Advanced Topics in Reinforcement Learning

Lecture 27: RL in the Real World

Josiah Hanna
University of Wisconsin — Madison

Announcements

- Final projects due tomorrow at midnight.
- Check your grades on Canvas.
- Please complete the course evaluation! At 67% right now.
  - Due December 14 (tomorrow)!!
- Today:
  - Challenges in real world RL.
  - Ethical issues in real world RL.

Real world challenges

1. Using offline data.
2. Using simulators.
3. Learning on the real system.
4. High-dimensional states and actions.
5. Safety constraints.
6. Partial observable or non-stationary environments.
7. Under-specified and multi-objective rewards.
8. Explainability of decisions.
9. Real-time inference.
10. Large and/or unknown delays in sensors or actions.

Challenges of Real World Reinforcement Learning. Dulac-Arnold et al. 2019.

Sim2Real Reinforcement Learning

- Many potential application domains for RL already have high fidelity simulators available in them.
    - Examples: robotics, autonomous driving, flight simulators, inventory and demand models.
- **The reality gap:** RL + simulation ≠ high performing policies on real system.
- Possible solutions:
    - System identification or data-driven grounding.
    - Domain randomization.
    - Policy warm-starting.

See “Related work” section of “Grounded Action Transformations for Sim-toReal Reinforcement Learning” [Hanna et al. 2021] for a survey on sim2real.

Sim2Real Reinforcement Learning

Offline training from logs

- Assume we have logged $\mathcal{D} = \{(s_t, a_t, r_t, s_{t+1})\}$ from an existing system policy.
- This gives us an offline RL problem (last week's topic).
- Alternatively, use logged data to warm start RL:
  - First, perform behavior cloning / imitation learning $\pi_0 = \max_{\pi} \sum_{i=0}^{m} \log \pi(a_i \mid s_i)$.
  - Initialize RL with policy π_0.
- Benefit: immediately jump to an acceptable policy.
- Downside: might bias future RL away from more novel behaviors.

Multiple Objectives

- Challenge: applications may involve competing objectives.
  - Example: YouTube recommendations should maximize ad revenue and user satisfaction.
- One solution is to just linearize the objectives into a single objective:
  - $R_t = \sum_{i=1}^{k} w_i R_t^{(i)}$ where $R_t^{(i)}$ is the reward for objective i and w_i is a scalar weight.
- Alternatively, find a set of policies that give different trade-offs and let the end-user decide.
  - E.g., find set of Pareto-optimal policies.

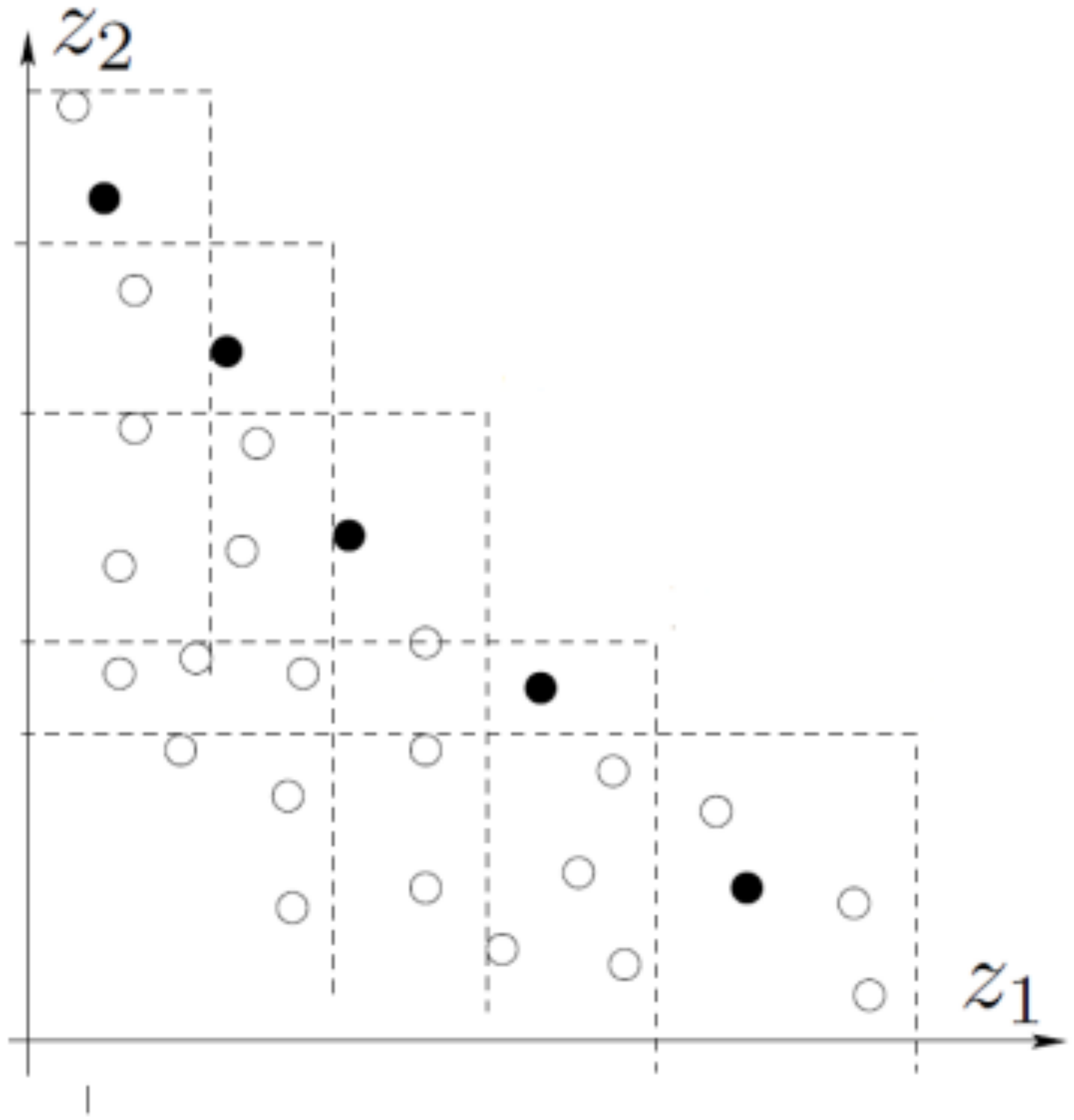


Image Credit: Approximation of Lorenz-Optimal Solutions in Multiobjective Markov Decision Processes. Perny et al. 2013.

Real-time decision-making

- Many control problems require decisions to be made with low latency.
  - Cache replacement in computer architecture requires nanosecond-latency but deep neural networks take milliseconds to output actions [1].
  - Many robotics control applications only allow 1-10 milliseconds per decision.
- Limits the complexity of policies we can learn.
- Limits the depth of search in MCTS (an anytime algorithm)
- Accept sub-optimality to be able to run in real-time.
- One solution: use anytime algorithms which can make a decision quickly but reach better decisions with more time.
  - MCTS-based approaches such as AlphaGo or RTMBA [2].

[1] Applying Deep Learning to the Cache Replacement Problem. Shi et al. 2019.

[2] RTMBA: A Real-Time Model-Based Reinforcement Learning Architecture for Robot Control. Hester et al. 2012.

Real-time Training

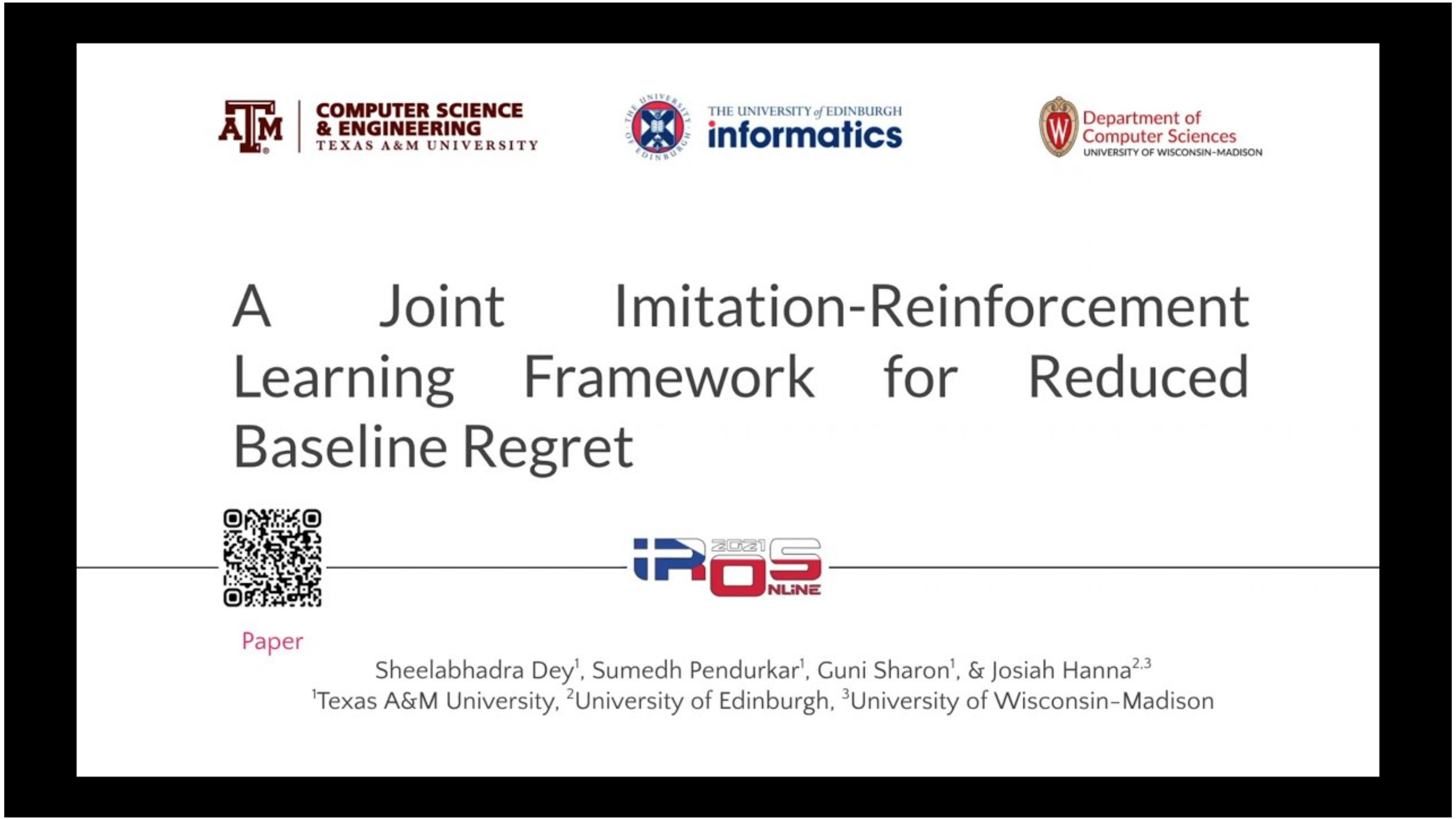
- The flip side: many control problems have high latency in state transitions and receiving rewards.
  - An automatic tutoring system that selects lessons to help students.
- Limits the amount of data that can be collected for learning.
- Makes credit-assignment challenging.
- Solutions:
  - Use simpler policy classes (e.g., linear) to reduce data requirements.

Safety

- Challenge: some actions may damage the controlled system or environment.
    - Example: a robot might break if it falls a lot or wear down over time; medical treatments might kill a patient.
- Possible solution: “watchdog” controller that takes over if the learning system take an unsafe action.
    - Example: Stabilizing controllers, shield functions [1].
- Various formulations for safety exist: constrained MDPs, budgeted MDPs (unknown constraint thresholds), safety layers; safe exploration.
    - Constraints on the final policy vs the learning process.

[1] Safe Reinforcement Learning via Shielding. Alshiekh et al. 2018.

Safety



COMPUTER SCIENCE & ENGINEERING
TEXAS A&M UNIVERSITY

THE UNIVERSITY *of* EDINBURGH
informatics

Department of Computer Sciences
UNIVERSITY OF WISCONSIN-MADISON

A Joint Imitation-Reinforcement Learning Framework for Reduced Baseline Regret

Paper

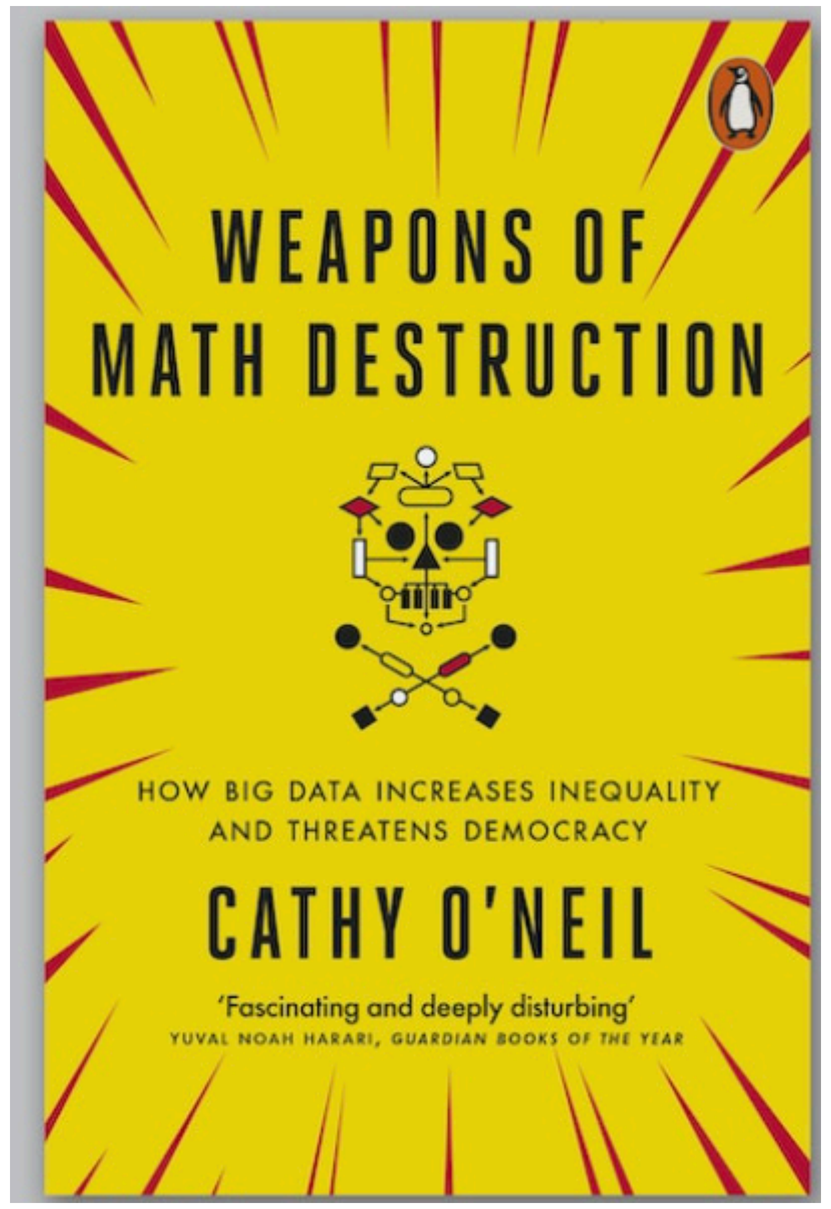
IROS 2021 ONLINE

Sheelabhadra Dey[1], Sumedh Pendurkar[1], Guni Sharon[1], & Josiah Hanna[2,3]
[1]Texas A&M University, [2]University of Edinburgh, [3]University of Wisconsin-Madison

A Joint Imitation-Reinforcement Learning Framework for Reduced Baseline Regret. Dey et al. 2021.

Ethical Issues

- Discrimination and bias in learned decision-making, particularly from offline data.
- Amplification of social injustices.
- Value alignment: does $\pi^\star$ behave as a person would for the same reward?
  - Example: paper-clip making robot; Facebook likes vs real enjoyment.
- How do actions affect people?
  - Example: social media addiction.


WEAPONS OF MATH DESTRUCTION
HOW BIG DATA INCREASES INEQUALITY AND THREATENS DEMOCRACY
CATHY O'NEIL
'Fascinating and deeply disturbing'
YUVAL NOAH HARARI, GUARDIAN BOOKS OF THE YEAR

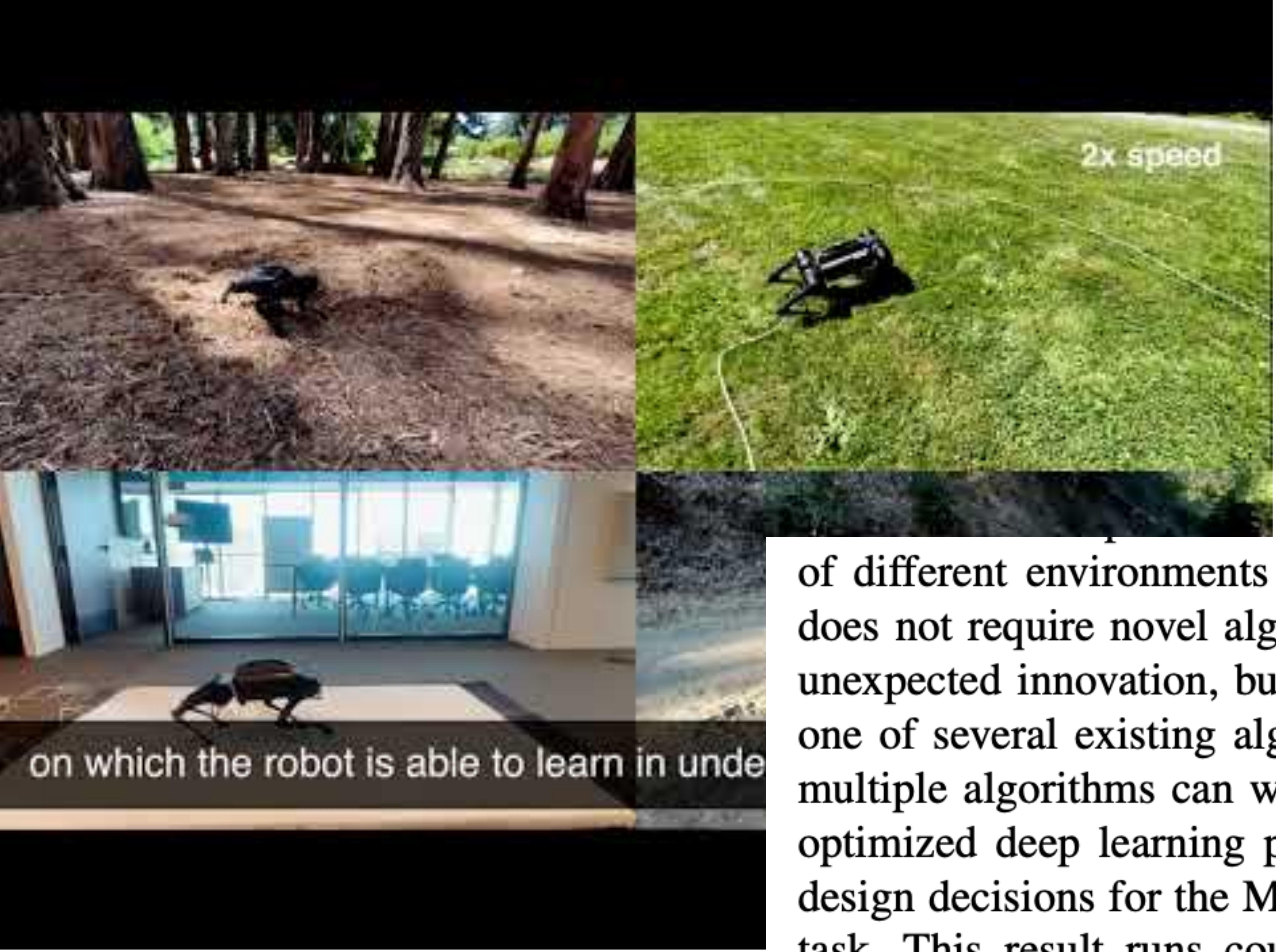
Recommended reading: “Weapons of Math Destruction.” Cathy O’Neil.

Hardik’s Presentation

- Slides

Summary

- Towards real world RL applications, we need:
    - Sample efficient and effective algorithms.
    - And also ways to address real world challenges.
- RL applications can harm people on the other end of the agent's actions.
    - As RL algorithm designers, we have a responsibility to avoid this harm.
- RL is one path towards complete autonomous agents that learn rich capabilities from experience.



of different environments and surface types. Crucially, this does not require novel algorithmic components or any other unexpected innovation, but rather careful implementation of one of several existing algorithmic frameworks (and indeed multiple algorithms can work well), combined with modern optimized deep learning packages and a number of careful design decisions for the MDP formulation of the locomotion task. This result runs counter to the principles articulated

A Walk in the Park: Learning to Walk in 20 Minutes With Model-Free Reinforcement Learning. Smith et al. 2022.

Thank you and good luck!