Introduction to the Course

Chen-Yu Wei

Learning To Make Decisions from Interactions

Games

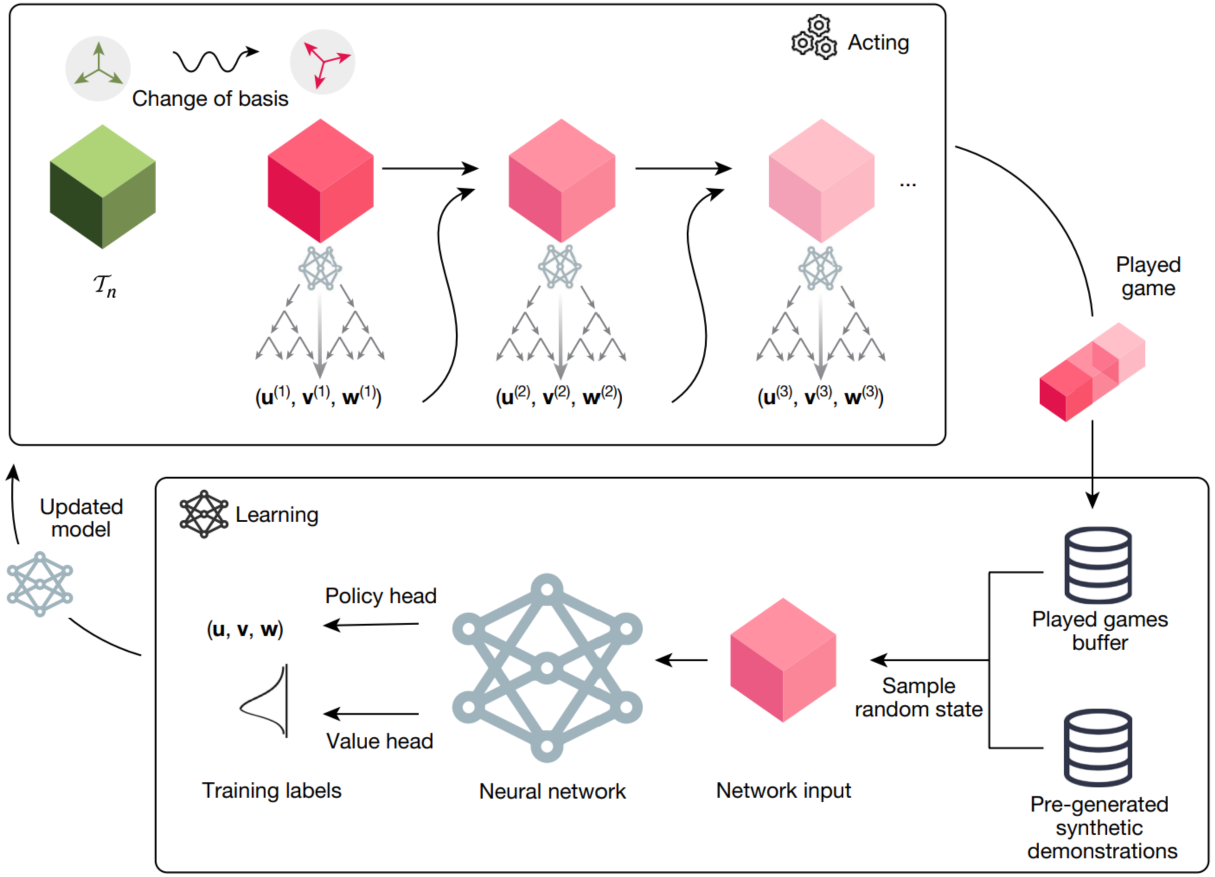
10 mins training

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	AlphaTensor rank 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

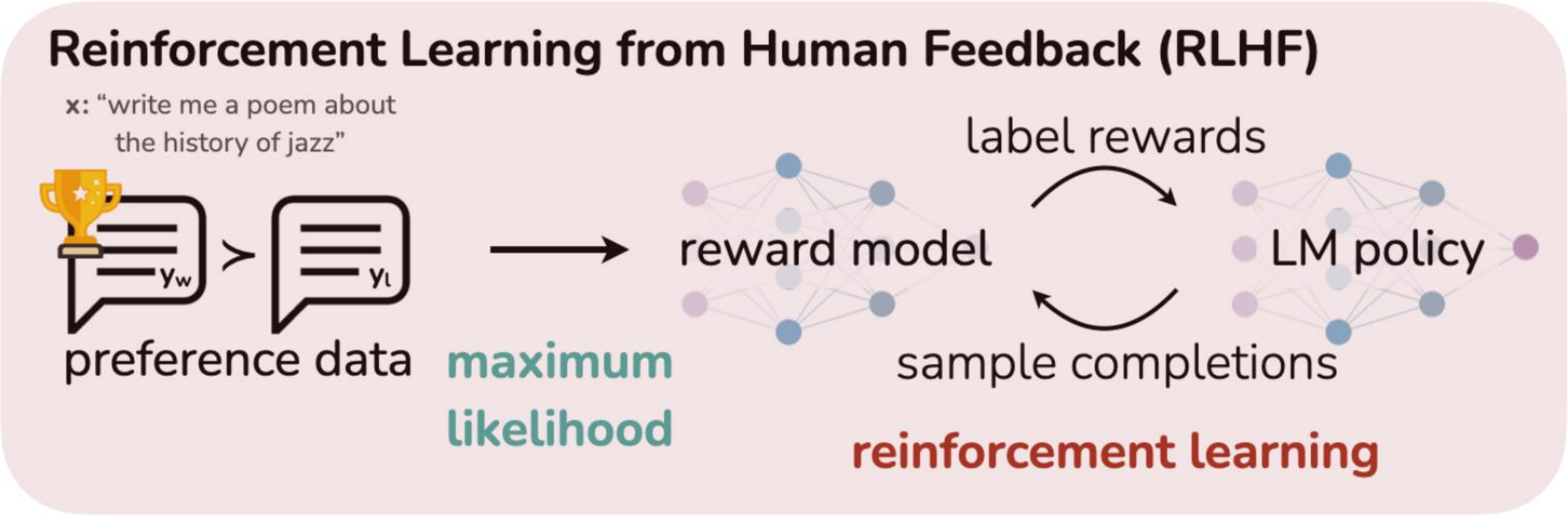
Autonomous Driving

RL in simulators

Safe 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

Languages



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

Closer Look at Reinforcement Learning

Supervised Learning

x → f → y

feature **label**

f () $=$ Cat

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

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

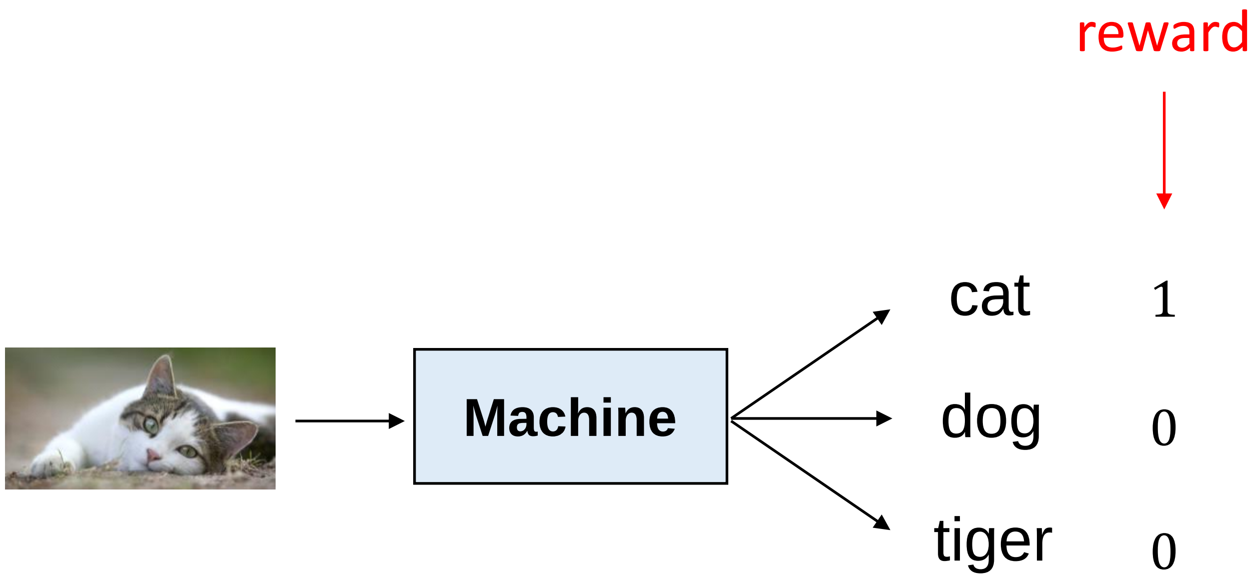
Reinforcement Learning

- Reinforce?

- Reinforce?

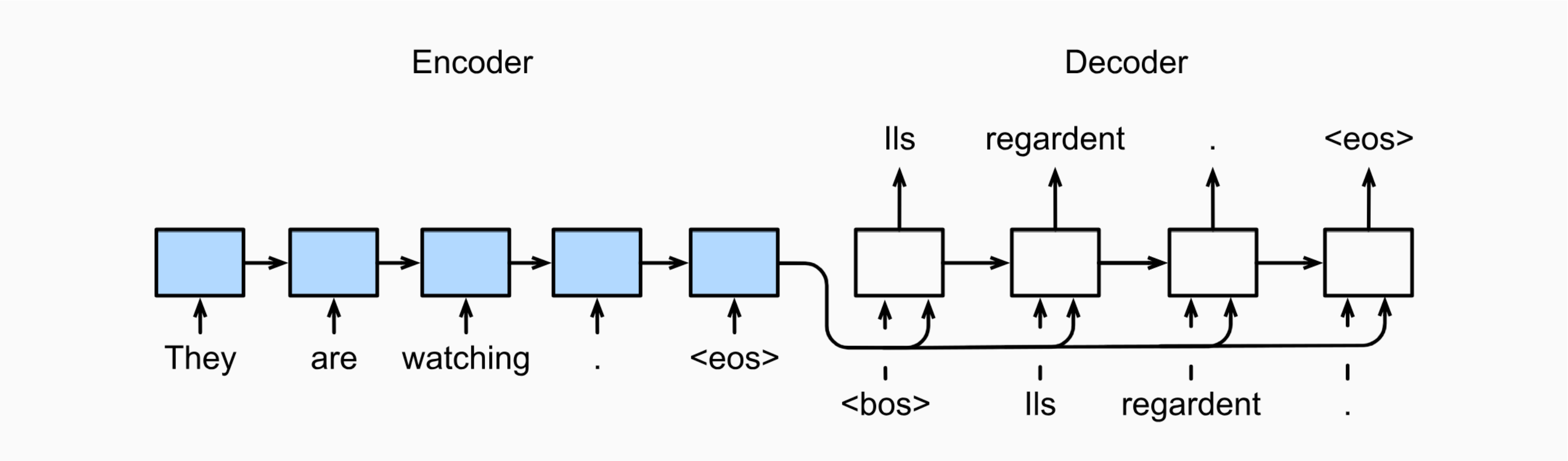
Reinforcement Learning

- Learning from reward feedback?



Reinforcement Learning

- Learning sequential decision making?



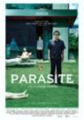
“Dive into Deep Learning”

Reinforcement Learning

feedback **collected data**

User profile → **Recommender**

4.2
?
?

(User,  , 4.2) Bandit feedback

 → **Classifier**

cat 1 ( , cat, 1)

dog 0 ( , dog, 0)

tiger 0 ( , tiger, 0)

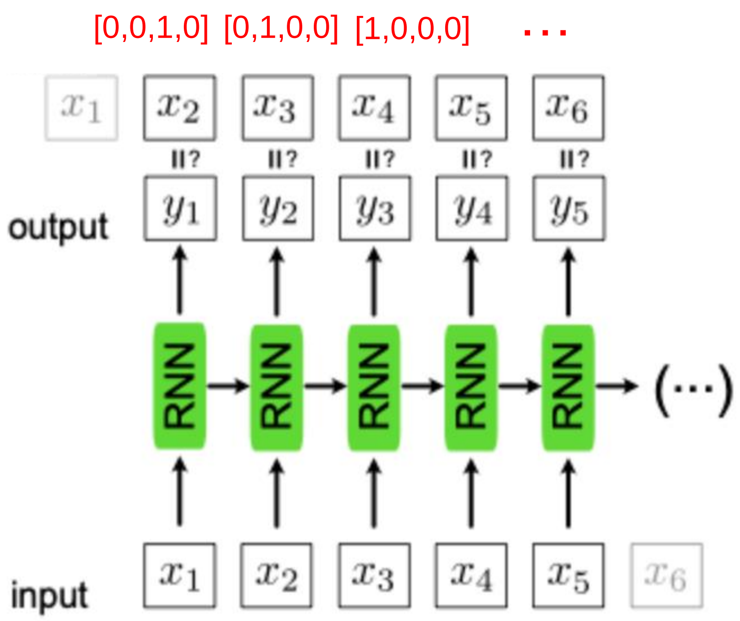
Full-information feedback

RL usually deals with bandit feedback

Bandit Feedback

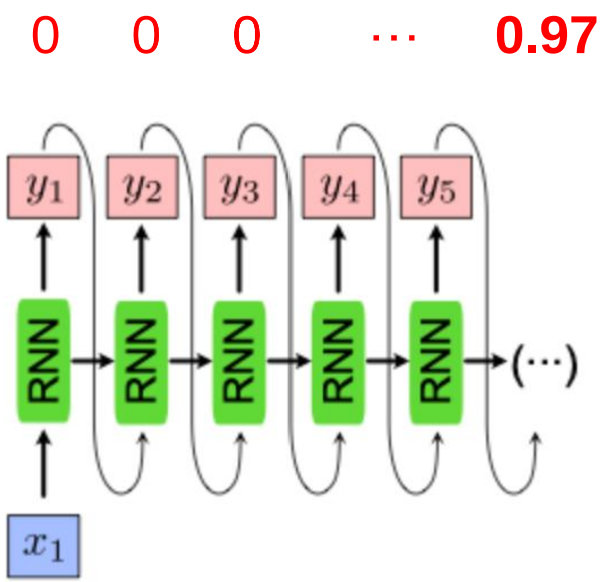
- Needs **exploration**

RL in Sequential Decision Making



(Machine Learning for Scientists)

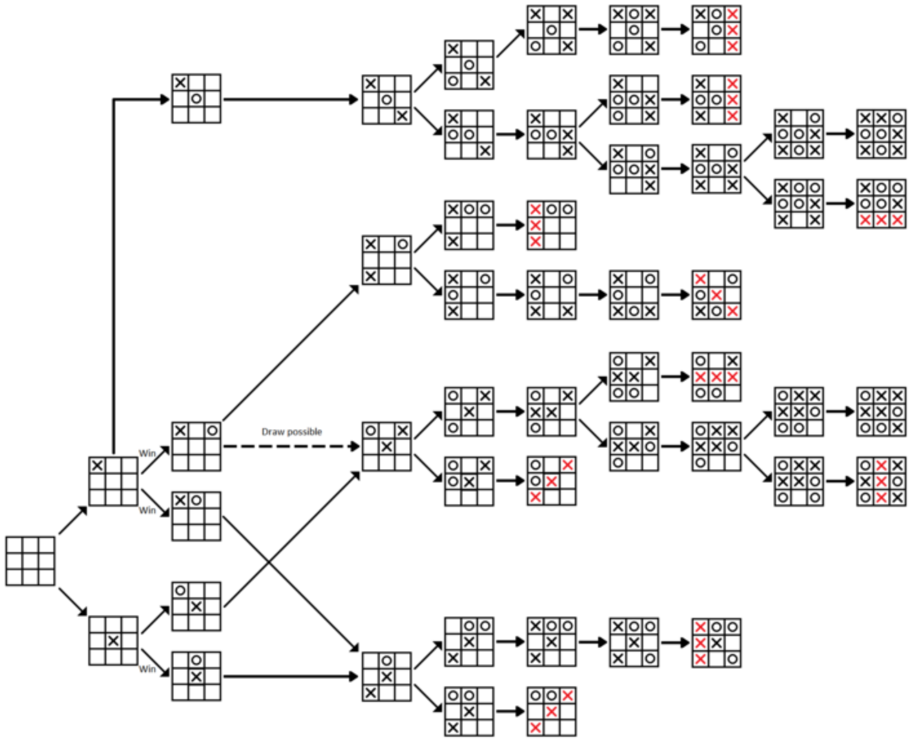
overall score for $(y_1, y_2, \dots)$



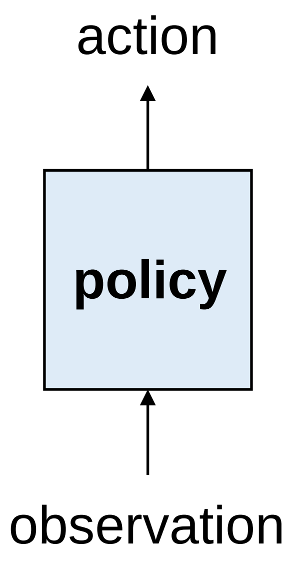
Bandit + **Delayed and Aggregated** Feedback

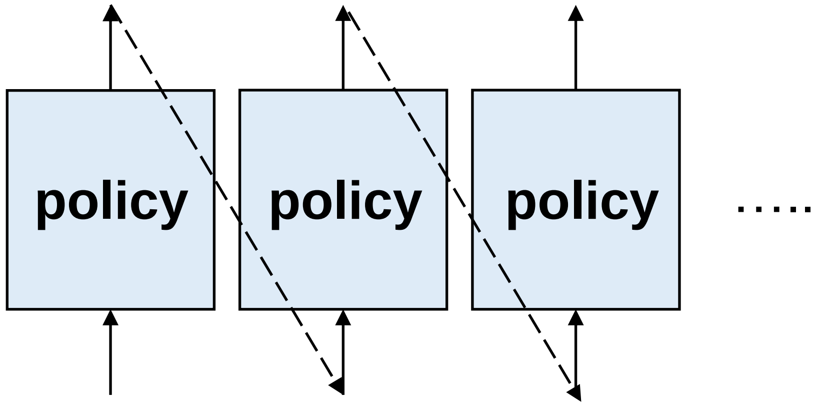
Delayed and Aggregated Feedback

- Need for **credit assignment**



RL vs SL

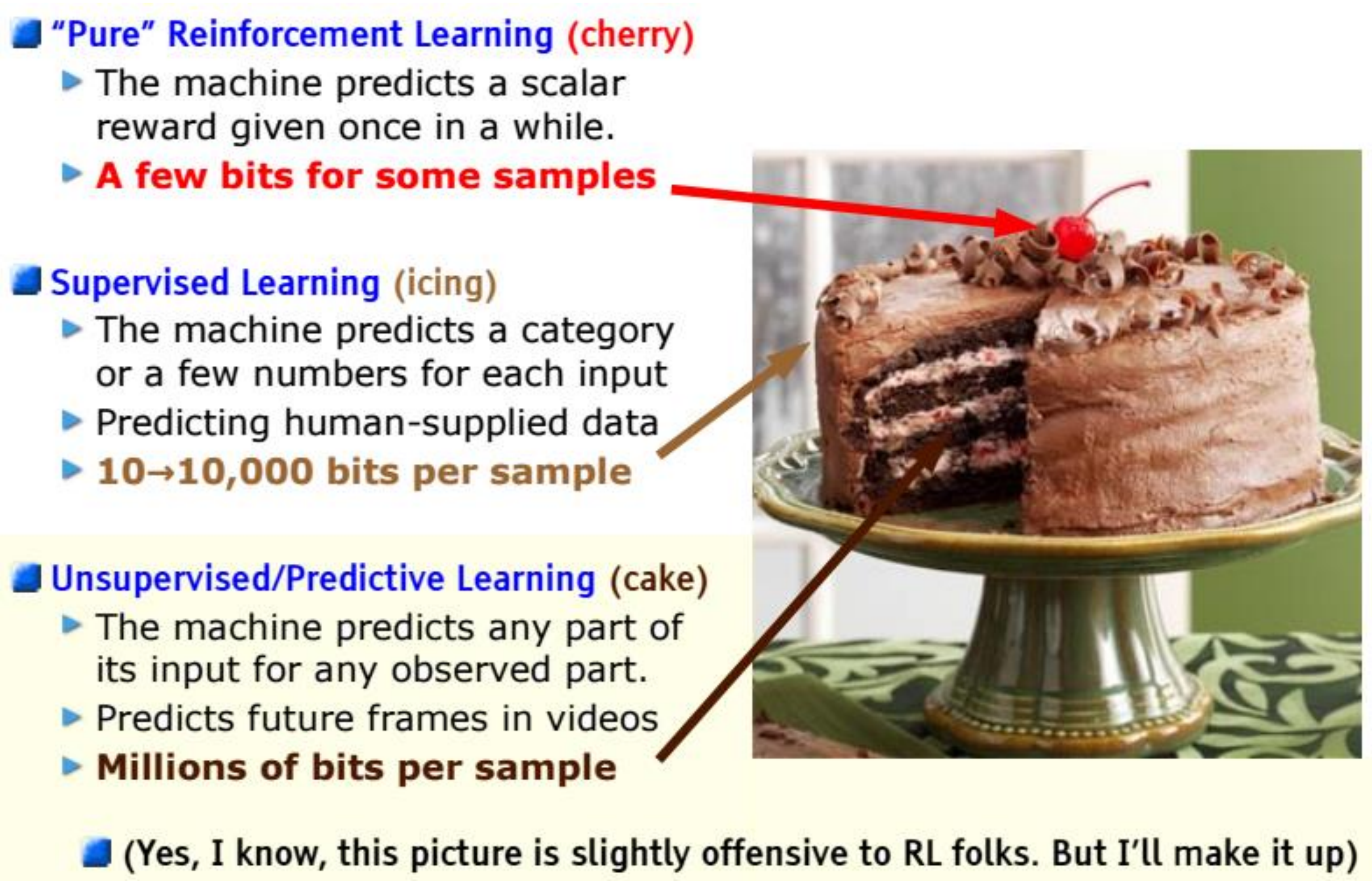




SL feedback: "what to do in each step" (full-information, immediate)

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

RL Signal Can Be Very Sparse



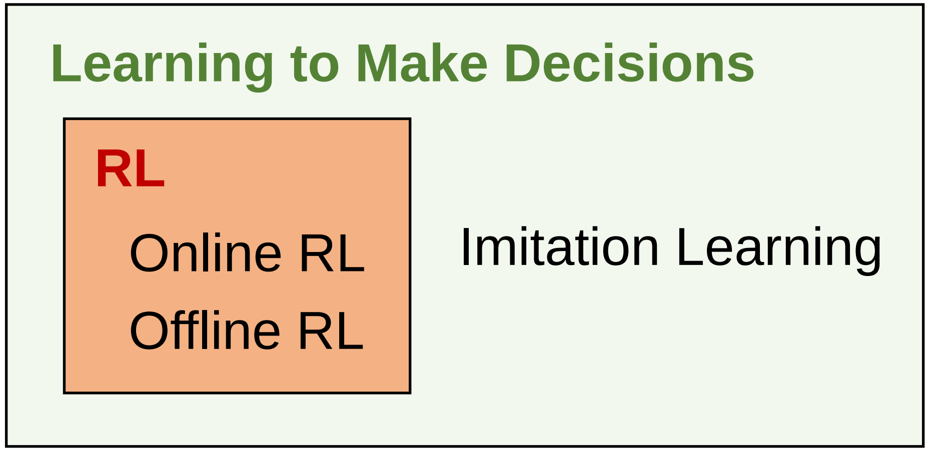
(Yann LeCun, 2016 NIPS)

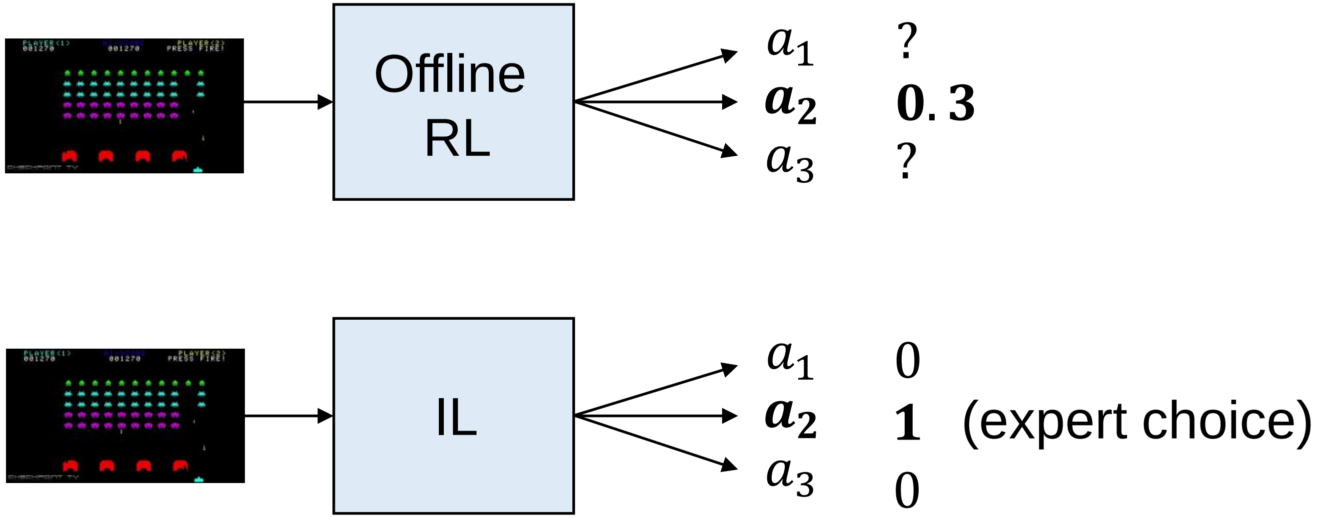
The Scope of This Course

Online RL: through interactions, under bandit / delayed feedback

Offline RL: through existing data, under bandit / delayed feedback

Imitation Learning: through expert data, under label feedback (not in our scope)





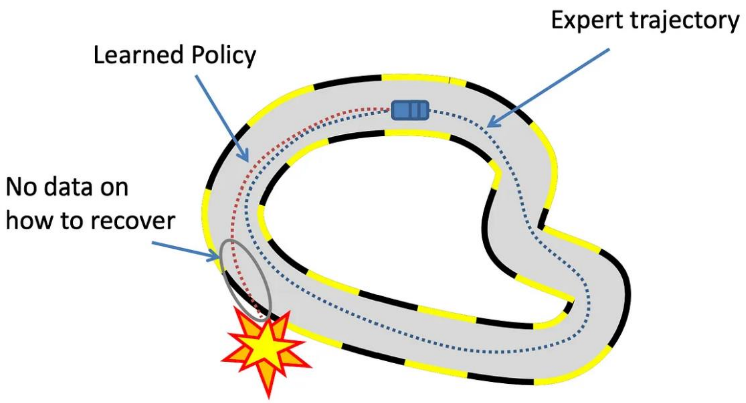
When Is IL (SL) Insufficient?

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

  ⇒ RL can **search** for better solutions

- The expert data has limited coverage
  - Autonomous driving

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



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

  ⇒ RL can provide alignment to the real objective

Challenges in RL

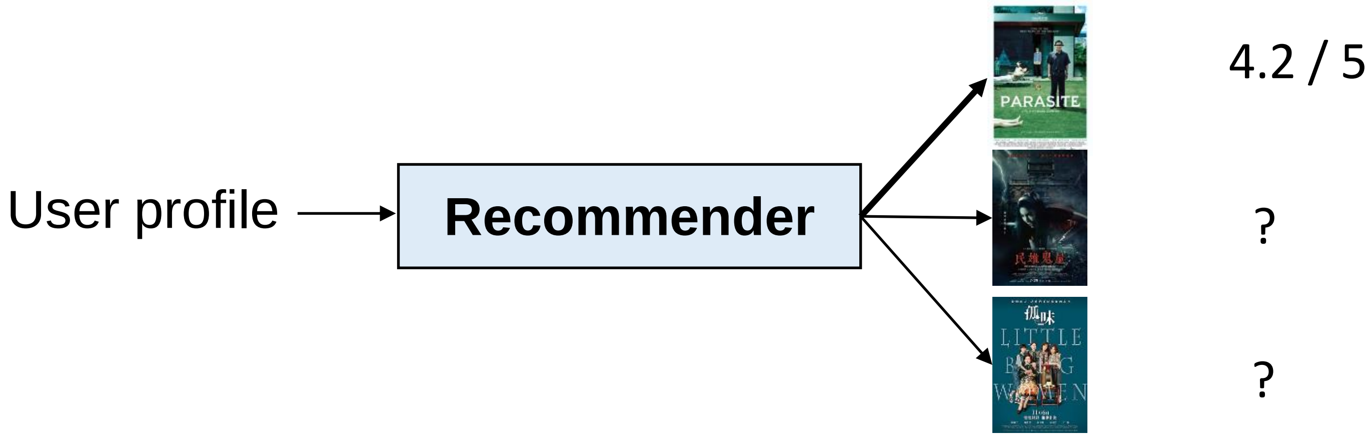
Challenges in RL (1)

Generalization: a key challenge in all machine learning paradigms

(Khosravian and Amirkhani, 2022)

Challenges in RL (2)

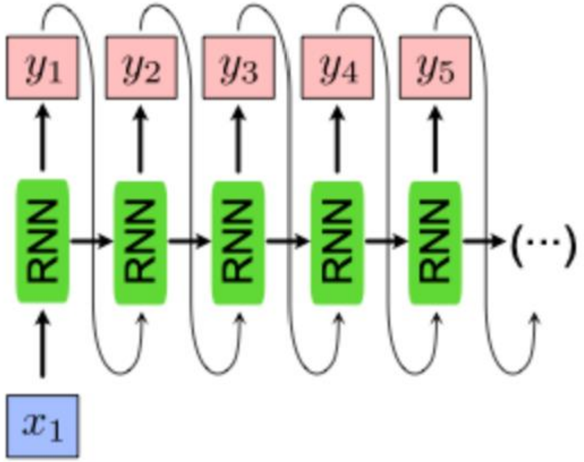
Exploration and exploitation tradeoff (due to bandit feedback)

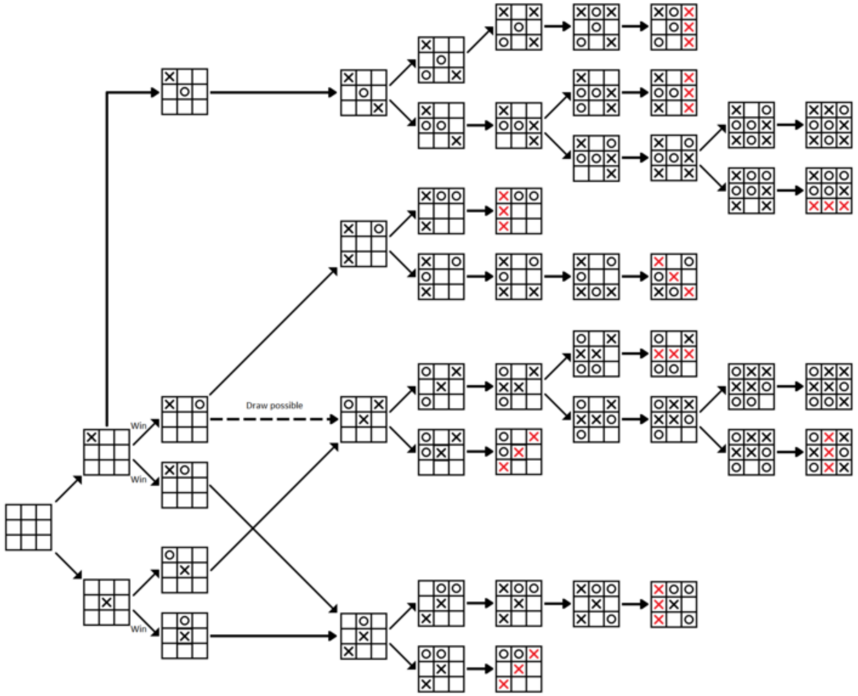


Challenges in RL (3)

Credit assignment (due to delayed and aggregated feedback)

0 0 0 ⋯ **0.97**

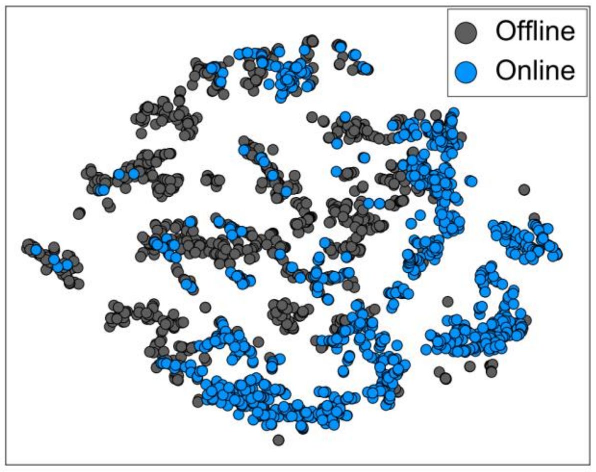


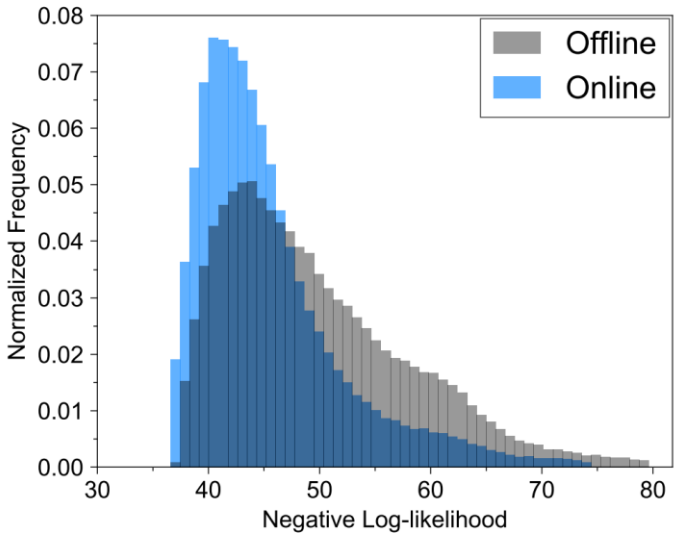


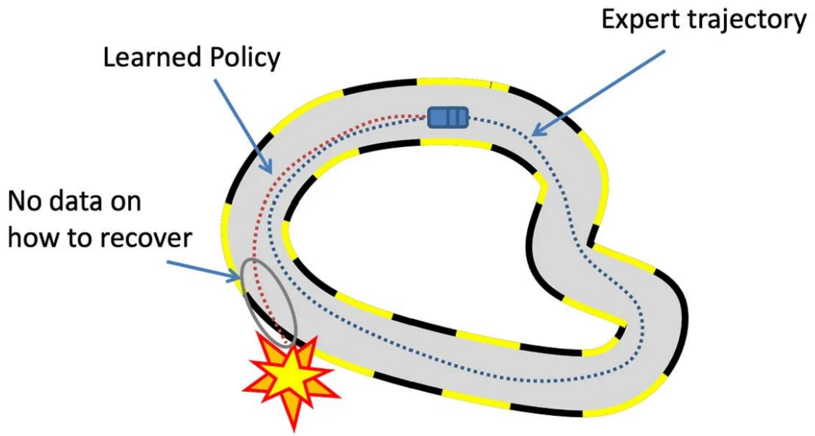
Identify the contribution of each action to the outcome

Challenges in RL (4)

Distribution mismatch / shift (especially in offline RL)







Lee et al., Addressing Distribution Shift in Online Reinforcement Learning with Offline Datasets

Other Challenges

- Reward design
- Safety and ethics
- Robustness under attacks

…

Course Content

Course Content

(Focusing on exploration-exploitation tradeoff)

Part I. Learning in Bandits

- Multi-armed bandits
- Linear bandits
- Contextual bandits
- Adversarial multi-armed bandits
- Adversarial linear bandits

Part II. Basics of MDPs

- Bellman (optimality) equations
- Value iteration
- Policy iteration

(Focusing on credit assignment and distribution mismatch)

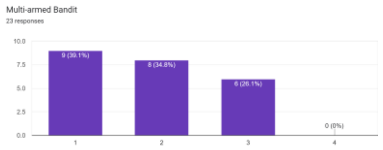
Part III. Learning in MDPs

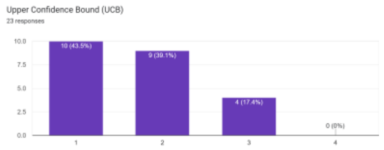
- Approximate value iteration and variants
  - Least-square value iteration
  - Q-Learning
  - DQN
- Policy evaluation
  - Temporal difference
  - Monte Carlo
- Approximate policy iteration and variants
  - Least-square policy iteration
  - (Natural) policy gradient and actor-critic
  - REINFORCE, A2C, PPO, SAC
  - DDPG

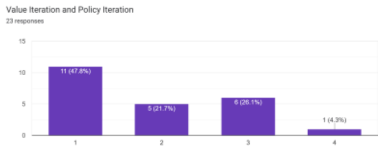
Part IV. Offline RL

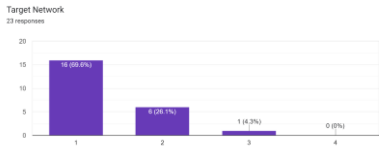
Student Project Presentation

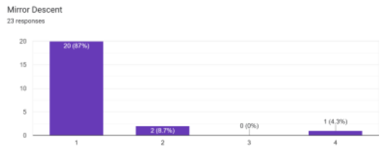
Students' Prior Knowledge

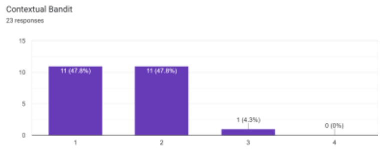

Multi-armed Bandit

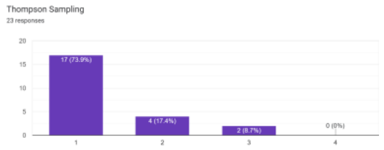

UCB

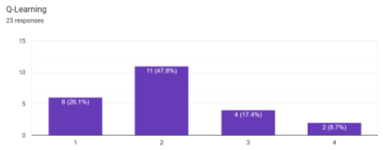

VI & PI

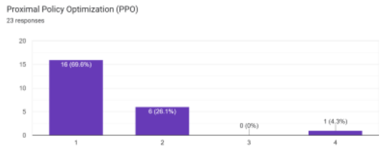

Target Network

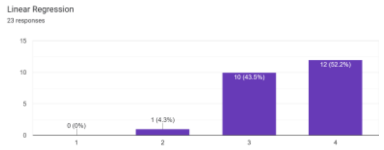

Mirror Descent

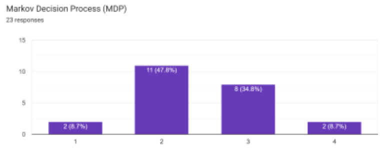

Contextual Bandit

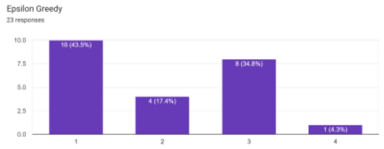

Thompson Sampling

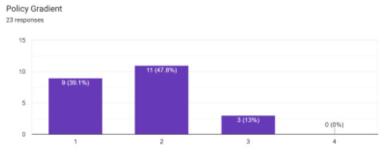

Q-Learning

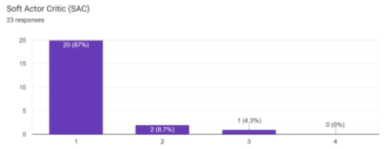

PPO

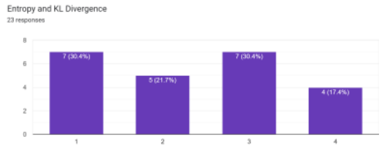

Linear Regression

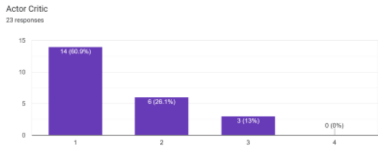

MDP

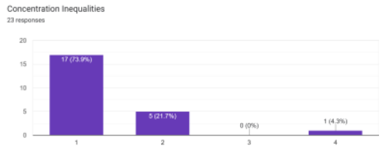

ϵ-greedy


Policy Gradient


SAC


Entropy & KL Divergence


Actor Critic


Concentration Inequality

What Students Want to Learn

- Multi-armed Bandit x1
- Contextual Bandit x1
- Q-learning x1
- Actor Critic x1
- Offline RL x1
- Hands-on programming x4
- RL theory x2
- AlphaGo x3
- RL in ChatGPT x1
- Imitation Learning x2
- Multi-agent RL x3
- RL for continuous robot learning x1

What Students Want to Learn

- Multi-armed Bandit x1
- Contextual Bandit x1
- Q-learning x1
- Actor Critic x1
- Offline RL x1
- Hands-on programming x4
- RL theory x2
- AlphaGo x3
- RL in ChatGPT x1
- Imitation Learning x2
- Multi-agent RL x3
- RL for continuous robot learning x1

Goal of This Course

We will

- Provide a **systematic overview** of basic techniques in RL
- Provide **reasonings** for the design of RL algorithms
- Provide **mathematical tools** to analyze RL algorithms

After taking this course, you should be able to

- Feel grounded when reading other RL materials
- Implement basic RL algorithms
- Know **design principles** of RL algorithms

Prerequisites

- Linear Algebra, Probability, Calculus
- (Optional but helpful) Machine Learning, Convex Optimization
- Python

Before enrolling in this course, note that..

- This is a new course, so there is a lot of uncertainty. We are planning to make RL a regular course, so you can also take it in future semesters.

- We'll go slightly deeper into the theoretical analysis of some topics.
  - May be more than you need
  - Sacrificing some breadth (imitation learning, some practical tricks are omitted)

- This course is neither necessary nor sufficient to learn RL
  - **Not sufficient**: the scope of this course is limited
  - **Not necessary**: The math may be more than you need
  - **Could be beneficial**: if you want a systematic view or unified understanding for various RL algorithms

Online Resources

- Youtube courses
  - UC Berkeley CS285
  - DeepMind x UCL RL Lectures
- Theoretical course materials
  - Csaba Szepesvari
  - Nan Jiang, Wen Sun, Chi Jin
  - Dylan Foster & Sasha Rakhlin
  - Haipeng Luo (bandit)
- Books
  - Sutton & Barto, Reinforcement Learning: An Introduction
  - Agarwal et al., Reinforcement Learning: Theory and Algorithms
  - Lattimore & Szepesvari, Bandit Algorithms (bandit)
- Implementations
  - OpenAI SpinningUp
  - OpenAI StableBaseline3
  - ShangtongZhang

Assignments (60%)

- **Four assignments.** Each consists of
  - Math / algorithm design problems
  - Programming tasks (using PyTorch)
  - PyTorch tutorial: https://www.youtube.com/watch?v=c36lUUr864M
- Assignment late policy
  - 5 late days distributed as you like
  - Each additional late day results in 20% deduction in the corresponding assignment
- The rules about discussion with classmates or LLM will be clarified in HW1

Final Project (35%)

- Breakdown
  - Proposal (5%)
  - Mid-term report (5%)
  - Presentation (10%)
  - Final report (15%)
- Types of projects (basically any!)
  - Application
  - Algorithm design
  - Systematic comparison
  - Theoretical understanding
  - Literature survey

  (see the specification on the website for more information)
- 2-3 students in a group
- Proposal deadline: **Feb.16** (feel free to schedule meeting with me before finalize it)

Class Participation (5%)

- In-class and Piazza discussions

TA & Office Hour

- **TA:** Haolin Liu
  - Email: srs8rh@virginia.edu
  - Office hour: M 11:00-12:00
- Me
  - Email: chenyu.wei@virginia.edu
  - Office hour: Th 15:30-16:30pm at Rice 409, or by appointment

Questions?