Reinforcement Learning: Introduction

Chen-Yu Wei

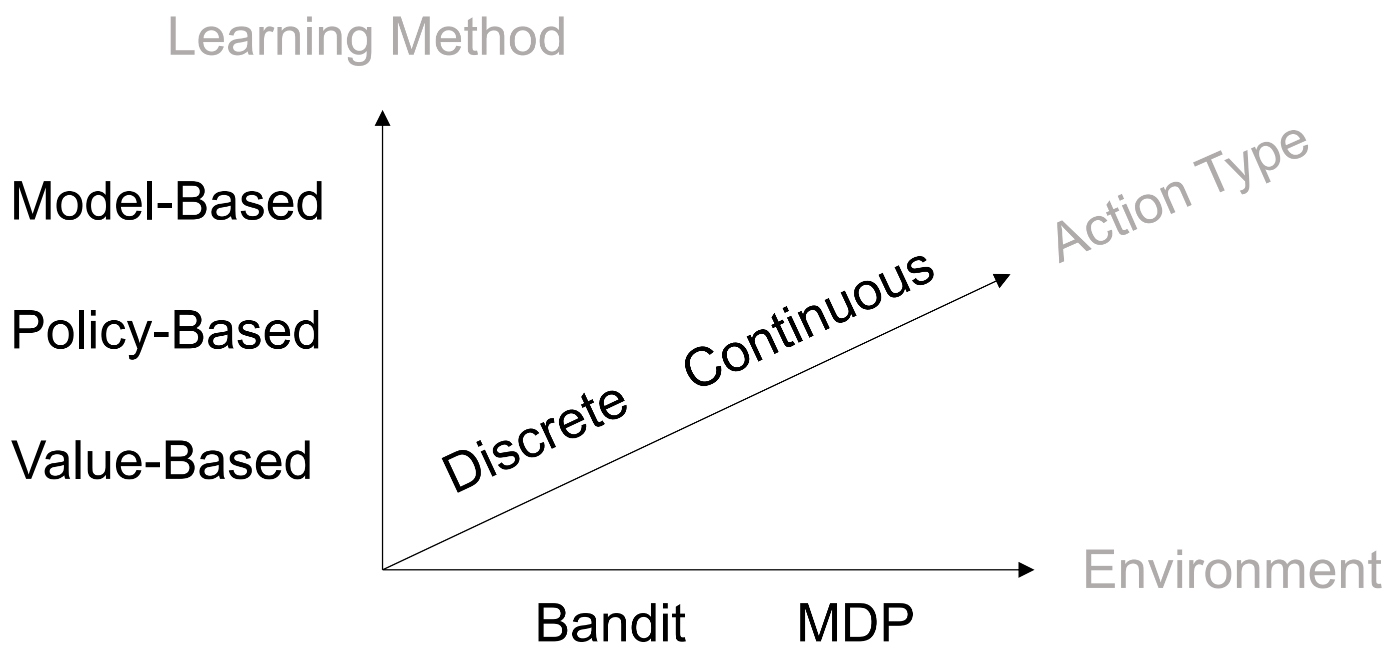
Platforms

- Course website: https://bahh723.github.io/rl2026sp/
  - Syllabus, announcement, slides, lecture recordings
  - Can be accessed from my personal website
- Gradescope (haven't created)
  - Homework submission
- Piazza (**just created today**)
  - Questions and discussions
- Canvas
  - We will **NOT** use canvas

Topics in This Course

- The **principles** behind **basic** RL algorithms
- The structure is similar to the previous semester (Fall 2025) (link)

Topics in This Course



Core Algorithms

Exploration in MDPs

Inference-Time Algorithms

Imitation Learning

Special Topics

Prerequisites

- Linear Algebra, Probability, Calculus, Machine Learning
- Python

Recommendation: Take Machine Learning first (or at the same time)

The RL course is unavoidably heavy in math. We use a lot of multi-dimensional calculus (e.g., gradient) and probability (e.g., unbiased estimation)

Resources

- Courses
  - UC Berkeley CS285
- Webpages
  - OpenAI SpinningUp
- Books
  - Sutton and Barto, Reinforcement Learning: An Introduction
- Implementations
  - OpenAI StableBaseline3
  - ShangtongZhang

Assignments (70%): 5-6 Problem Sets

- Programming tasks (using **PyTorch**)
  - Might need you to plot results or report numbers
  - Submission: Gradescope

Assignments (70%): 5-6 Problem Sets

- Late policy
  - 12 free late days distributed to all assignments as you like
  - No assignment can be submitted 7 days after its deadline
  - Each additional late day results in 10% deduction in the semester's assignment grade
  - Late day count is rounded up (1 hour late = 1 day late)

- Examples
  - HW1: **3** days late, HW2: **3** days late, HW3: **3** days late, HW4: **5** days late, HW5: **3** days late
    → HW grade *= 0.5
  - HW1: **8** days late, HW2: **6** days late, HW3: **3** days late, HW4: **4** days late, HW5: **1** day late
    → HW1 = 0 points **and** HW grade *= 0.8

Exams (30%)

- Midterm (12%)
  - February 26 (in class)
  - Everything covered before this point
- Final (18%)
  - May 1 (9AM-12PM)
  - Everything covered in the semester
- Exams are **open notes**
  - Your notes, printed slides are allowed
  - Books, electronic devices are not allowed

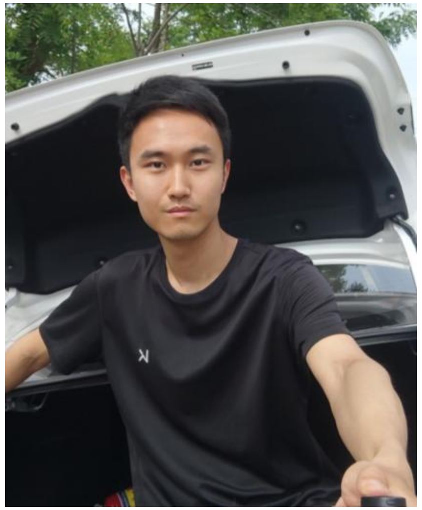
Exams (30%)

- **All exams are in person**. No online option is available.
- For both the midterm and the final, one (and at most one) make-up exam session may be arranged within one week.
- If you miss the midterm due to extenuating circumstances
  - E.g., illness, family emergency
  - You may use the final exam to replace the midterm score
- If you miss the final due to extenuating circumstances
  - You may request incomplete grade and complete the exam after the semester.

TAs

Xinyu Liu

Fengyu Gao



Yufeng Gao

They will grade the assignments and hold a one-hour office hour per week (each).

This starts from the next week. The time will be announced on the website.

Short Survey

- Do you know how to code in **Python**?
- Have you used **PyTorch** before?
- Have you taken **probability** course?
- Have you taken **machine learning** course before?
- Do you know **gradient descent**?

Let the Machine Learn To Make Decisions from Interactions

(Trial and Error)

Games

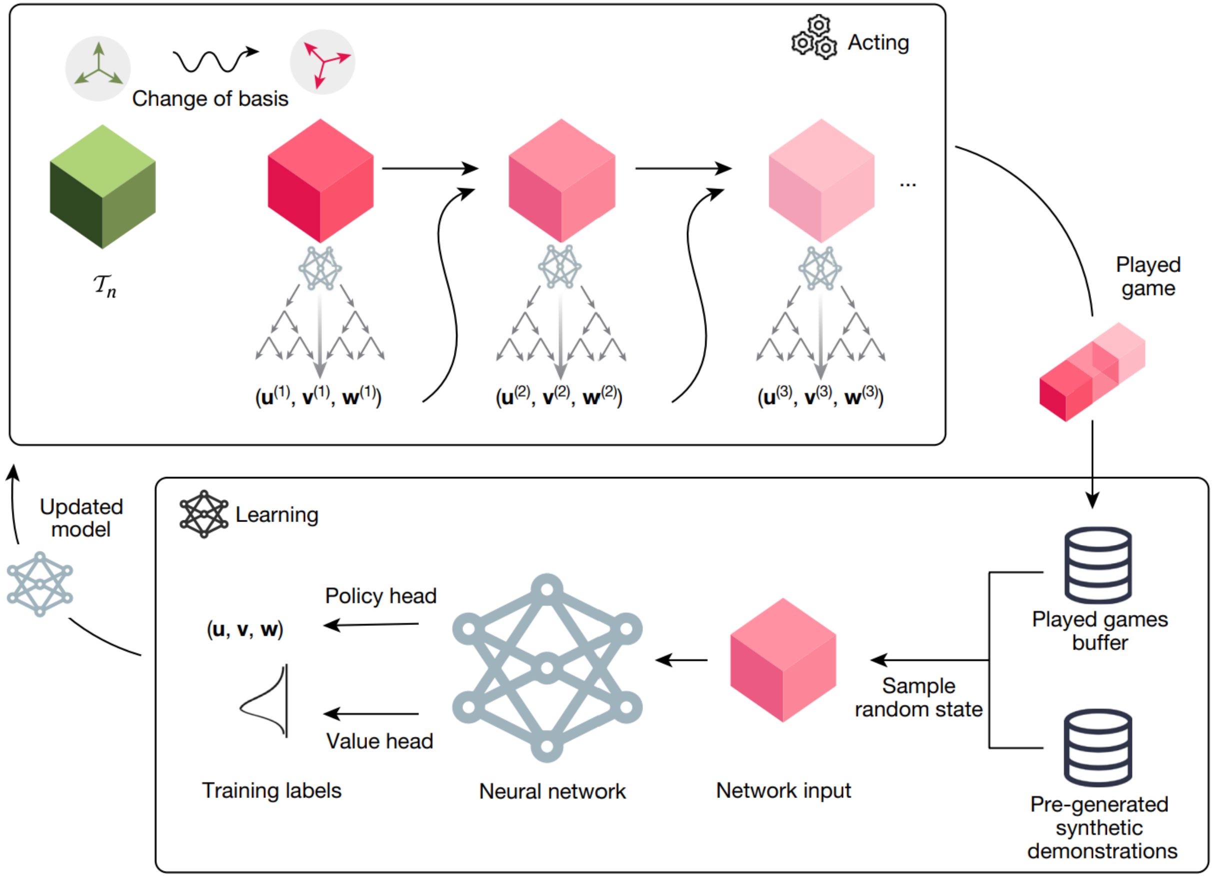
10 mins training

120 mins

240 mins

Mnih et al., Playing Atari with Deep Reinforcement Learning, 2015

Algorithm Discovery (faster matrix multiplication)



Size (*n*, *m*, *p*)	Best method known	Best rank known	AlphaTensor rank Modular	Standard
(2, 2, 2)	(Strassen, 1969)[2]	7	7	7
(3, 3, 3)	(Laderman, 1976)[15]	23	23	23
(4, 4, 4)	(Strassen, 1969)[2] (2, 2, 2) ⊗ (2, 2, 2)	49	47	49
(5, 5, 5)	(3, 5, 5) + (2, 5, 5)	98	96	98
(2, 2, 3)	(2, 2, 2) + (2, 2, 1)	11	11	11
(2, 2, 4)	(2, 2, 2) + (2, 2, 2)	14	14	14
(2, 2, 5)	(2, 2, 2) + (2, 2, 3)	18	18	18
(2, 3, 3)	(Hopcroft and Kerr, 1971)[16]	15	15	15
(2, 3, 4)	(Hopcroft and Kerr, 1971)[16]	20	20	20
(2, 3, 5)	(Hopcroft and Kerr, 1971)[16]	25	25	25
(2, 4, 4)	(Hopcroft and Kerr, 1971)[16]	26	26	26
(2, 4, 5)	(Hopcroft and Kerr, 1971)[16]	33	33	33
(2, 5, 5)	(Hopcroft and Kerr, 1971)[16]	40	40	40
(3, 3, 4)	(Smirnov, 2013)[18]	29	29	29
(3, 3, 5)	(Smirnov, 2013)[18]	36	36	36
(3, 4, 4)	(Smirnov, 2013)[18]	38	38	38
(3, 4, 5)	(Smirnov, 2013)[18]	48	47	47
(3, 5, 5)	(Sedoglavic and Smirnov, 2021)[19]	58	58	58
(4, 4, 5)	(4, 4, 2) + (4, 4, 3)	64	63	63
(4, 5, 5)	(2, 5, 5) ⊗ (2, 1, 1)	80	76	76

Deepmind, “Discovering faster matrix multiplication algorithms with reinforcement learning”, 2022

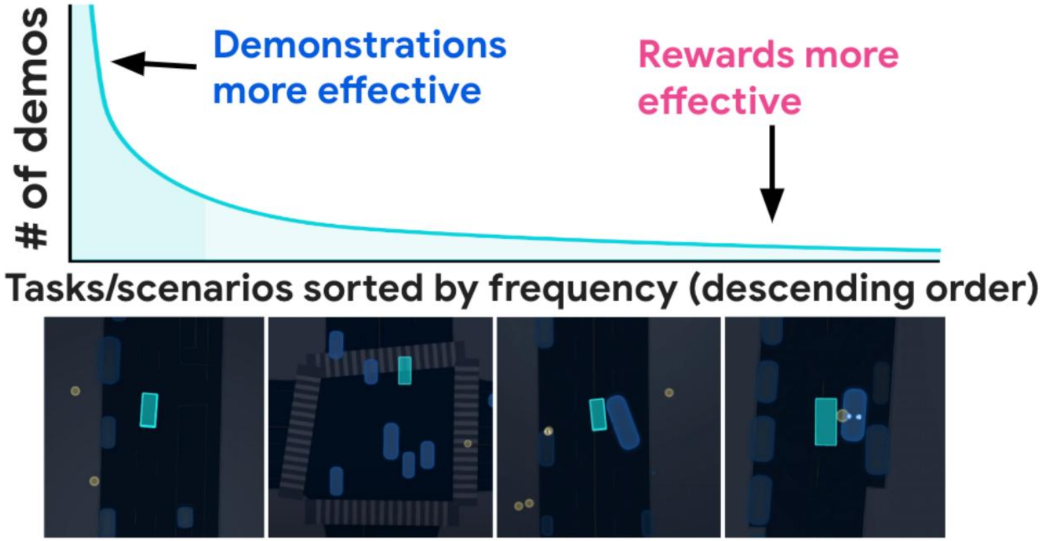
Autonomous Driving

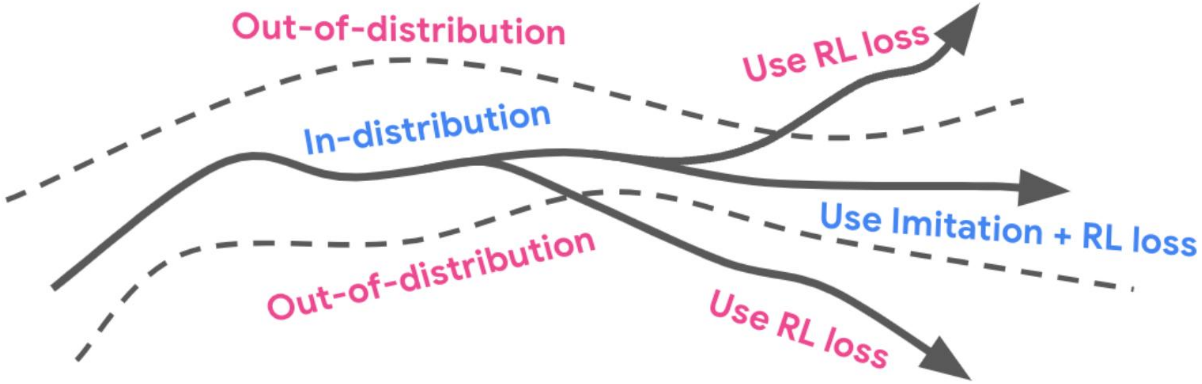
RL in simulators

Self-driving on the road

Amini et al., “VISTA 2.0: An Open, Data-driven Simulator for Multimodal Sensing and Policy Learning for Autonomous Vehicles”, 2021

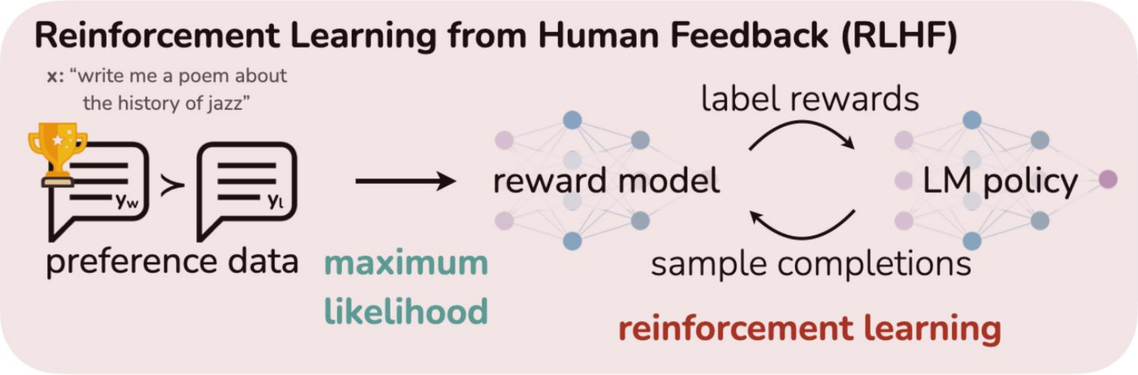
Autonomous Driving



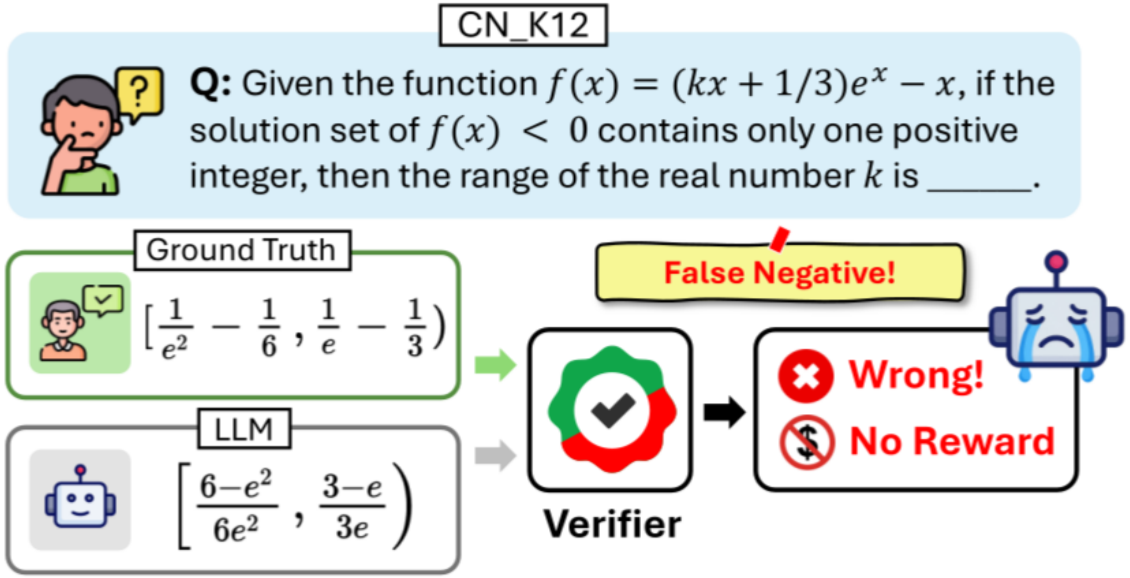


Lu et al., “Imitation Is Not Enough: Robustifying Imitation with Reinforcement Learning for Challenging Driving Scenarios”, 2022

Training Large Language Models



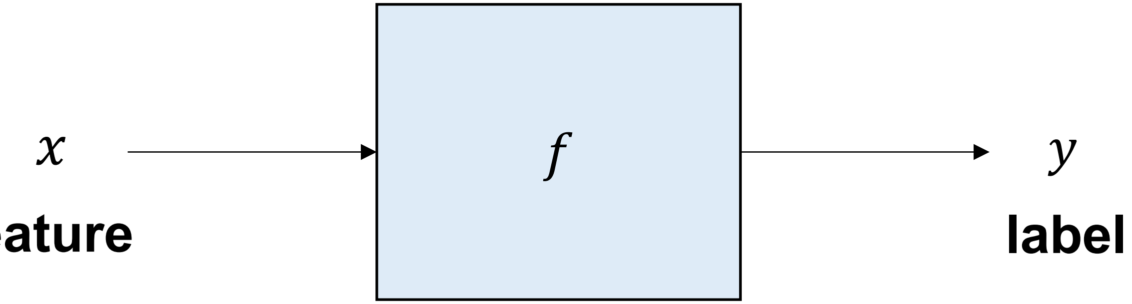
Rafailov et al., "Direct Preference Optimization: Your Language Model is Secretly a Reward Model", 2023



Xu et al. "TinyV: Reducing False Negatives in Verification Improves RL for LLM Reasoning", 2025

Closer Look at Reinforcement Learning

Supervised Learning

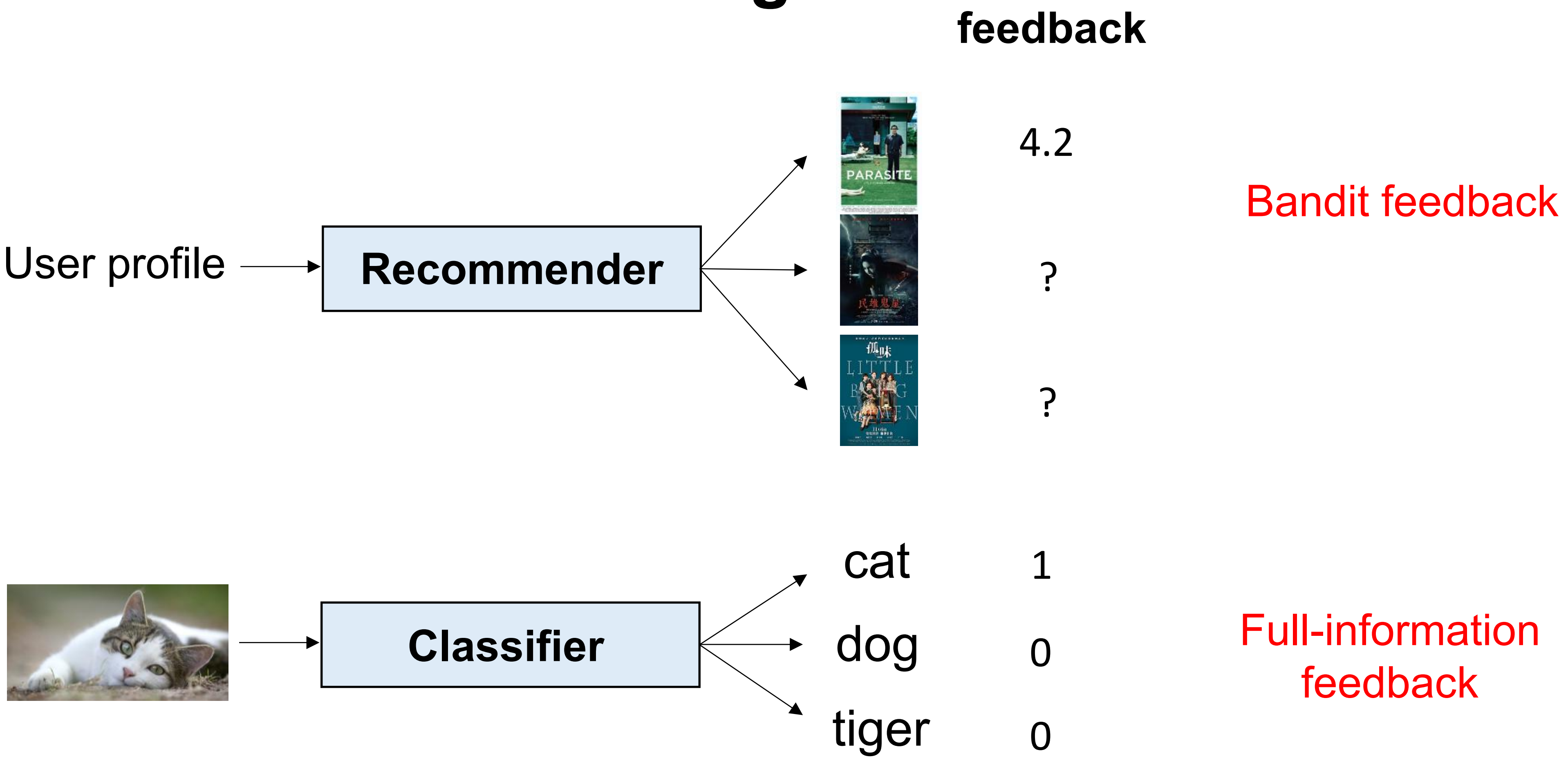


f () = Cat

f (temperature, humidity,...) = 1000mm precipitation

Given a lot of (x, y) pairs, find an f that such that $f(x) \approx y$

Reinforcement Learning



RL usually deals with bandit feedback

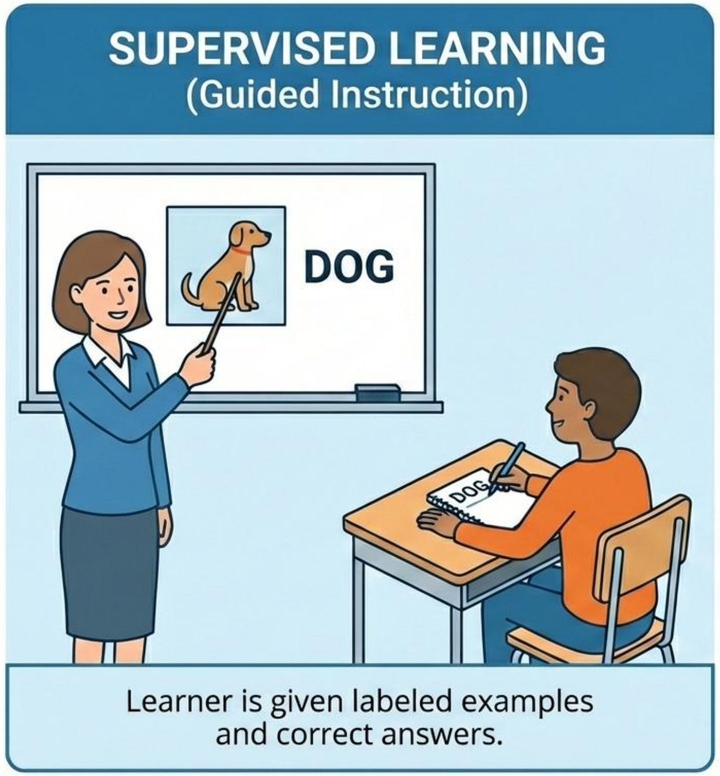
Bandit Feedback

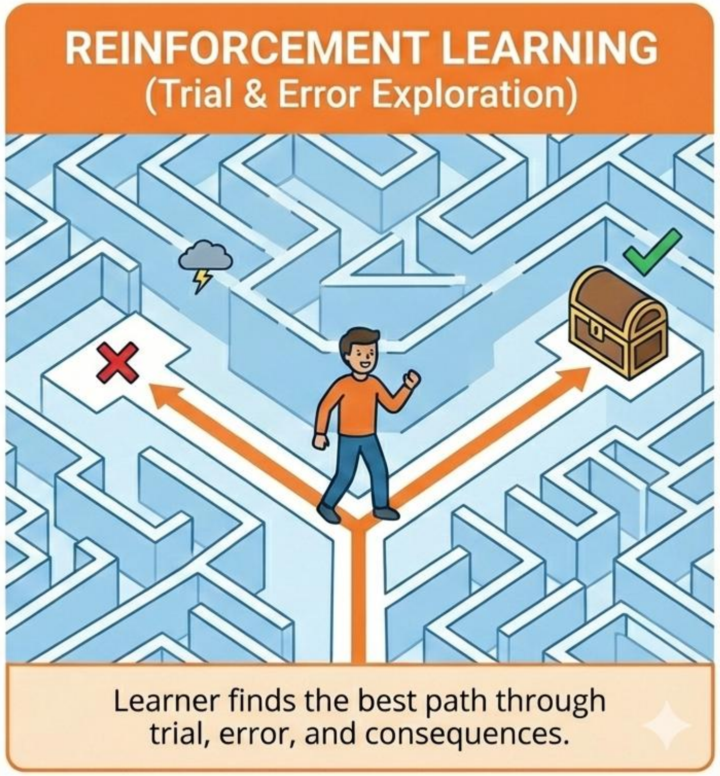
- Needs **exploration** (trial and error)

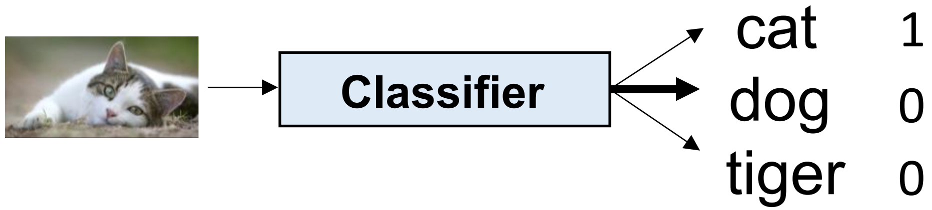
Bandit Feedback

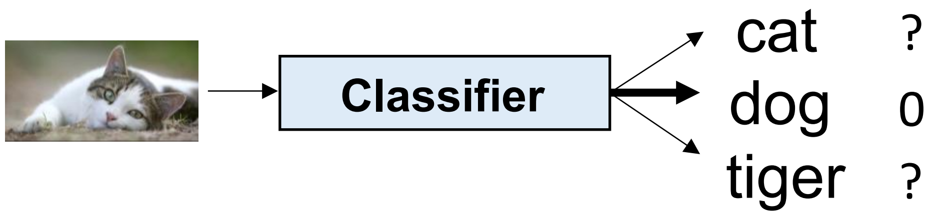
- Learning from reward feedback → Learning from **bandit** reward feedback
- SL and RL differs in the way of training, not the modeling
- E.g., Bandit classification

Reinforcement Learning







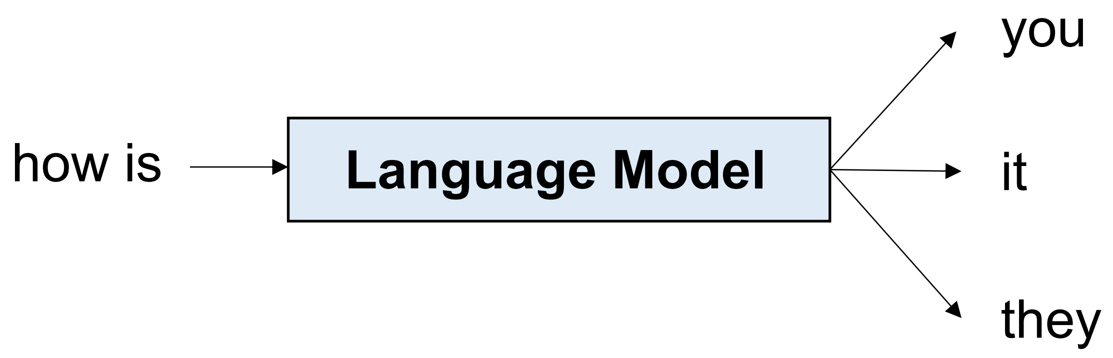


Reinforcement Learning – Challenge 1

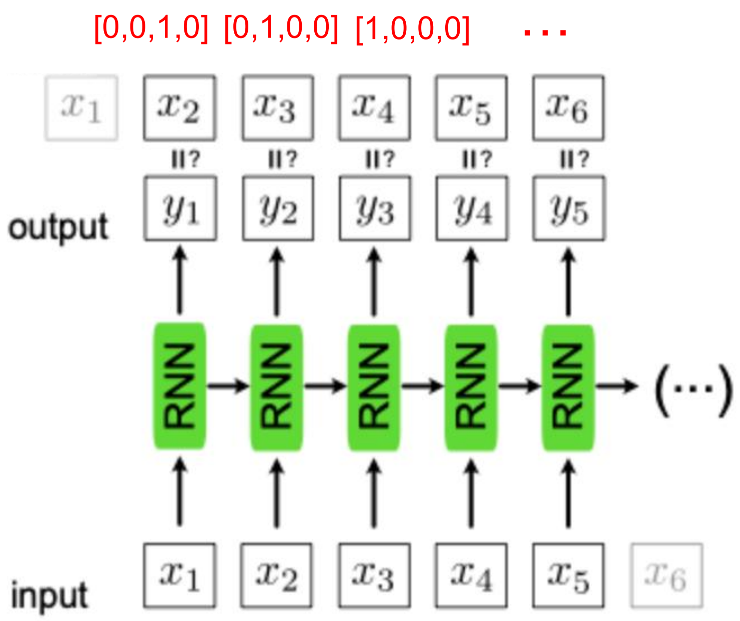
- Challenge: Bandit feedback
- Strategy: **Exploration**

RL in Sequential Decision Making

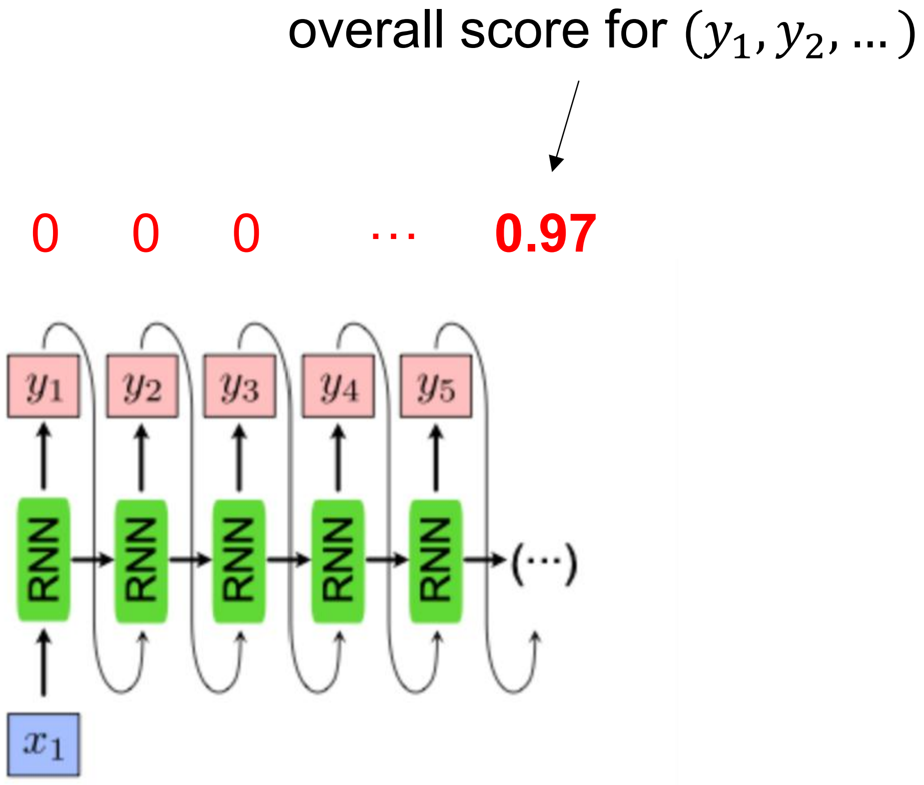
Often, a task is accomplished by a **sequence of action**, e.g., language.



RL in Sequential Decision Making



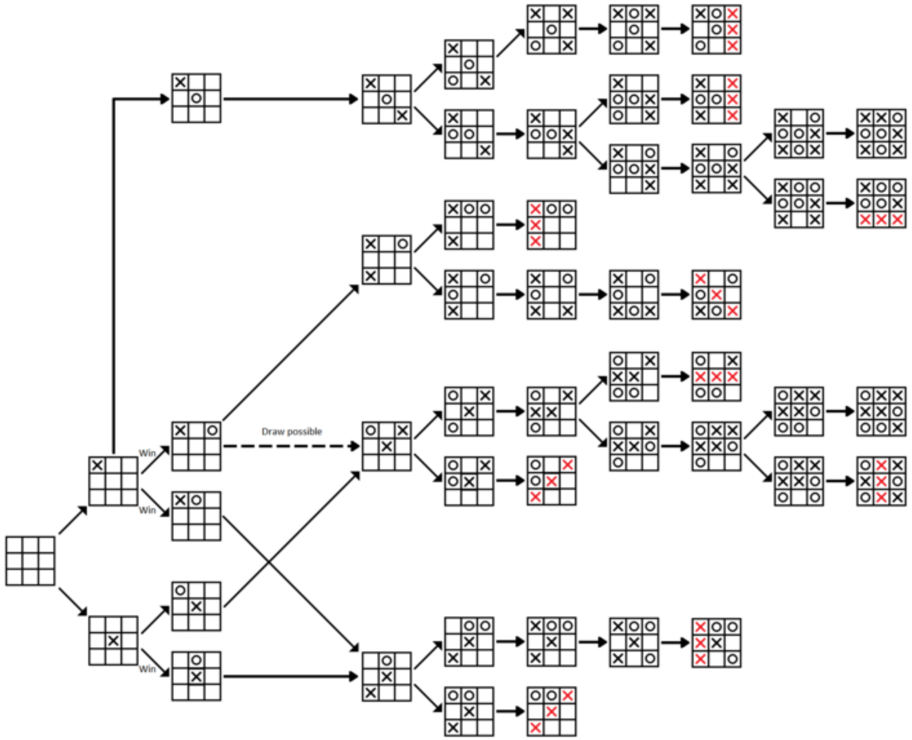
(Machine Learning for Scientists)



Bandit + **Delayed and Aggregated** Feedback

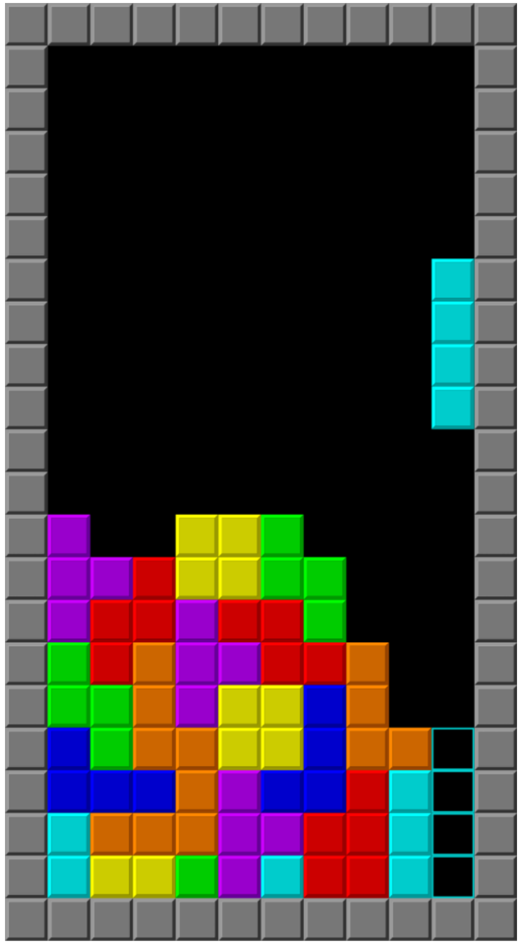
Delayed and Aggregated Feedback

- Need for **(temporal) credit assignment**



Delayed and Aggregated Feedback

- Need for **(temporal) credit assignment**



RL in Sequential Decision Making

Learning sequential decision making

→ Learning sequential decision making **with bandit and delayed feedback**

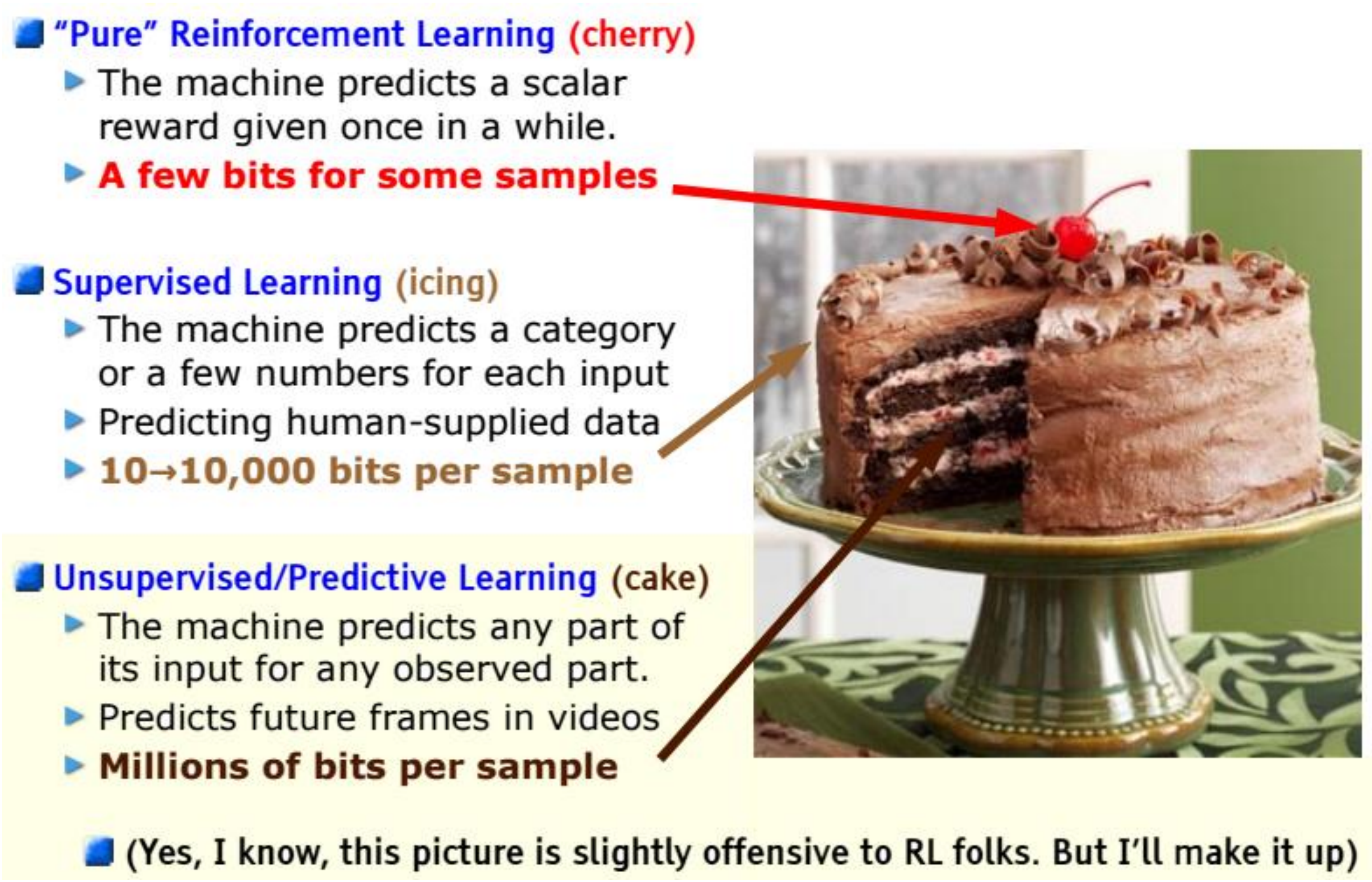
SL: "what to do in each step" (full-information, immediate)

RL: "how you're doing overall" (bandit, delayed)

Reinforcement Learning – Challenge 2

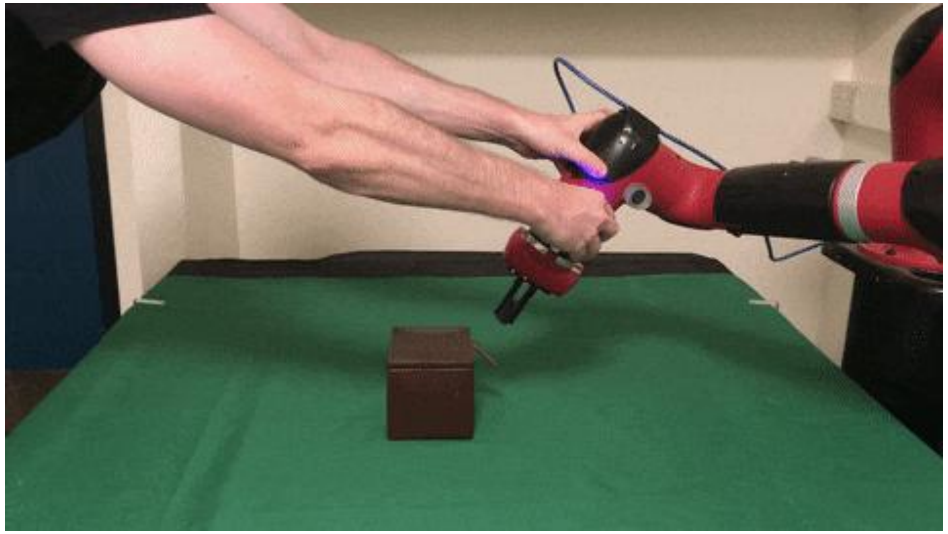
- Challenge: Delayed and aggregated feedback
- Strategy: **Credit assignment**

RL Uses Much Sparser Signal than SL

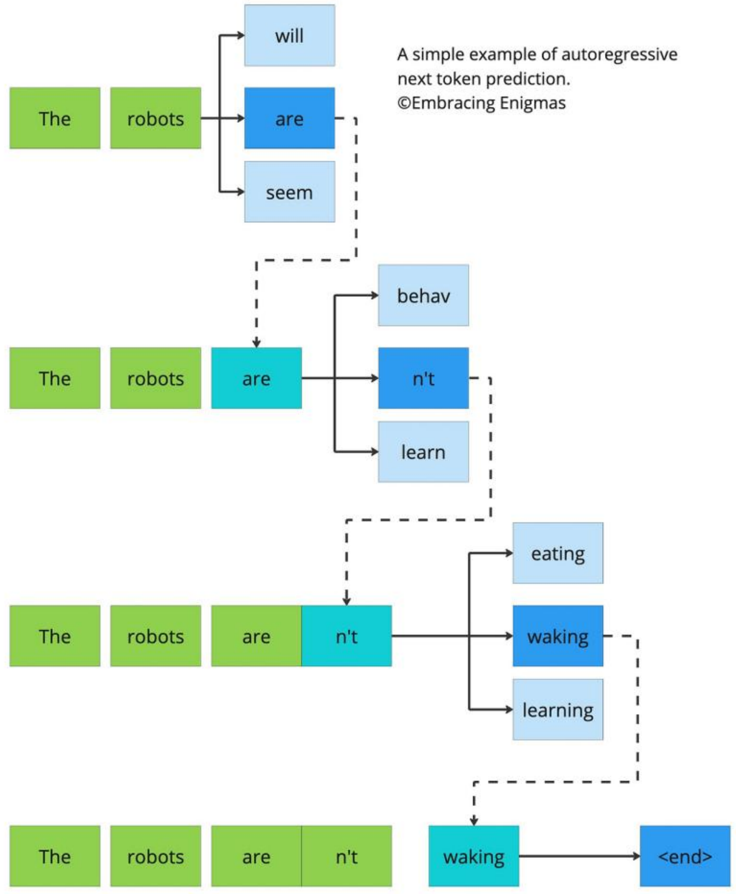


(Yann LeCun, 2016 NIPS)

Imitation Learning ∈ Supervised Learning



Robot manipulation



Next token prediction

When Is IL (SL) Insufficient?

- The truly best policy is unknown / expert is imperfect
  - Atari game, Go
  - Faster matrix multiplication

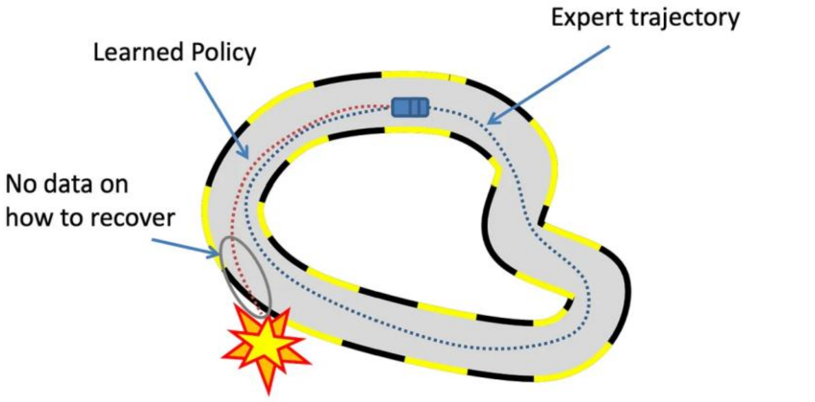
  ⇒ RL can **search** for better solutions

- RL signal may more faithfully reflect our real objective
  - RL from Human Feedback

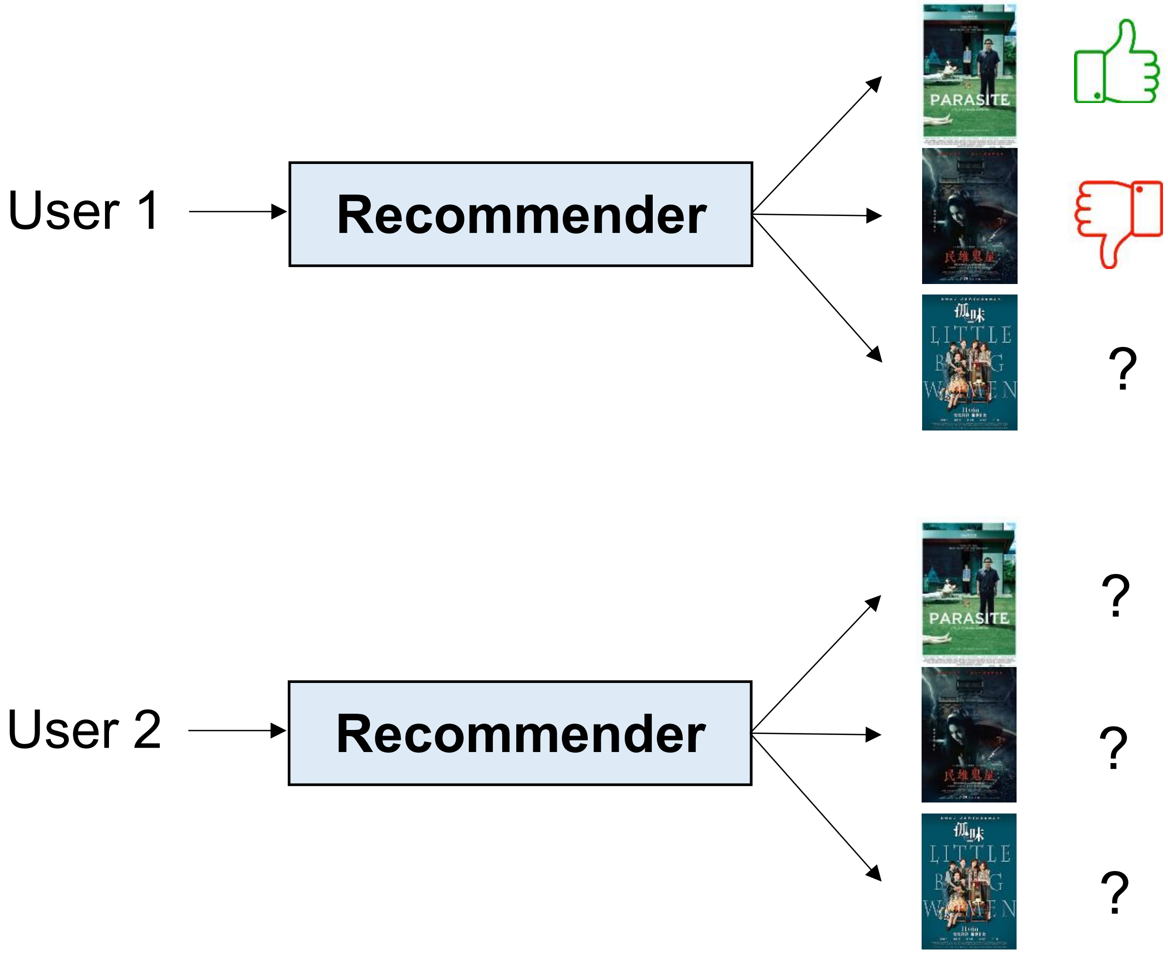
  ⇒ RL can provide **alignment** to the real objective

- The expert data has limited coverage
  - Autonomous driving

  ⇒ RL can explore edge cases and **robustify** solutions



Learning in Diverse Environments



Summary: Challenges in RL

Core Challenges in RL

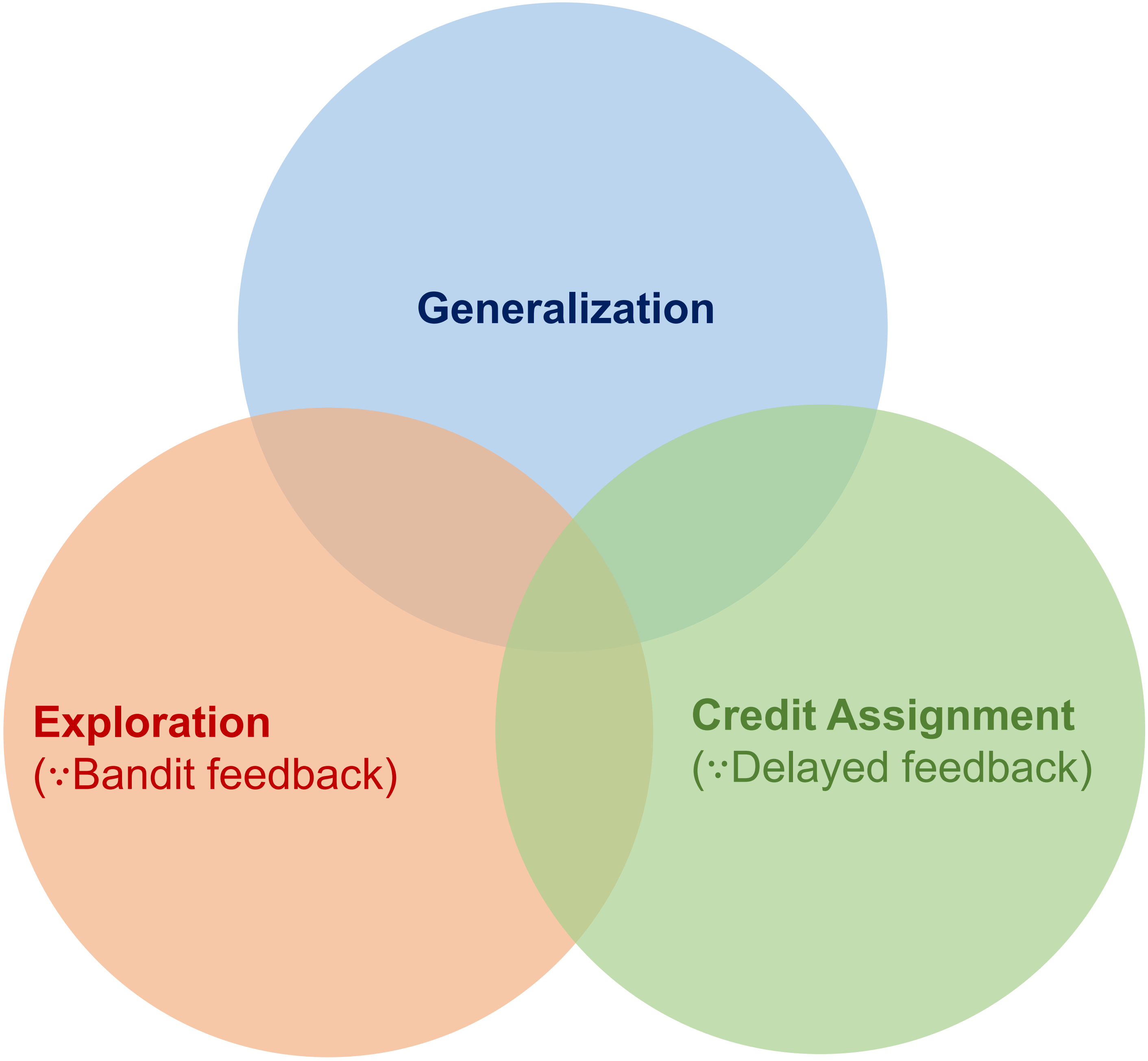
- Bandit feedback
  - Need exploration

- Delayed and aggregated feedback
  - Need credit assignment

- Unseen input / untried decisions (also a challenge in SL)
  - Need generalization

Other Challenges

- Reward design
- Simulation-to-reality gap
- Safety, robustness under attacks, … (similar challenges are also in SL)


Generalization
Exploration
(∵Bandit feedback)
Credit Assignment
(∵Delayed feedback)