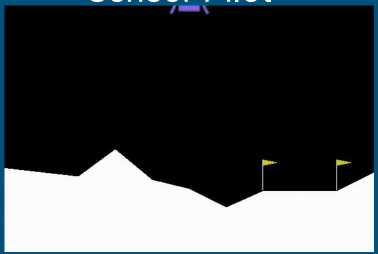
Deep Q-Learning

- 4 methods: Penalty, Penalty Adapting, Budget, Baseline*
- Use each of the 5 simulated humans
- Agents trained using Deep Q-Learning

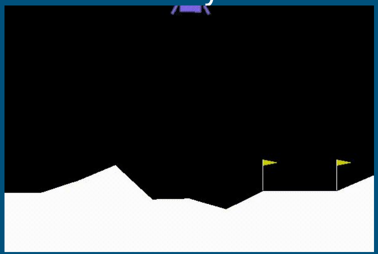
* Siddharth Reddy, Anca Dragan, and Sergey Levine. Shared autonomy via deep reinforcement learning. 2018.

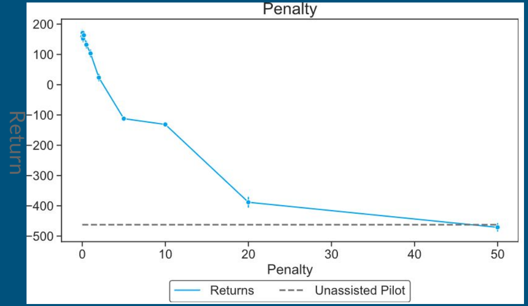
Penalty Method

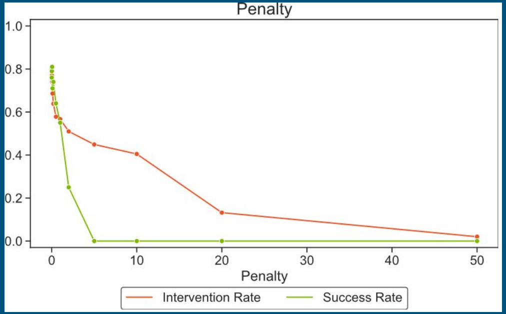
Sensor Pilot



Penalty = 50







Penalty = 10

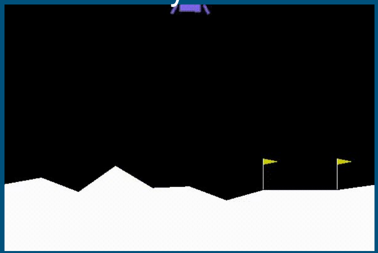
Penalty = 5

Penalty = 1

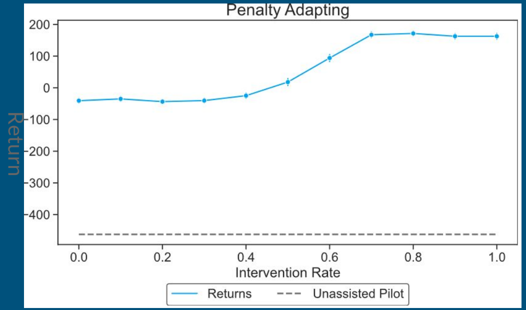
Penalty = 0.5

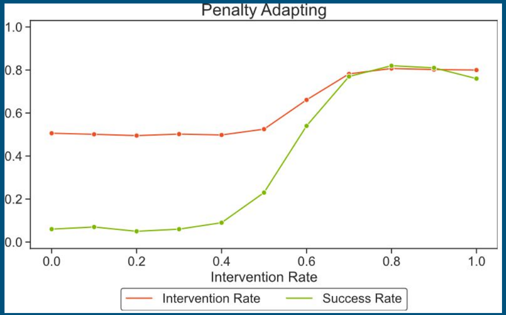
Penalty = 0.1

Penalty = 0.05

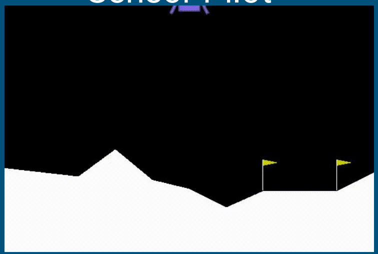


Penalty Adapting Method

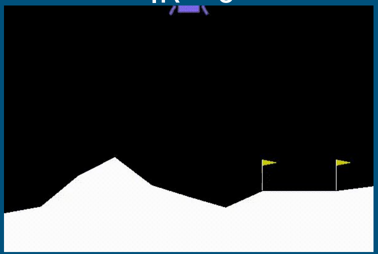




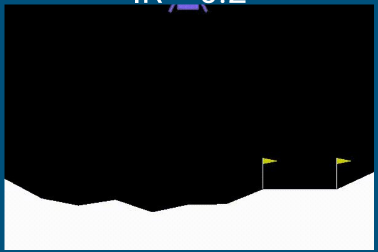
Sensor Pilot



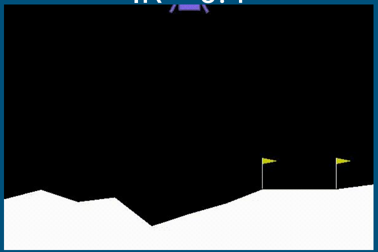
IR = 0



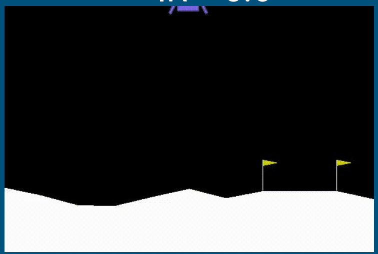
IR = 0.2



IR = 0.4

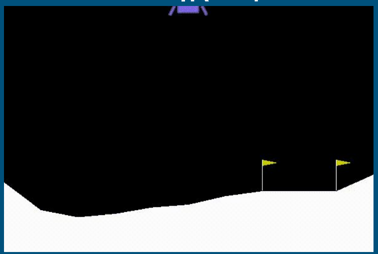


IR = 0.6

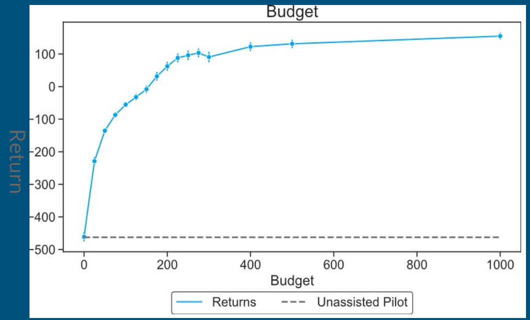


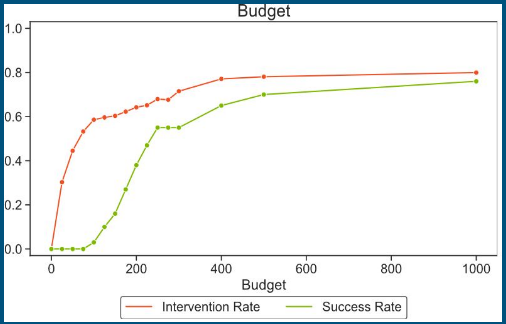
IR = 0.8

IR = 1



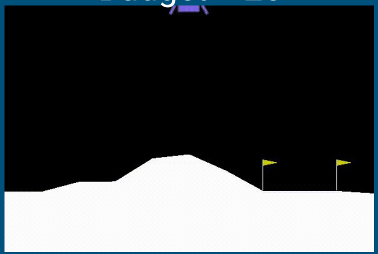
Budget Method





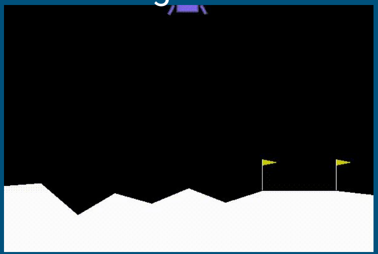
Sensor Pilot

Budget = 25

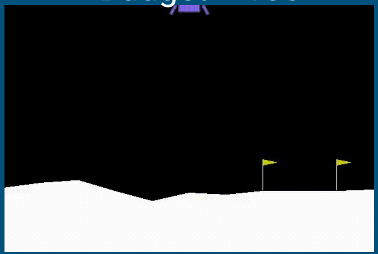


Budget = 50

Budget = 75



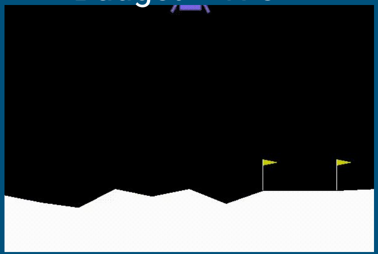
Budget = 100



Budget = 125

Budget = 150

Budget = 175



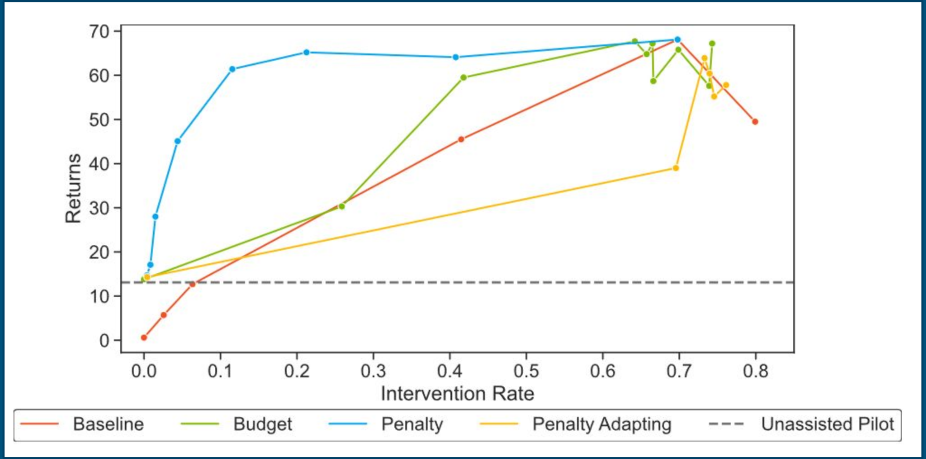
Super Mario

Noisy Pilot

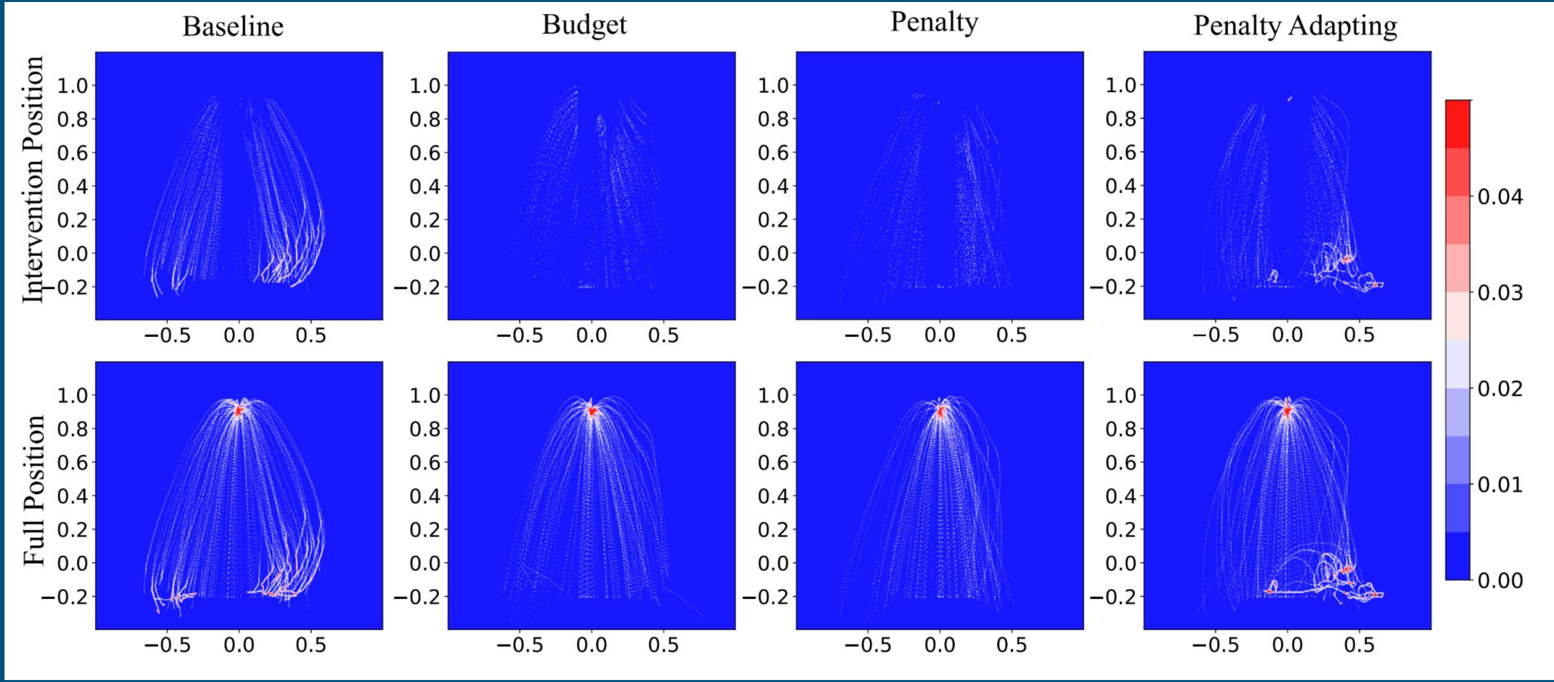
Penalty = 1

Budget = 300

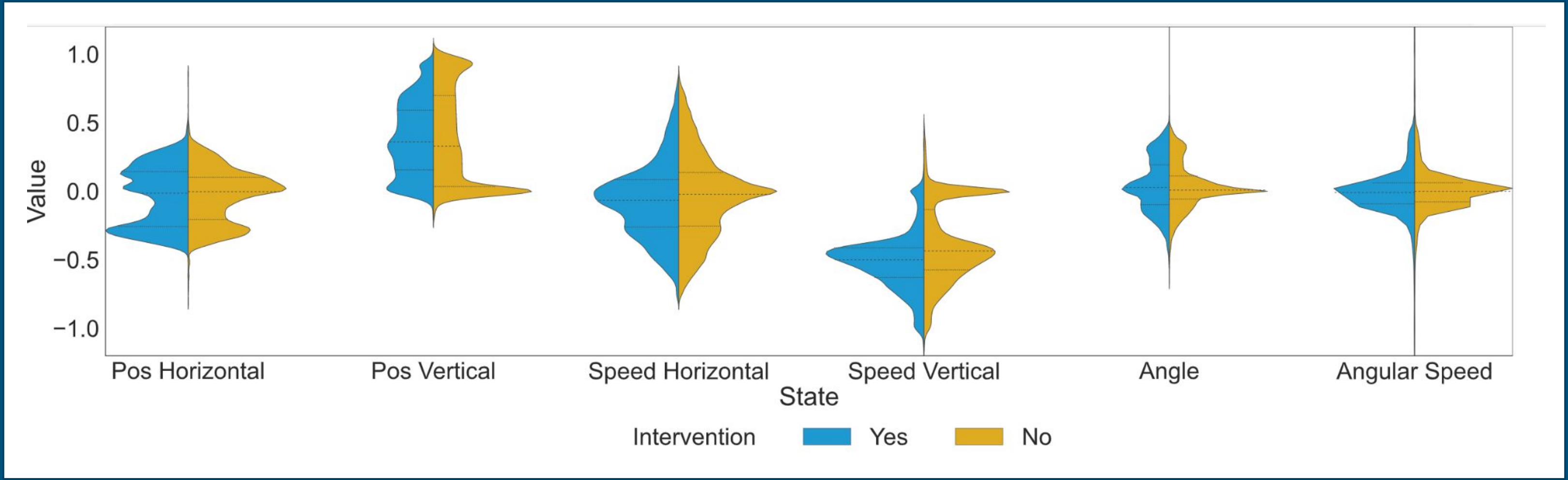
IR = 0.4



When Interventions Happen



When Interventions Happen

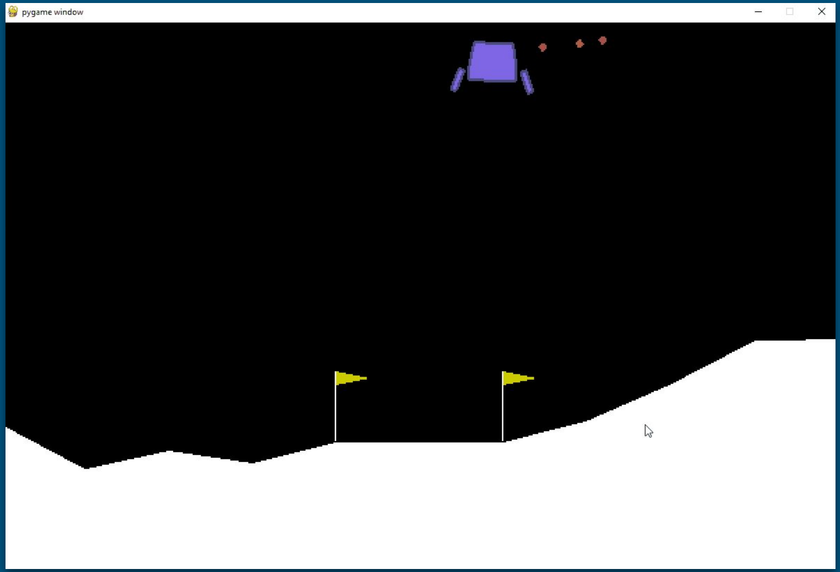


Distributions of each Lunar Lander state according to if an intervention happened, or not

- *Vertical position*, *vertical speed*, and *angle* show the agent is intervening more at states that are more likely to lead to imminent failure.

Demo

- Best way to know if the system is behaving as expected
- github.com/DavidKoleczek/hitl_demo



Red square means agent intervened

Next Steps & Challenges

Practical Application

What would it take to implement this in practice?

1. Access to a game environment
    a. API for internal state representation or video “pixel” output
    b. Programmatically take actions
2. Human data or simulated humans
    a. Simulated humans that transfer well to the actual distribution
    b. Dataset of $([s, a^h], a, r, s')$ tuples

Practical Application - Challenges

Open-ended games

- How do we know what a human wants to do?
- Do we have a reward function?
- What is the reward function telling us?

Practical Application

What would it take to implement this in practice?
Can we train a RL agent to perform well on a complicated game in the first place?

- Large amount of compute
- Difficult to get working

Thanks! Any Questions?

AAAI 2022

- Arxiv

Human AI @ ICML, 2021.

- pdf
- Poster

Project: https://github.com/DavidKoleczek/human_marl

Demo: https://github.com/DavidKoleczek/hitl_demo