Introduction

COS IW 03: Hands-on Reinforcement Learning

Karthik Narasimhan

Introductions

Instructor:

Karthik Narasimhan

TA:

Xi Chen

Time/Location:

- Tuesdays 3:00pm-4:20pm in CS 301
- Office hours:
  Karthik Narasimhan: Mondays 4pm-5pm, Computer Science 422
  Xi Chen: Fridays 1:30 - 3:30pm, CS 003

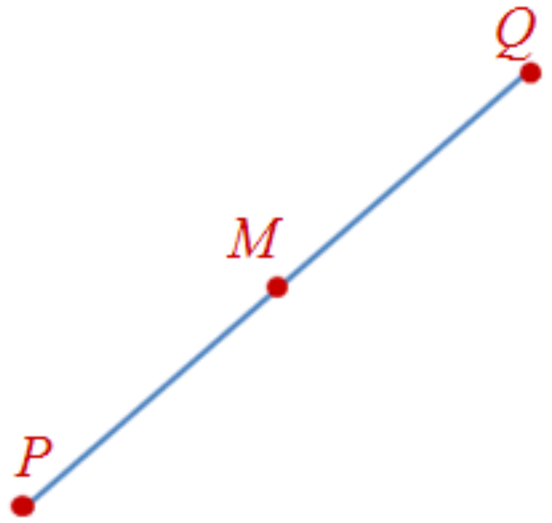
Logistics

- Not typical lecture-style. This is independent work after all!
- Attendance is mandatory
- Please use Piazza for all questions and to share useful information with each other!
- Office hours will be updated on class website
- ‘Intro to Neural Networks’ tutorial by Xi (date: TBD)

The semester in phases

P1: Brainstorm, literature review, finalize ideas
(Feb 18)

P2: Initial setup (data collection, tool survey, etc)
(Feb 25)



P3: Midpoint (preliminary) results
(Mar 10)

P4: Endgame (finalize results, write report, presentations)
(end of April)

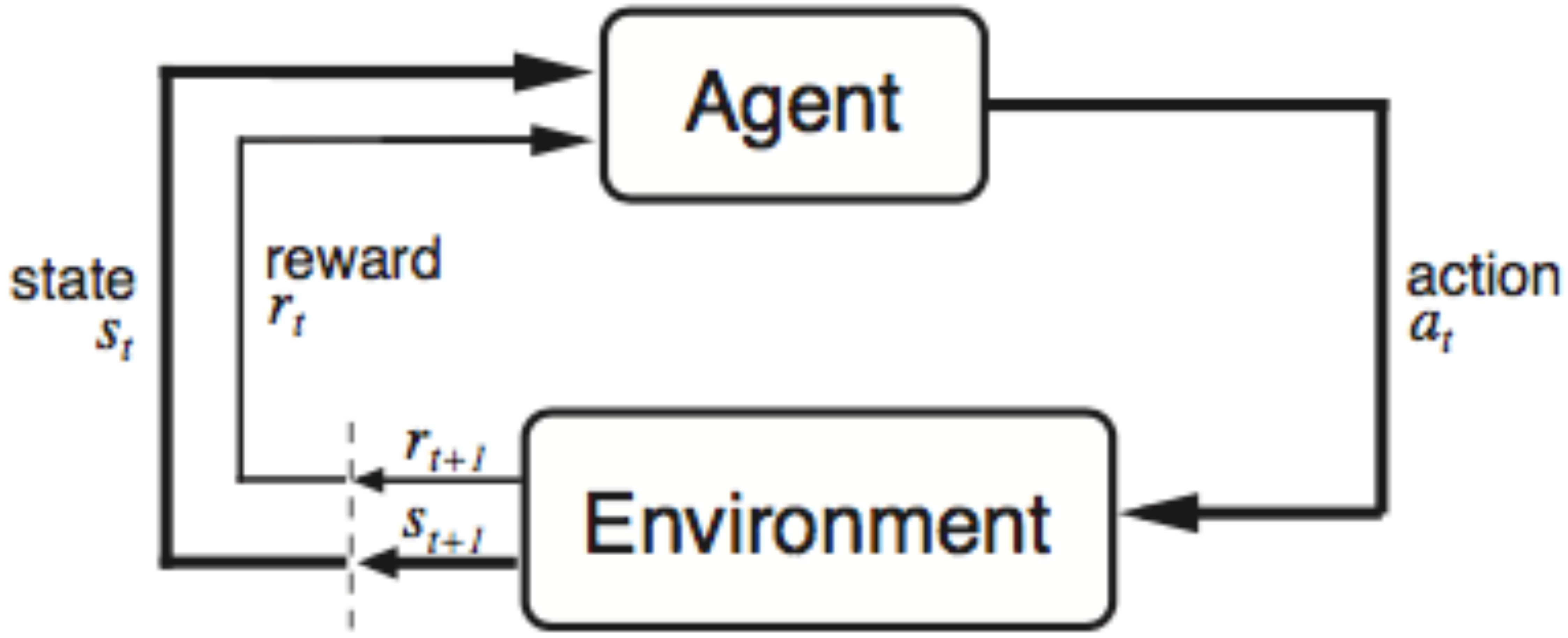
Important IW milestones

Spring 2020 Single Semeseter Projects Schedule

Feb. 6, 2020	Attend the "Getting Started" Information Meeting 4:30p.m. in CS 105
Feb. 17, 2020 5:00pm	SEAS Funding Deadline ***Due in person by 5pm.** ****Open to ANY COS IW student** **(Both AB and BSE can apply. Both single-term and two-term/ thesis can apply)**
Feb. 23, 2020	Submit a Written Project Proposal, by 11:59p.m.
March 8, 2020	Submit the Checkpoint Form, by 11:59p.m.
March 30, 2020 - April 3, 2020	Sign Up to Give an Oral Presentation ****First-time BSE IW students, only** **Note for seminar students:** All seminar students will give an Your instructor will share information about logistics.
March 31, 2020	Attend "How to Give an IW Talk" 4:30p.m. in CS 105
April 14, 2020	Attend "How to Write an IW Paper" 4:30p.m. in CS 105
April 19, 2020	Submit Slides for an Oral Presentation, by 11:59p.m. ****All seminar students and first-time BSE IW students, only**
April 20-24, 2020	Give an IW Oral Presentation ****First-time BSE IW students, only** **Note for seminar students:** All seminar students will give an oral presentation. Your instructor will share information about logistics.
May 1, 2020	Submit a Written Final Report, by 11:59p.m.

Apply to this!

Reinforcement Learning



Sequential Decision Making

- Agent : has ability to affect change
- Action : causes change
- Environment : is affected by action
- Reward : feedback w.r.t a goal
- Time

Why is RL challenging?

- Delayed feedback

action 1 ○○○○ *action n*

Reward +10

$\Rightarrow$ *How to perform credit assignment for individual actions*

- Large number of possible action sequences

$\Rightarrow$ *Need for effective exploration*

Markov Decision Process

State s = Observed Environment

Action a = Command to execute

State 1	*Action*	*State 2*	
Location: Field *Wind level*: 3 *Time*: 12pm *Mailbox*: closed	open mailbox	*Location*: Field *Wind level*: 3 *Time*: 12pm *Mailbox*: open	*Reward* +1

Markov assumption:
limited history

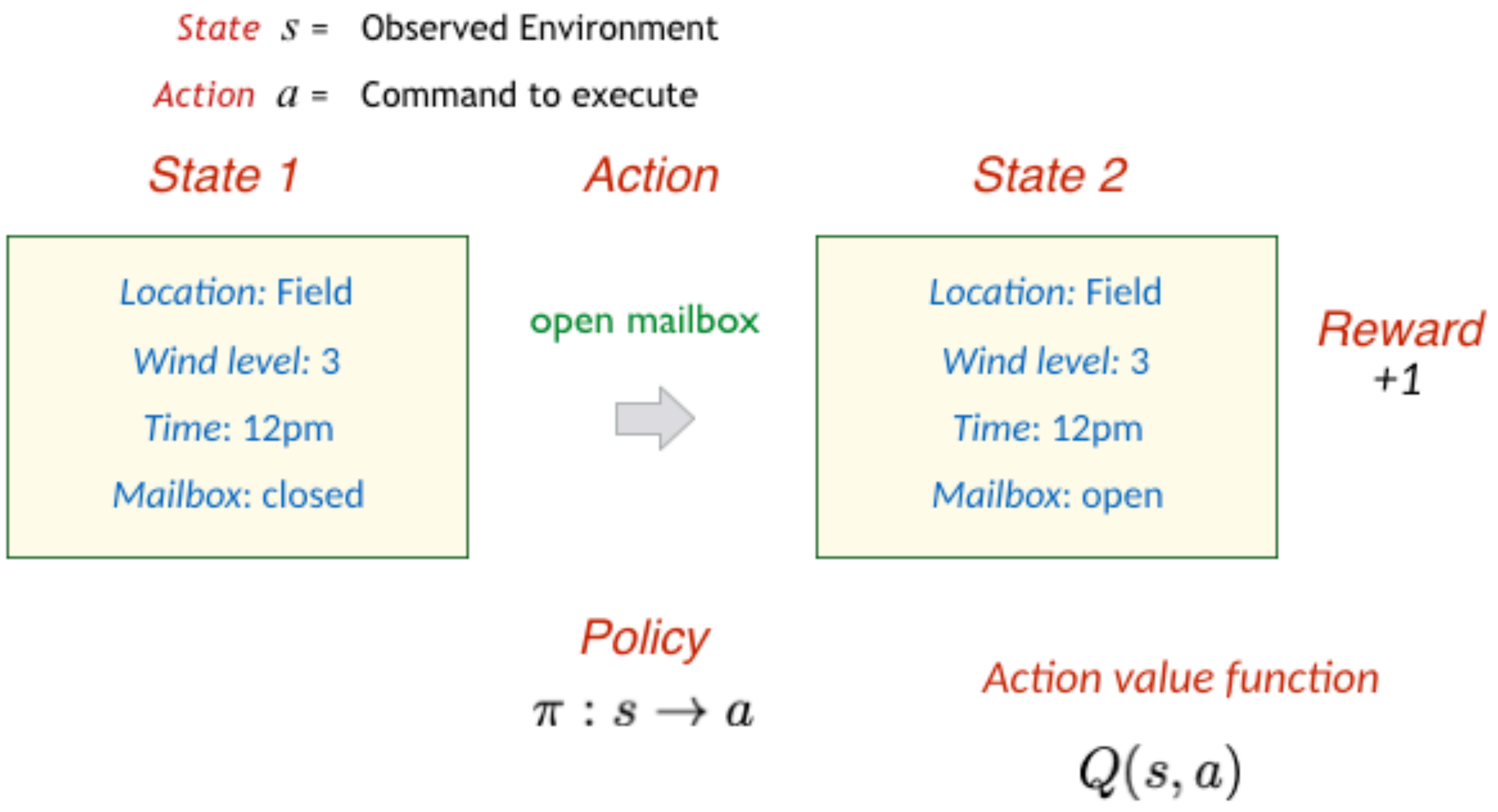
Policy

$$\pi : s \rightarrow a$$

Action value function

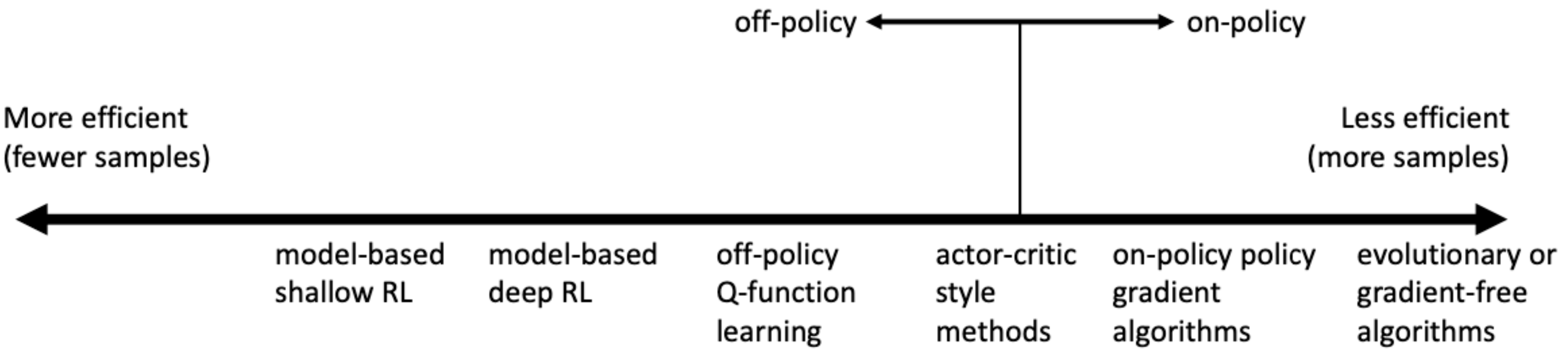
$$Q(s, a)$$

Types of RL algorithms



- **Policy gradients:** directly learn a policy to maximize reward
- **Value-based:** estimate value function or Q-function of the optimal policy (no explicit policy)
- **Actor-critic:** estimate value function or Q-function of current policy and use it to improve the policy
- **Model-based RL:** estimate transition and reward models of the environment, then use it for planning, improving policy, etc.

Comparison: sample efficiency



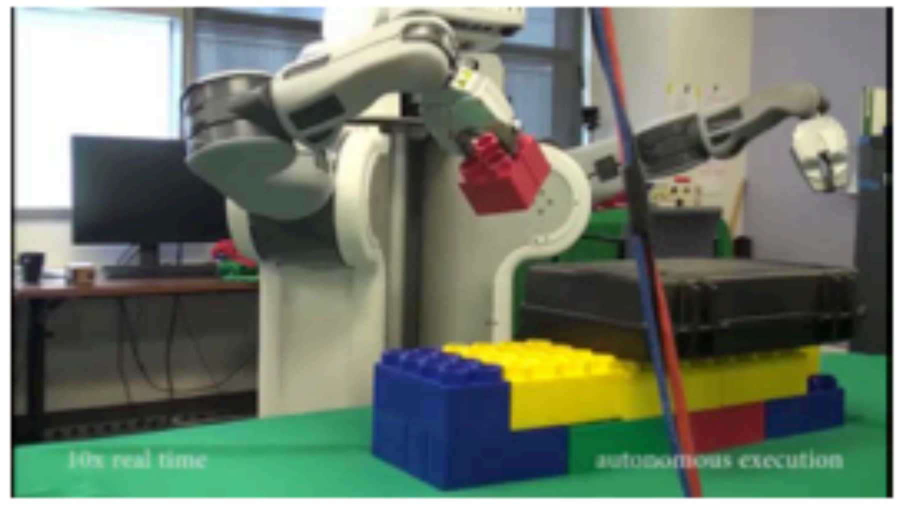
Why would we use a *less* efficient algorithm?

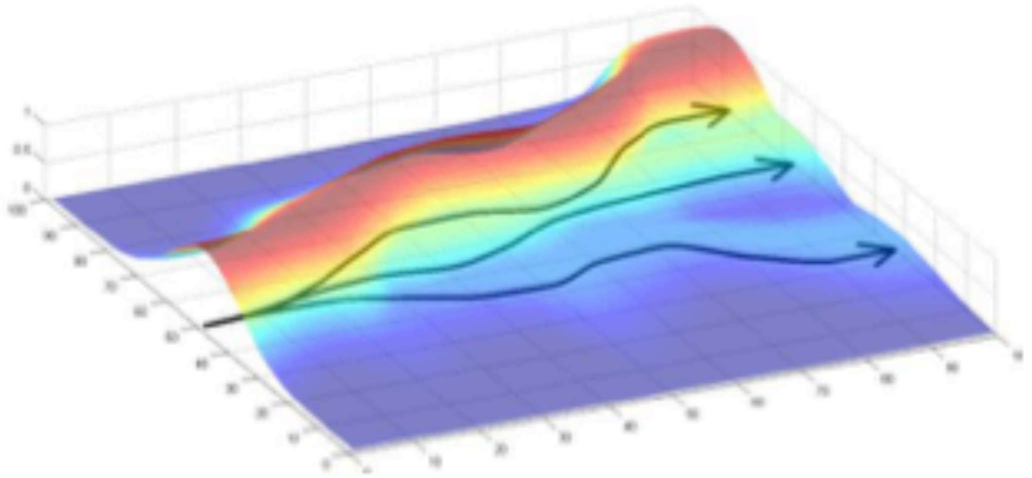
Wall clock time is not the same as efficiency!

(slide credits: Sergey Levine)

Comparison: assumptions

- Common assumption #1: full observability
  - Generally assumed by value function fitting methods
  - Can be mitigated by adding recurrence
- Common assumption #2: episodic learning
  - Often assumed by pure policy gradient methods
  - Assumed by some model-based RL methods
- Common assumption #3: continuity or smoothness
  - Assumed by some continuous value function learning methods
  - Often assumed by some model-based RL methods





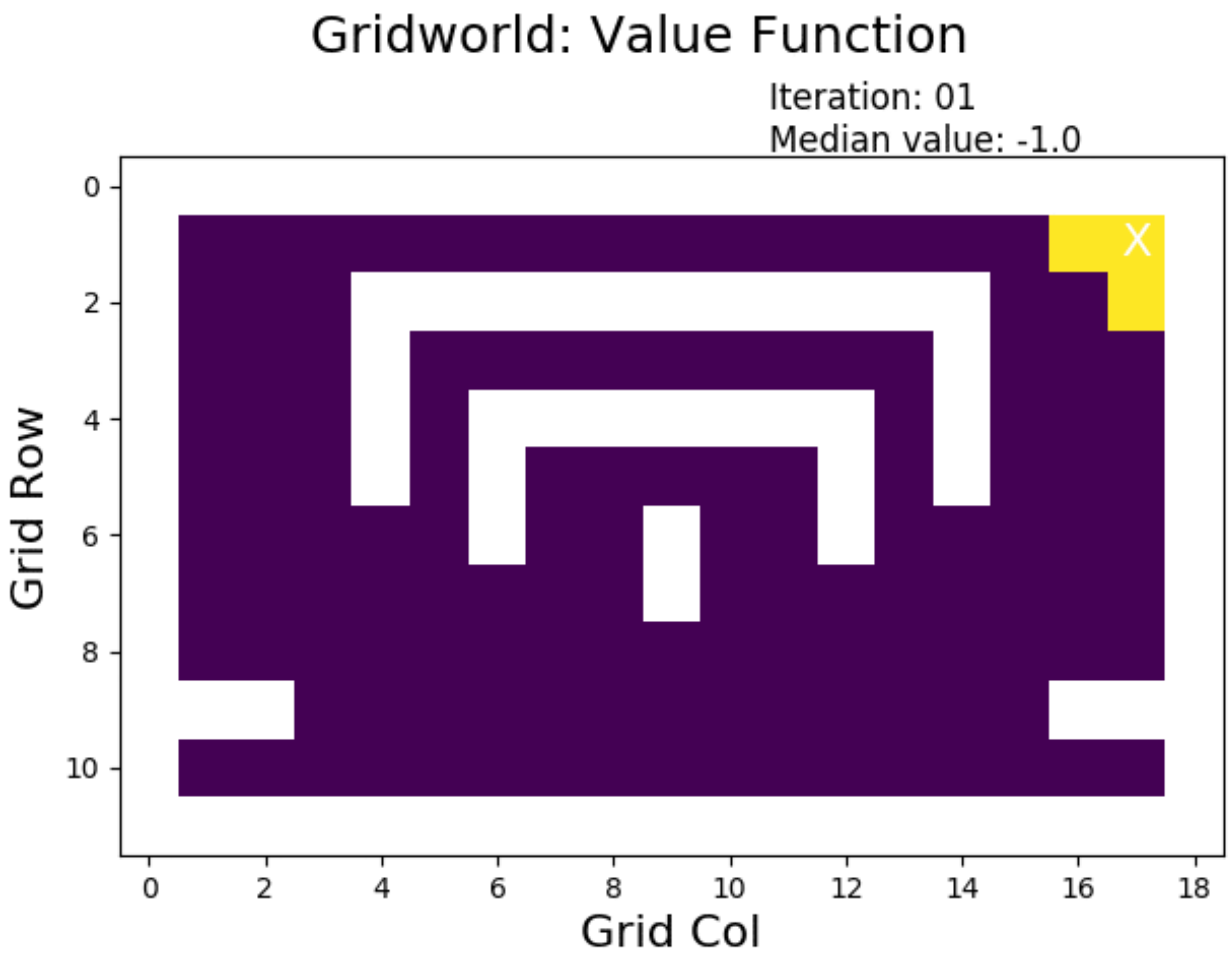
(slide credits: Sergey Levine)

Example: value iteration

Initialize V arbitrarily, e.g.,$V(s) = 0$, for all $s \in \mathcal{S}^+$

Repeat
$\quad \Delta \leftarrow 0$
$\quad$ For each $s \in \mathcal{S}$:
$\quad\quad v \leftarrow V(s)$
$\quad\quad V(s) \leftarrow \max_a \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V(s')\right]$
$\quad\quad \Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$ (a small positive number)

Output a deterministic policy, π, such that
$\quad \pi(s) = \arg\max_a \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V(s')\right]$



OpenAI Gym

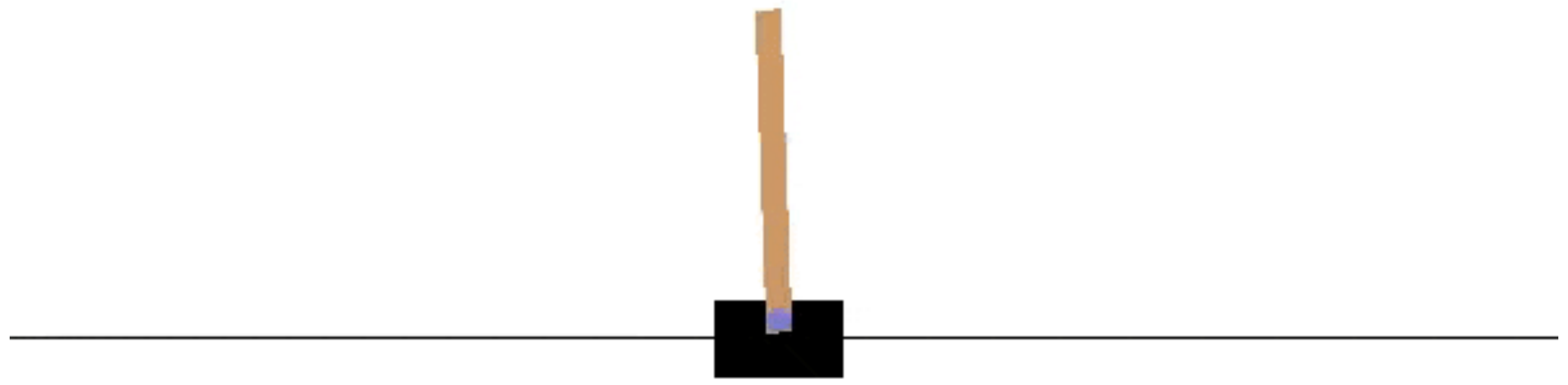
Gym

Gym is a toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games like Pong or Pinball.

View documentation ›

View on GitHub ›

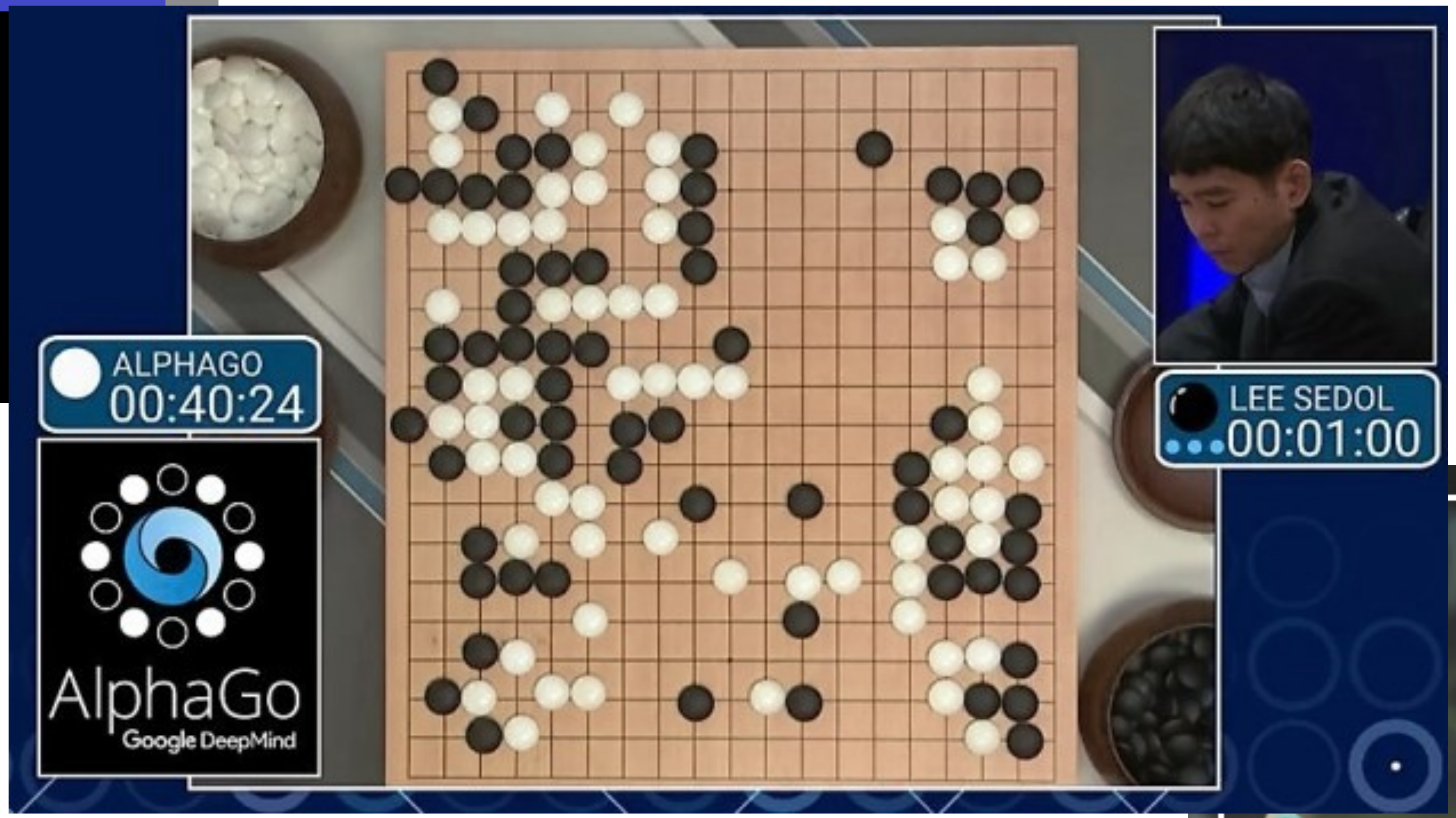
```python
import gym
env = gym.make("CartPole-v1")
observation = env.reset()
for _ in range(1000):
  env.render()
  action = env.action_space.sample() # your agent here (this takes random actions)
  observation, reward, done, info = env.step(action)

  if done:
    observation = env.reset()
env.close()
```



Great API for designing and using environments

Recent successes

Mnih et al., 2015



Silver et al., 2016

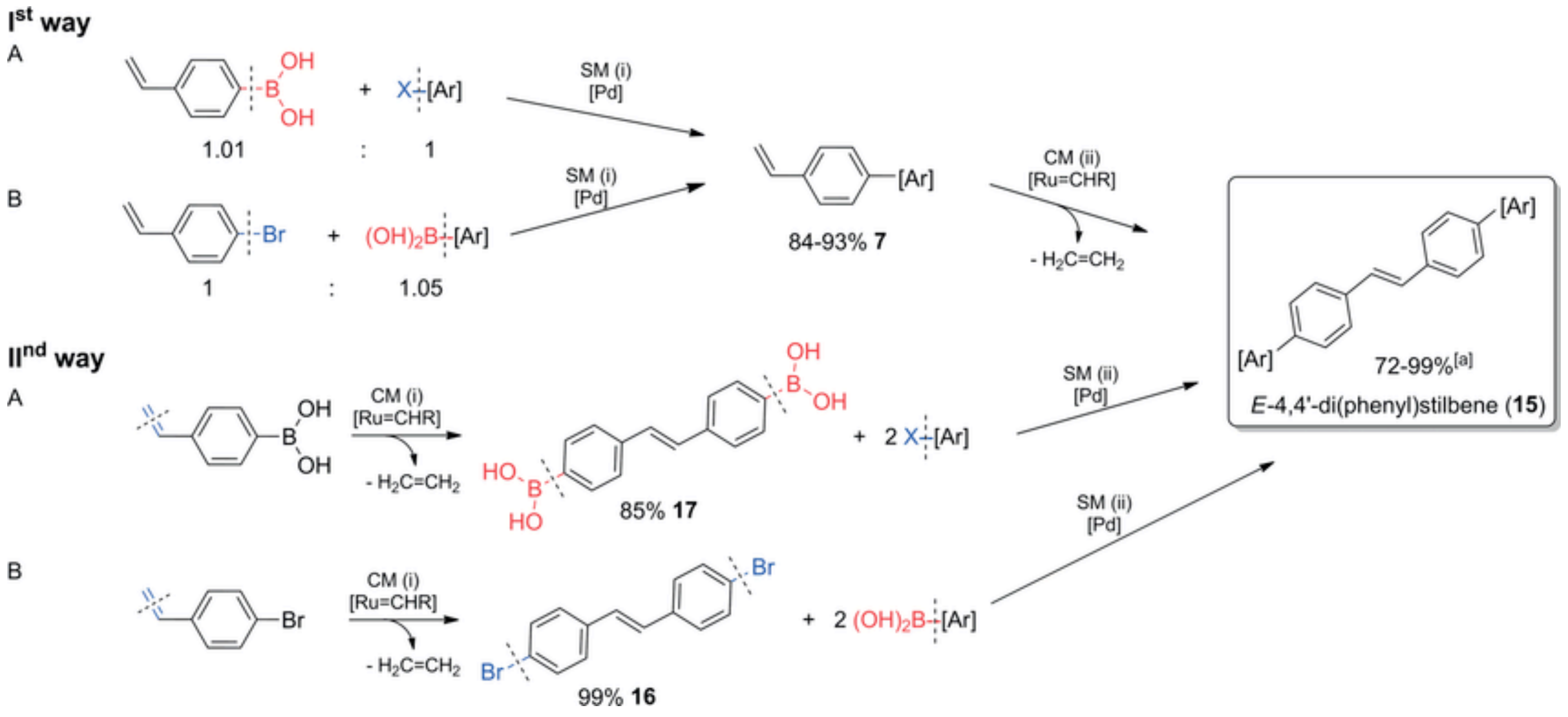
OpenAI, 2018

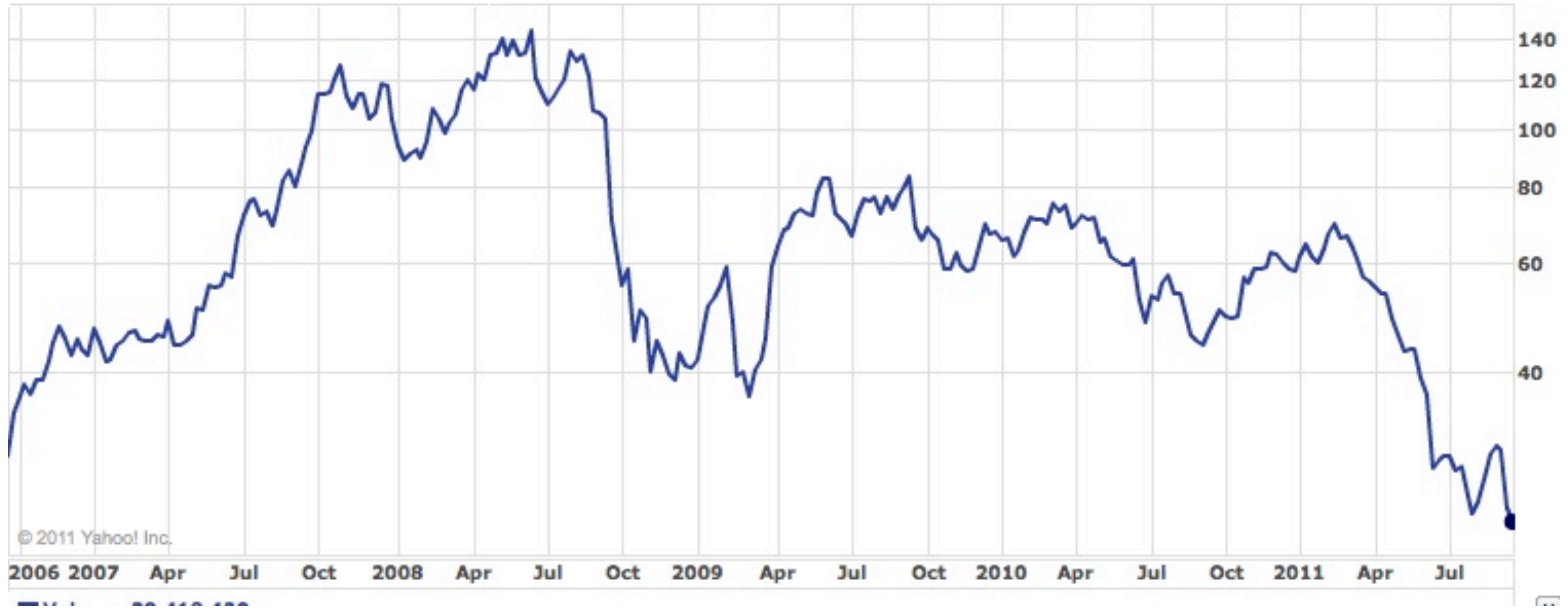
Some project ideas

- Develop an environment for a new game and train RL agents on it
    - Use OpenAI gym for environment framework
    - Investigate a wide range of RL agents (e.g. from OpenAI baselines)

Some project ideas

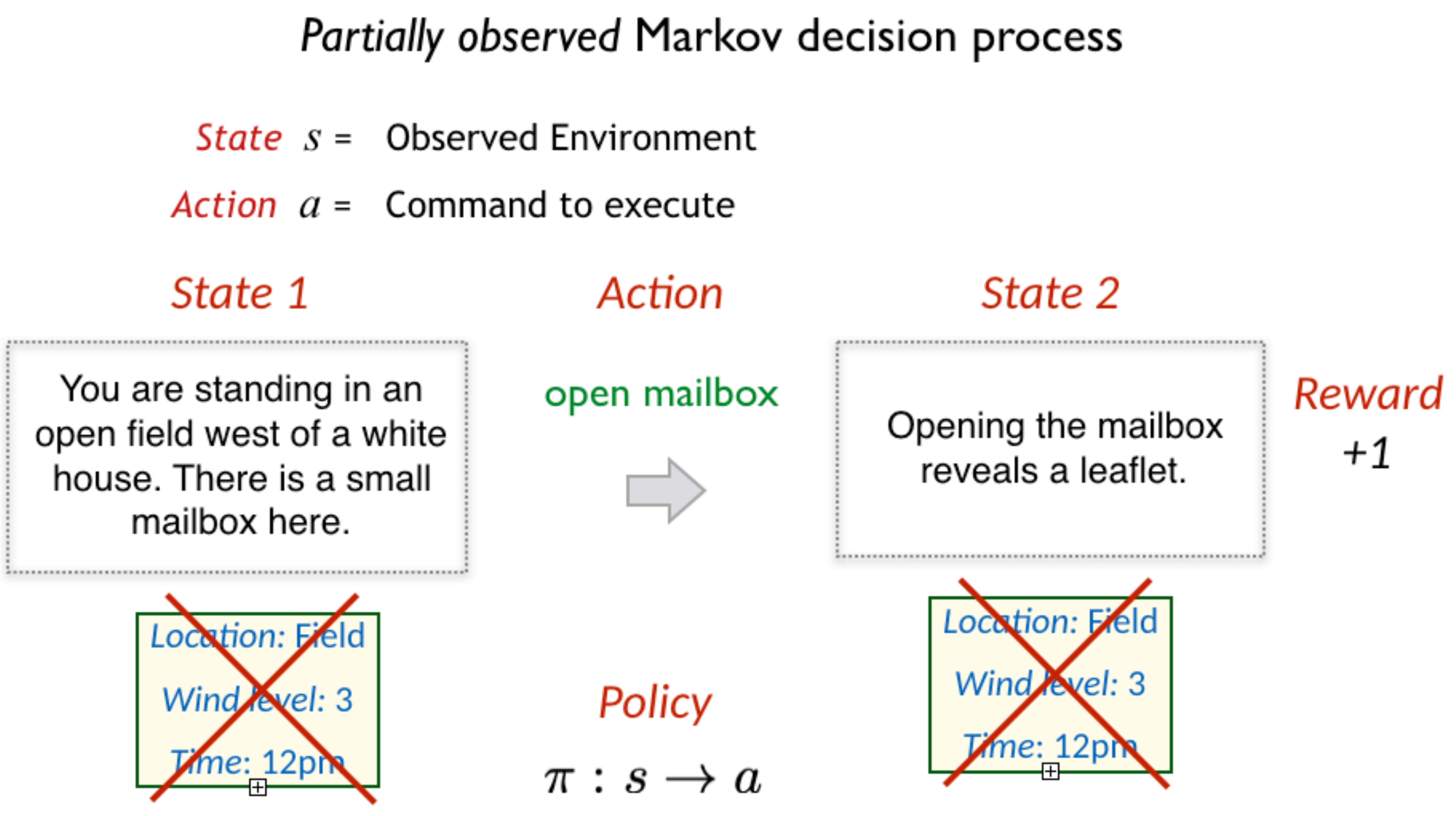
- Use reinforcement learning to tackle a decision making problem in real-world domains
  - Traffic control (https://flow-project.github.io/)
  - Chemical reaction synthesis
  - Financial prediction (tricky!)
  - Robotics
  - Solve differential equations
  - …





Some project ideas

- Text adventure games



Narasimhan et al., 2015

Microsoft TextWorld is an open-source, extensible engine that both generates and simulates text games. You can use it to train reinforcement learning (RL) agents to learn skills such as language understanding and grounding, combined with sequential decision making.

```
You are navigating through a house. You've just entered a serious
study. There is a gross looking mantle in the room. It has nothing
on it. You see a closed rusty toolbox. Now why would someone leave
that there?

Looks like there is a locked door. Find the key to unlock the
door. You should try going east.
```

Microsoft Textworld

Some project ideas

- Learning without rewards?

Curiosity-driven Exploration by Self-supervised Prediction

Deepak Pathak Pulkit Agrawal Alexei A. Efros Trevor Darrell

University of California, Berkeley

ICML 2017

[Download Paper] [Github Code]

(a) learn to explore on Level-1 (b) explore faster on Level-2

Some project ideas

- Develop and study new RL algorithms! Challenges to tackle include:
  - **Sample efficiency:** How fast can you learn?
  - **Ability to generalize:** Can a policy trained on one Atari game generalize to another?
  - **Multi-task learning:** Can you train a single agent to be good at several tasks?
  - **Robustness:** Is your algorithm robust to noise in environmental signals (state observations, rewards, etc.)? Analyze what breaks and why. Even better, try to fix it!
  - **Fairness/Bias:** Can RL algorithms be biased/unfair in the same way as supervised methods like image classification? How can we characterize this?

If you have other ideas, come talk to us!

Collaboration

- Working in teams is great! (2-3 people at max)
- Make sure each member has a clearly defined sub-part and responsibility
- Please do share ideas and resources with each other!

Project updates

- We will create a shared Google drive folder where you can each upload slides every week
- Running presentation, append new slides to the same deck
- Key points to include:
    1. What you accomplished the previous week
    2. Any hurdles you are currently encountering
    3. Concrete plan for next week

Brainstorming

For next meeting

- Please fill out Google form with initial ideas before the next meeting (will be announced on Piazza later tonight)