# Deep Reinforcement Learning with Burn

Hao Xiang Li

# Motivation

#### Rust

- Safe (Statically typed, borrow checker)
- Performant
- Scalable
- Established ecosystem



#### Burn

- General Rust machine learning framework
- Backend agnostic
- Authors claim Rust allows for unique, safe optimisations



## Reinforcement Learning

- Jones, <u>Debugging RL</u>, <u>Without the</u> <u>Agonizing Pain</u>
- Community feedback: top post on

r/reinforcementlearning



- Henrik, Machine Learning and the Physical World, Lecture 8
  - RL is very hard meaning its as much an art as a science to get it to work
  - Don't do RL for your project!
  - Please don't

# RL is **hard**! Especially for **production**!

#### Reinforcement Learning



Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Sculley et al. Hidden Technical Debt in Machine Learning Systems

## Reinforcement Learning

- Tightly coupled stateful components
- Logic coupled with execution
- Feedback is poor with nonlocal errors
- Majority of libraries are not ready for production

Can Rust and Burn present a pathway to more effective reinforcement learning structures?



### Open Source Project

Rust Machine	Learning Eco	system			
Clustering	Data Processing	Data Structures	Decision Trees	GPU Computing	Linear Classifiers
Metaheuristics	MLOps	Neural Networks	Natural Language Processing	Reinforcement Learning	Scientific Computing

## What's next?

- This project aims to comprehensively evaluate the potential of using the Rust and Burn ecosystem for a reinforcement learning library.
- Existing implementation: Yun, https://github.com/yunjhongwu/burn-rl-examples/tree/master
- I will implement a set of reinforcement learning algorithms and parallelization strategies.

#### Goals

I speculate that this ecosystem will have advantages in

- Reduced overhead relative to Python
- Encourage stronger system design and safe code with the borrow checker, as well as a comprehensive set of tests.
- Traits system improving configuration management and modularity

Deep dive into Burn on a lower level, and how it can aid in RL development.

#### Evaluation

- Qualitative comparison of the Burn ecosystem against Python wrapper alternatives.
- Quantitative comparison of overhead and efficiency

# Thank you!