Reinforcement Learning

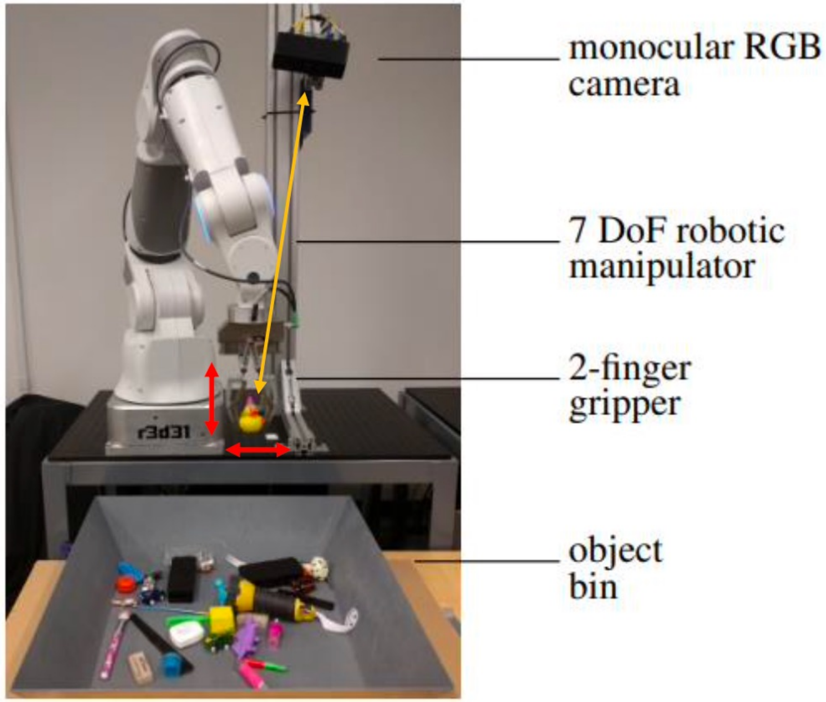
Computer Engineering Department
Sharif University of Technology

Mohammad Hossein Rohban, Ph.D.

Spring 2025

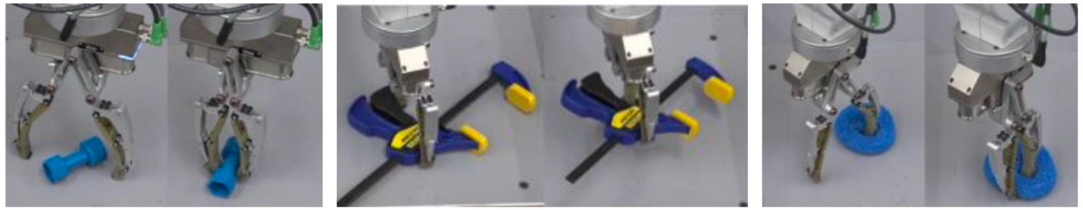
Courtesy: Some slides are adopted from CS 285 Berkeley, and CS 234 Stanford, and Pieter Abbeel's compact series on RL.

Motivation



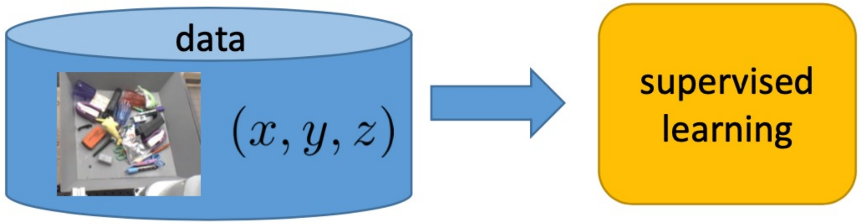
Option 1:

Understand the problem, design a solution



Option 2:

Set it up as a machine learning problem

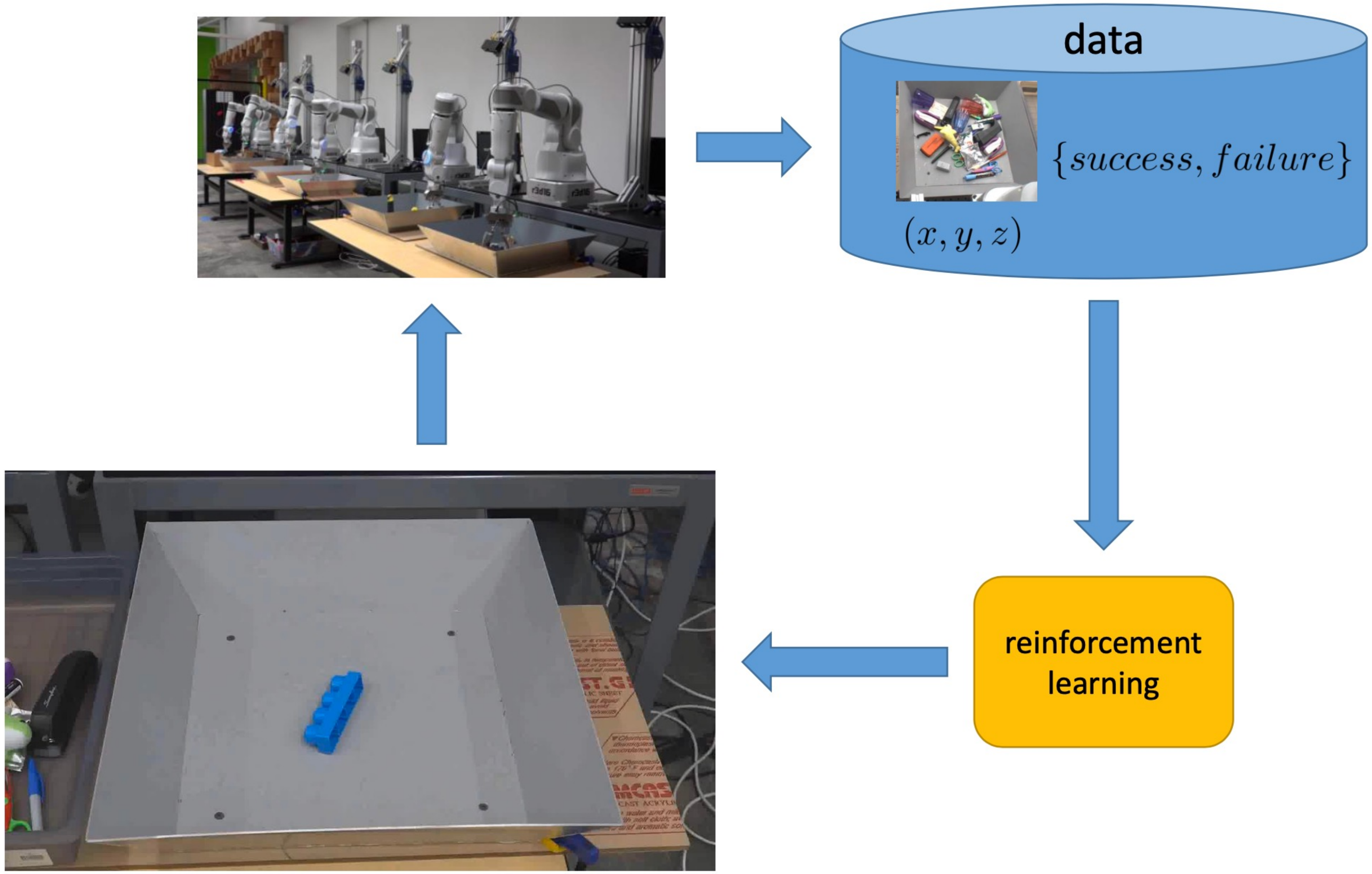


Courtesy: CS 285 course, Berkeley

Motivation (cont.)

Courtesy: CS 285 course, Berkeley

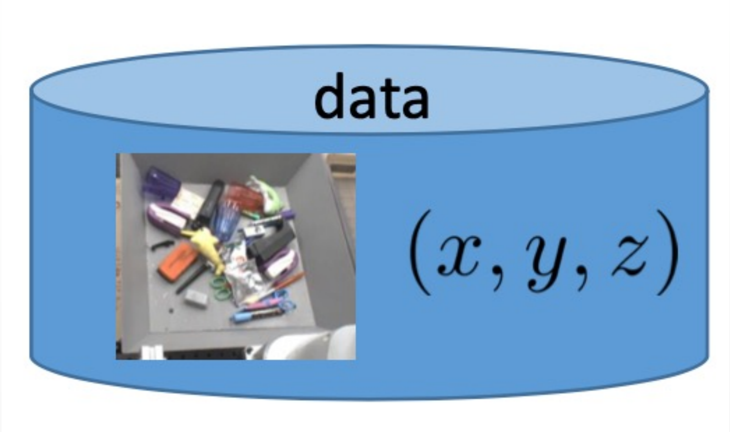
Motivation (cont.)

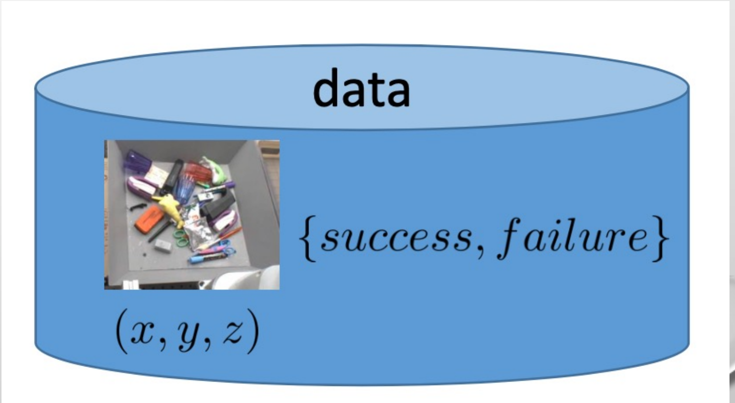


Courtesy: CS 285 course, Berkeley

Motivation (cont.)

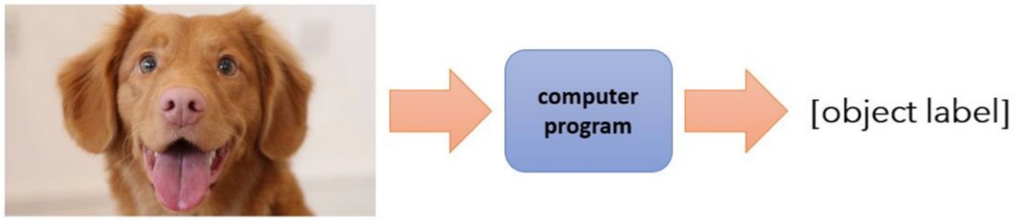
- Supervised learning:
  - Ground truth is known in advance.
  - Training data are usually static and iid.
- Reinforcement learning:
  - The best action (**policy**) is usually unknown a priori.
  - Sequence of actions is needed.
  - A series of trial and error (search) is performed.
    - Usually delayed **reward** shows goodness of the trial.
  - Data is dynamic (**exploration**) and non-iid.





What is Reinforcement Learning?

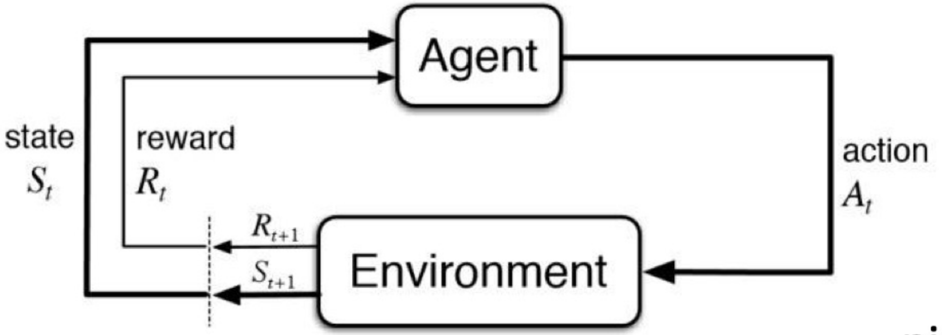
supervised learning



$$
\begin{aligned}
\text{input: } & \mathbf{x} \\
\text{output: } & \mathbf{y} \\
\text{data: } & \mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\} \\
\text{goal: } & f_\theta(\mathbf{x}_i) \approx \mathbf{y}_i
\end{aligned}
$$

someone gives this to you

reinforcement learning

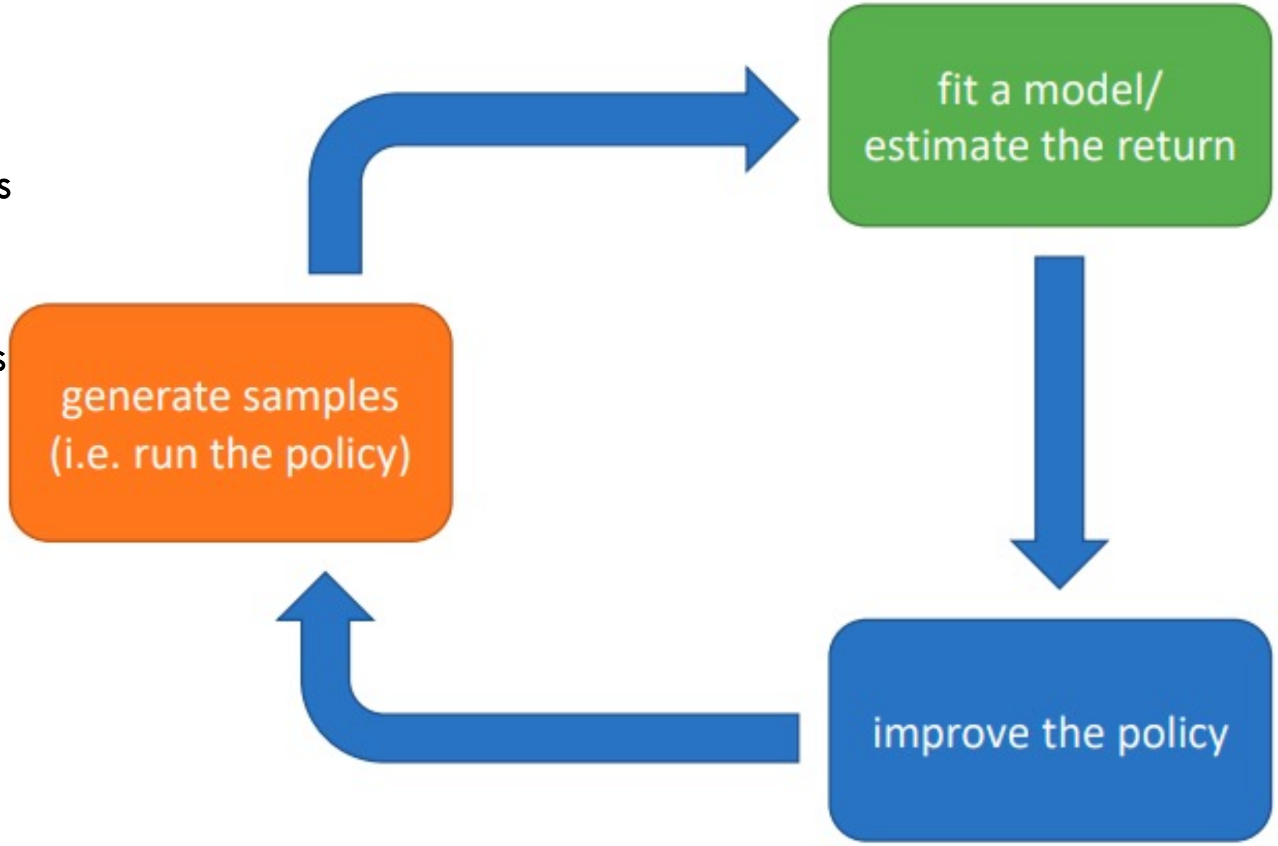


$$
\begin{aligned}
\text{input: } & \mathbf{s}_t \; \textit{at each time step} \\
\text{output: } & \mathbf{a}_t \; \textit{at each time step} \\
\text{data: } & (\mathbf{s}_1, \mathbf{a}_1, r_1, \ldots, \mathbf{s}_T, \mathbf{a}_T, r_T) \\
\text{goal: } & \text{learn } \pi_\theta : \mathbf{s}_t \rightarrow \mathbf{a}_t \\
& \text{to maximize } \textstyle\sum_t r_t
\end{aligned}
$$

pick your own actions

Courtesy: CS 285 course, Berkeley

The Anatomy of Reinforcement Learning



gripping objects on random or else “best guess” locations

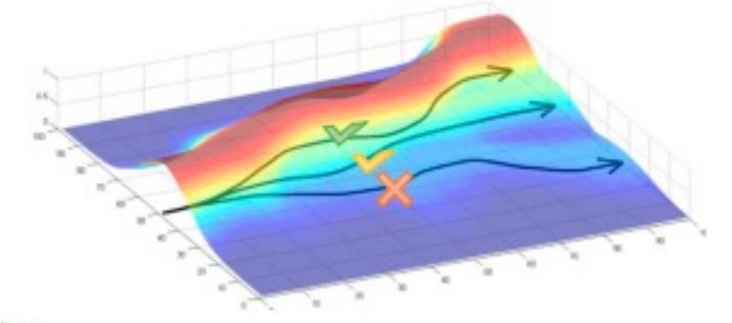
how well was our choice of (x, y, z); execute gripping!

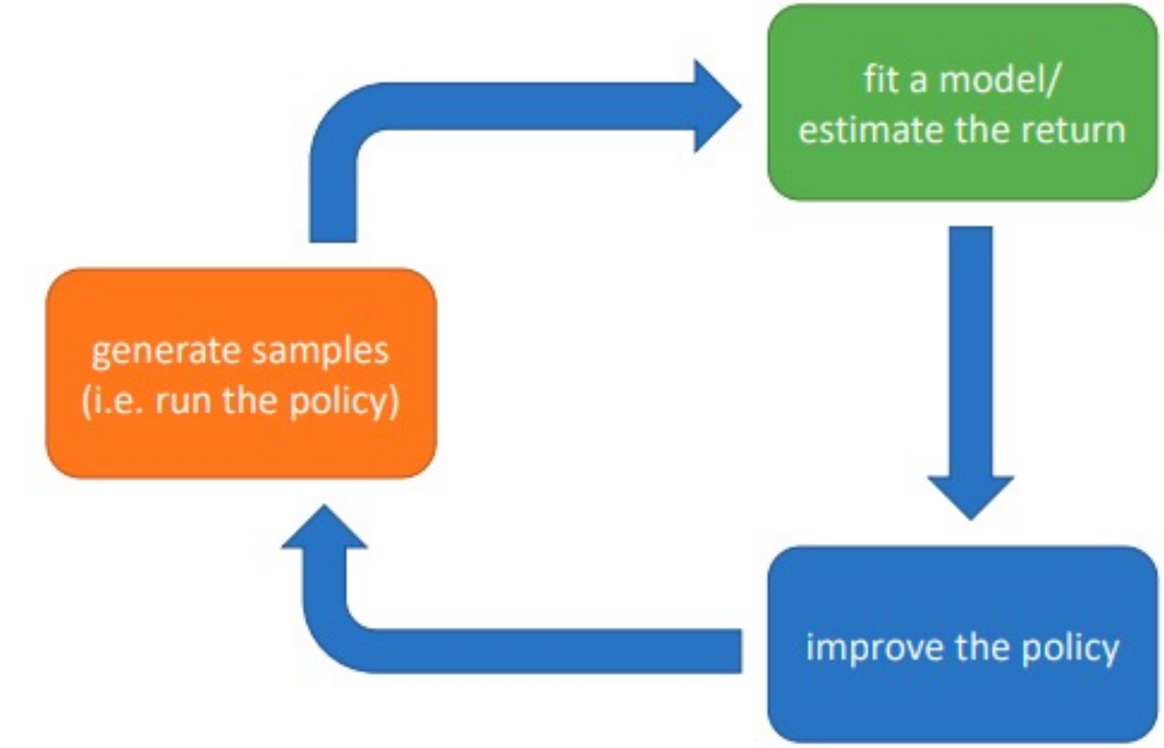
or … when I grip some object on a specific location, what happens physically; i.e. how its (x, y, z)’s change?

improve our “best guess”

Courtesy: CS 285 course, Berkeley

A Simple Example



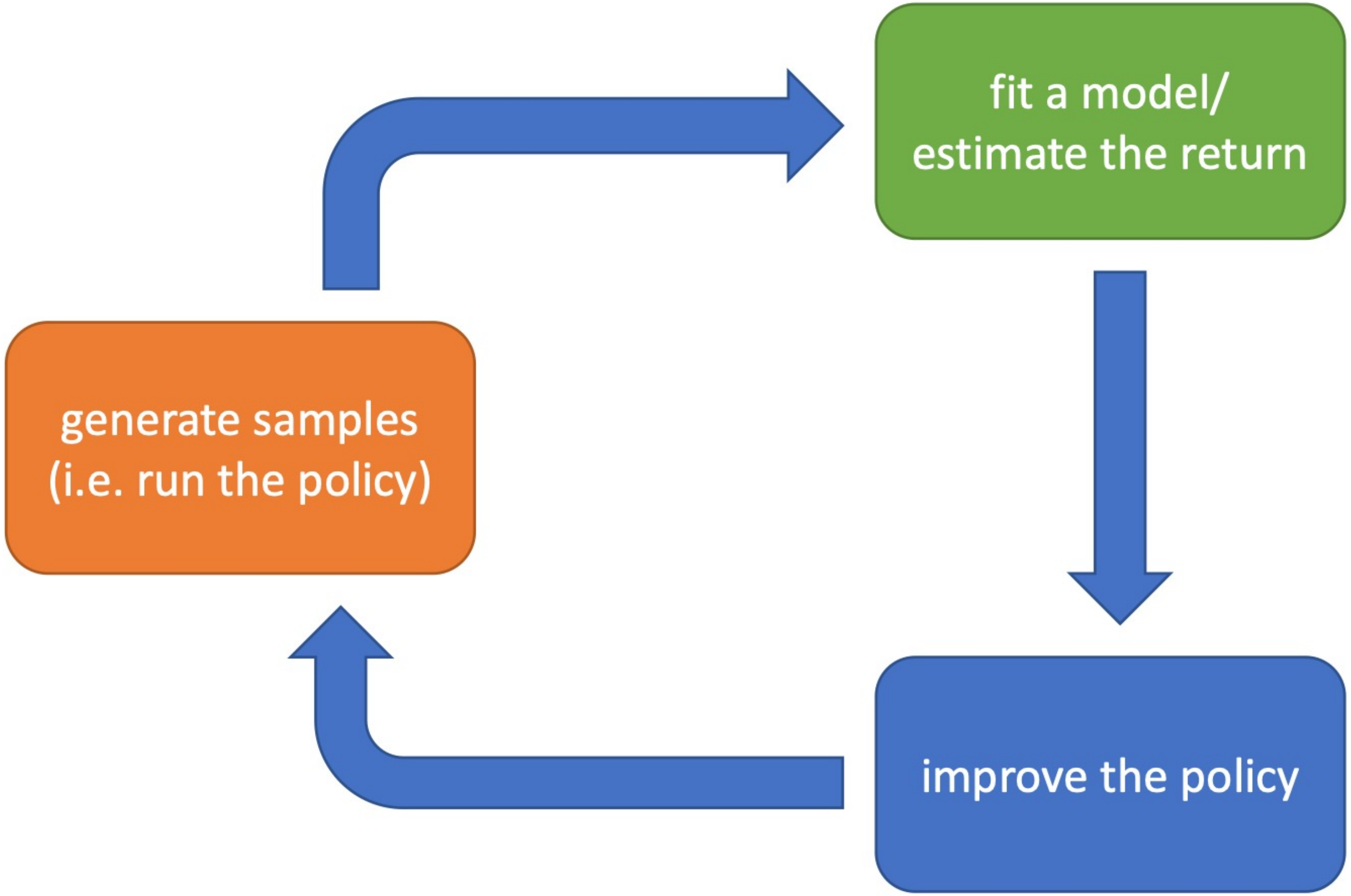


$$J(\theta) = E_\pi\left[\sum_t r_t\right] \approx \frac{1}{N}\sum_{i=1}^{N}\sum_t r_t^i$$

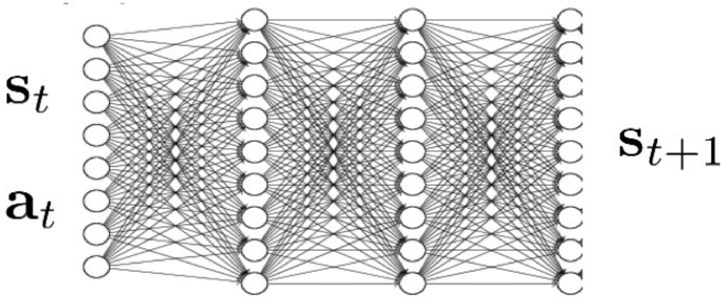
$$\theta \leftarrow \theta + \alpha\nabla_\theta J(\theta)$$

Courtesy: CS 285 course, Berkeley

Another Example



learn f_ϕ such that $\mathbf{s}_{t+1} \approx f_\phi(\mathbf{s}_t, \mathbf{a}_t)$

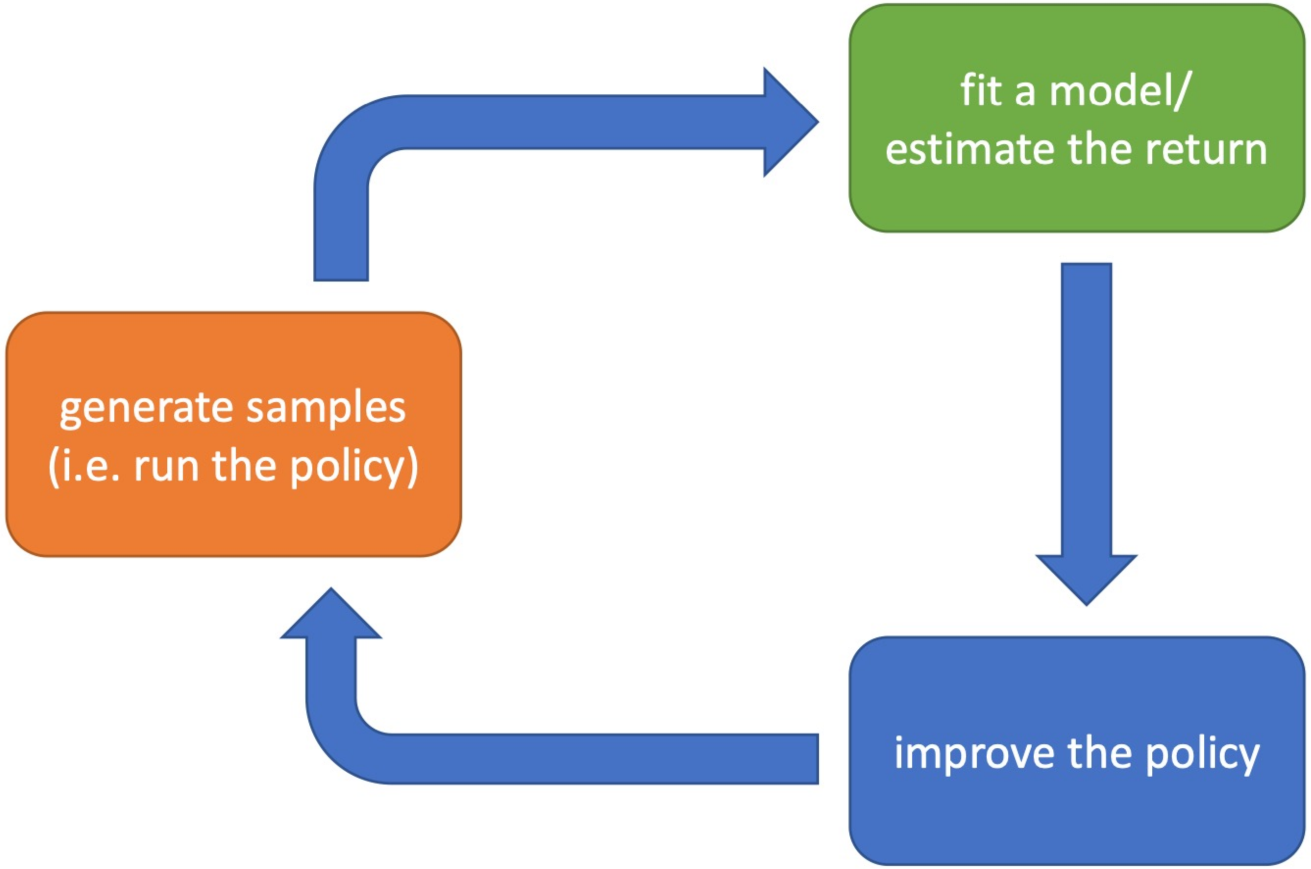


backprop through f_ϕ and r to train $\pi_\theta(\mathbf{s}_t) = \mathbf{a}_t$

Which parts are expensive?

real robot/car/power grid/whatever:
1x real time, until we invent time travel

MuJoCo simulator:
up to 10000x real time



$$J(\theta) = E_\pi\left[\sum_t r_t\right] \approx \frac{1}{N}\sum_{i=1}^{N}\sum_t r_t^i$$

trivial, fast

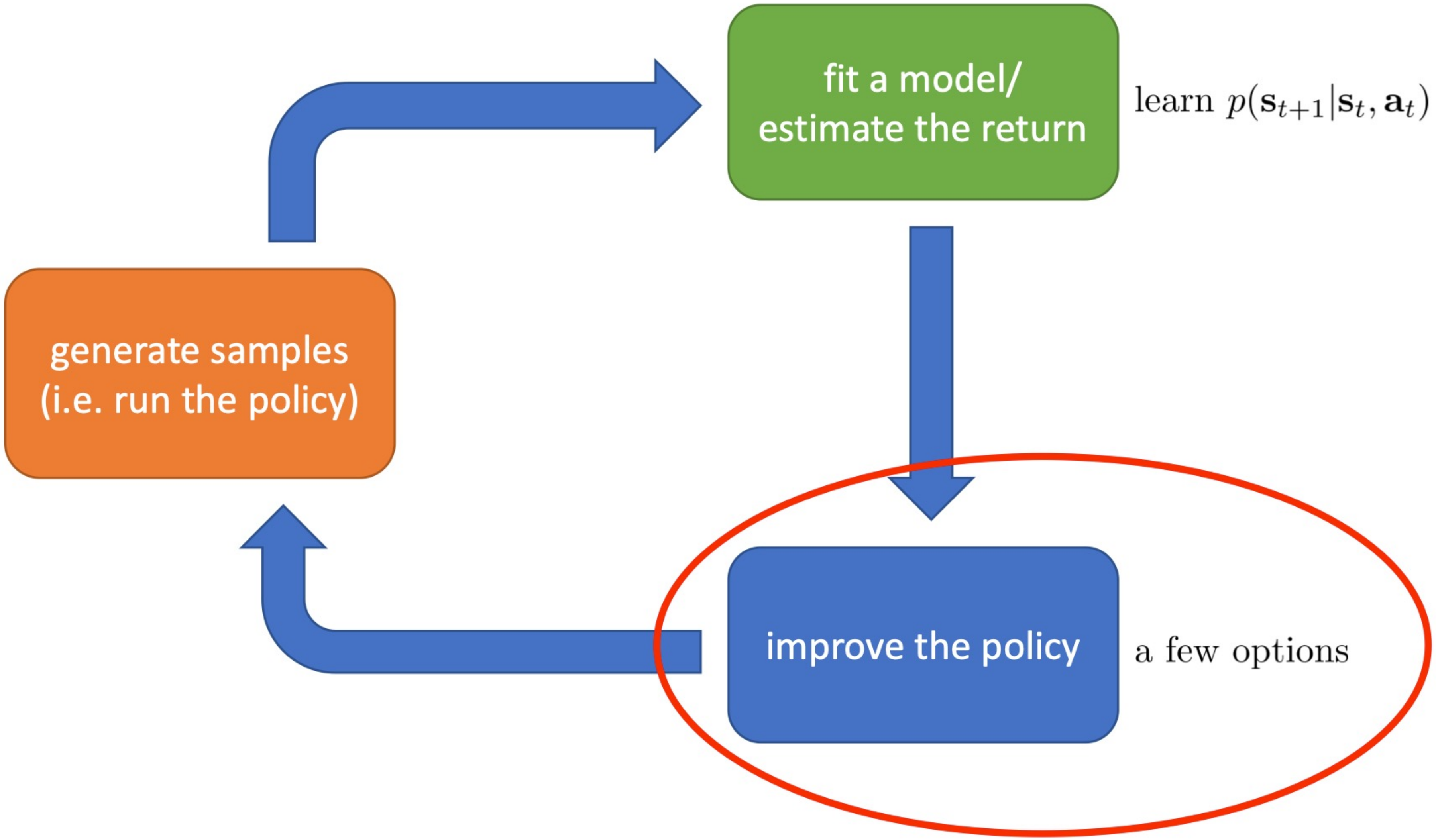
learn $\mathbf{s}_{t+1} \approx f_\phi(\mathbf{s}_t, \mathbf{a}_t)$

expensive

$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

backprop through f_ϕ and r to train $\pi_\theta(\mathbf{s}_t) = \mathbf{a}_t$

Model-based RL



Courtesy: CS 285 course, Berkeley

Value-based RL

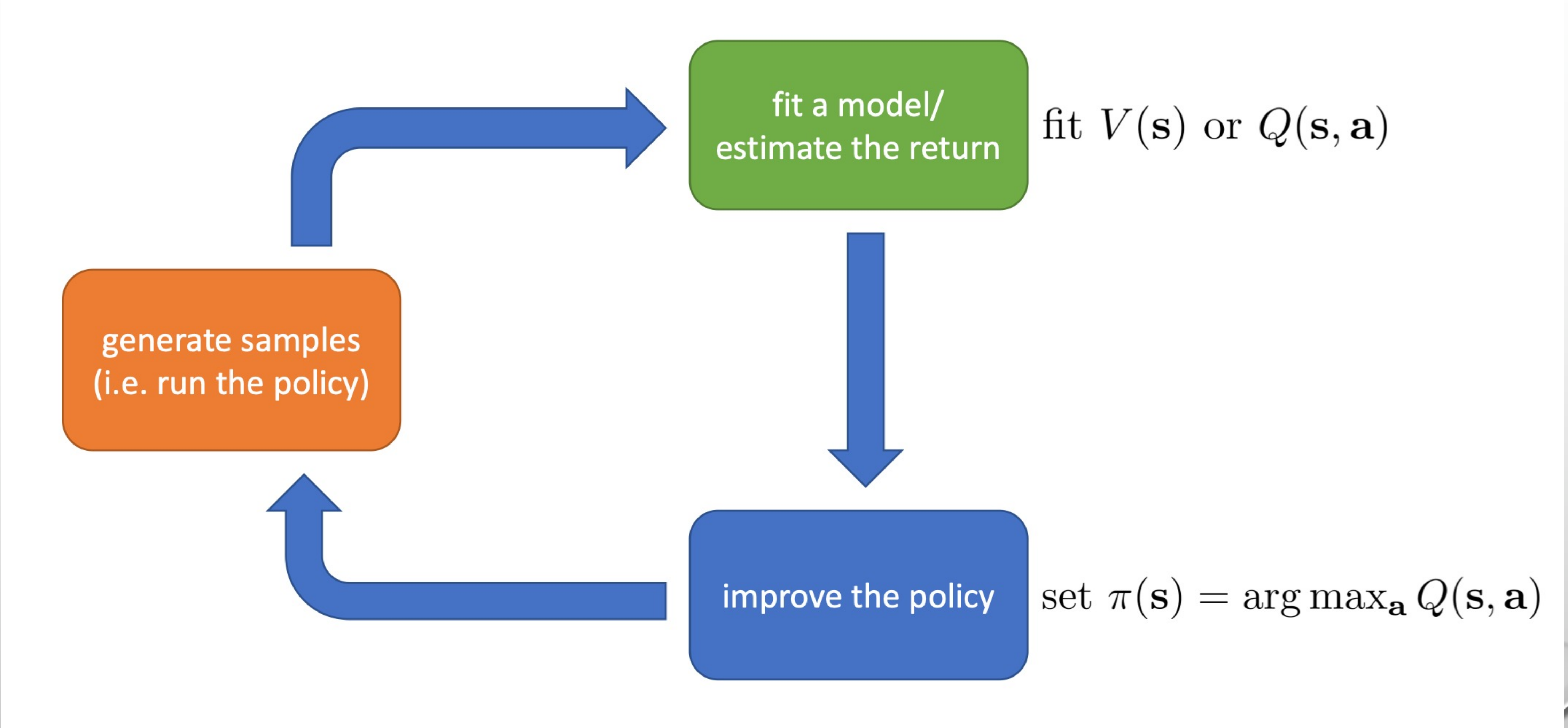
$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) = \sum_{t'=t}^{T} E_{\pi_\theta}\left[r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) | \mathbf{s}_t, \mathbf{a}_t\right]$: total reward from taking $\mathbf{a}_t$ in $\mathbf{s}_t$

$V^\pi(\mathbf{s}_t) = \sum_{t'=t}^{T} E_{\pi_\theta}\left[r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) | \mathbf{s}_t\right]$: total reward from $\mathbf{s}_t$

$$V^\pi(\mathbf{s}_t) = E_{\mathbf{a}_t \sim \pi(\mathbf{a}_t | \mathbf{s}_t)}[Q^\pi(\mathbf{s}_t, \mathbf{a}_t)]$$

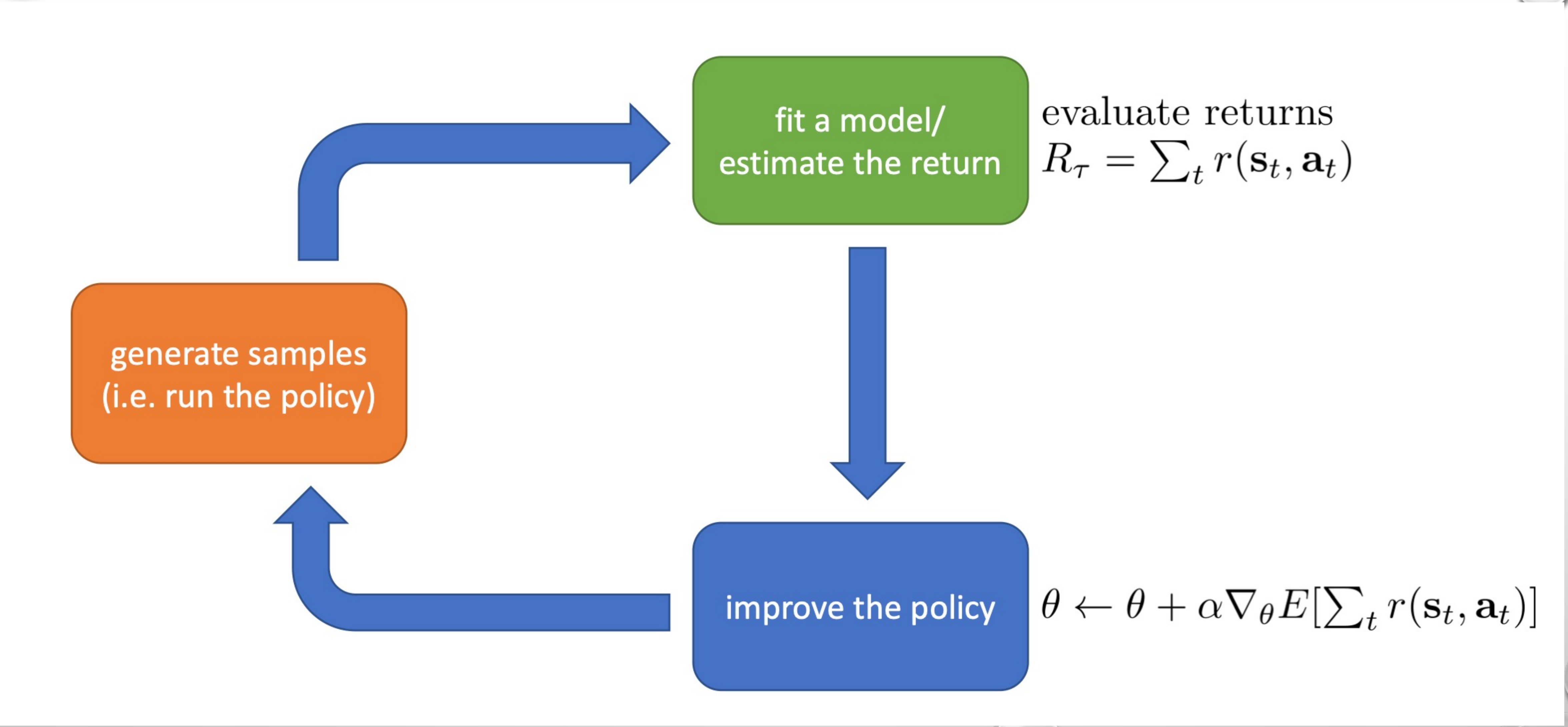
$E_{\mathbf{s}_1 \sim p(\mathbf{s}_1)}[V^\pi(\mathbf{s}_1)]$ is the RL objective!

Value-based RL (cont.)



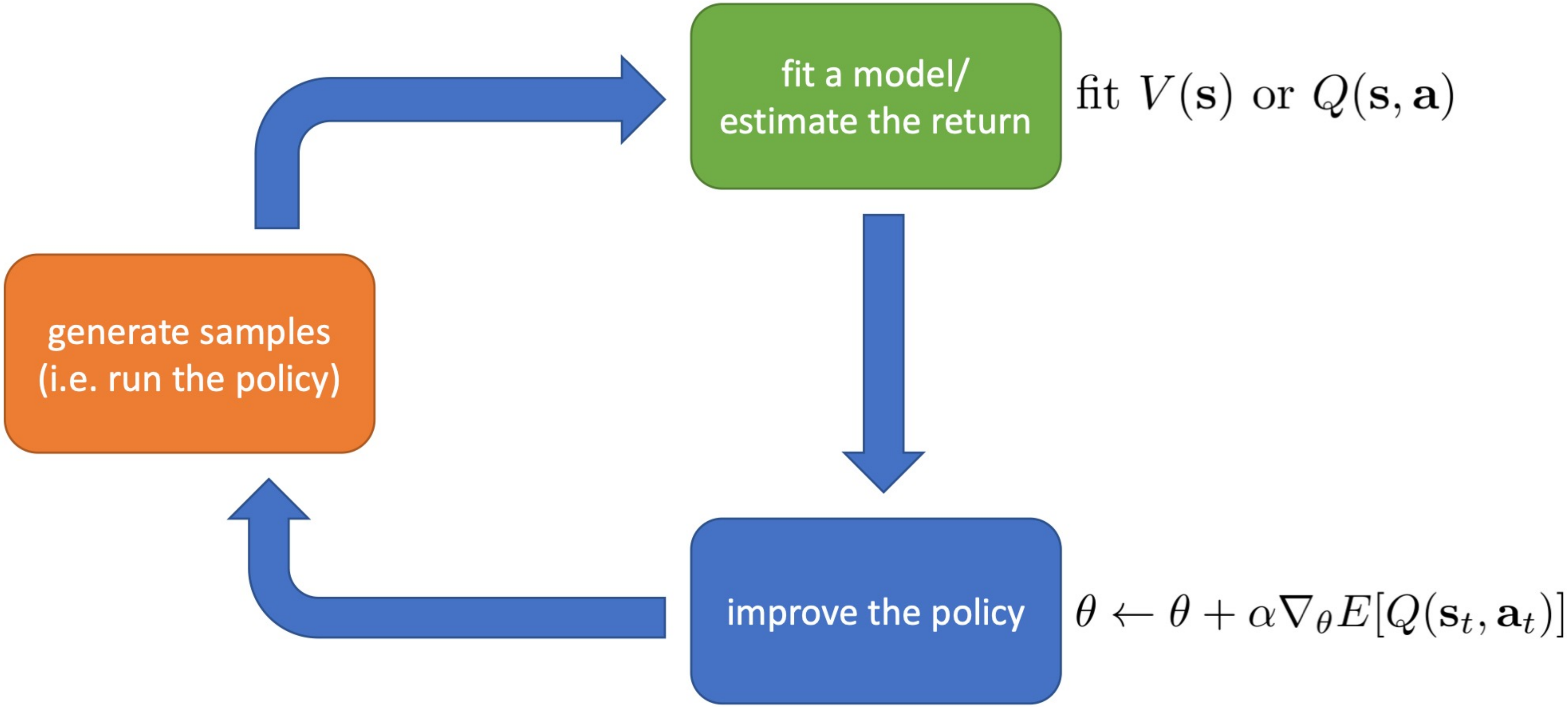
Courtesy: CS 285 course, Berkeley

Direct Policy Gradient



Courtesy: CS 285 course, Berkeley

Actor-critic: value functions + policy gradients



Courtesy: CS 285 course, Berkeley

Where do rewards come form?

- An expert gives us the reward
- Learning from demonstrations
  - Directly copying observed behavior
  - Inferring rewards from observed behavior (inverse reinforcement learning)

Motivation (cont.)

	AI Planning	SL	UL	RL	IL
Optimization	X			X	X
Learns from experience		X	X	X	X
Generalization	X	X	X	X	X
Delayed Consequences	X			X	X
Exploration				X	

- SL = supervised learning; UL = unsupervised learning; RL = reinforcement learning; IL = imitation learning
- Imitation learning typically assumes input demonstrations of good policies
- IL reduces RL to SL. IL + RL is promising area

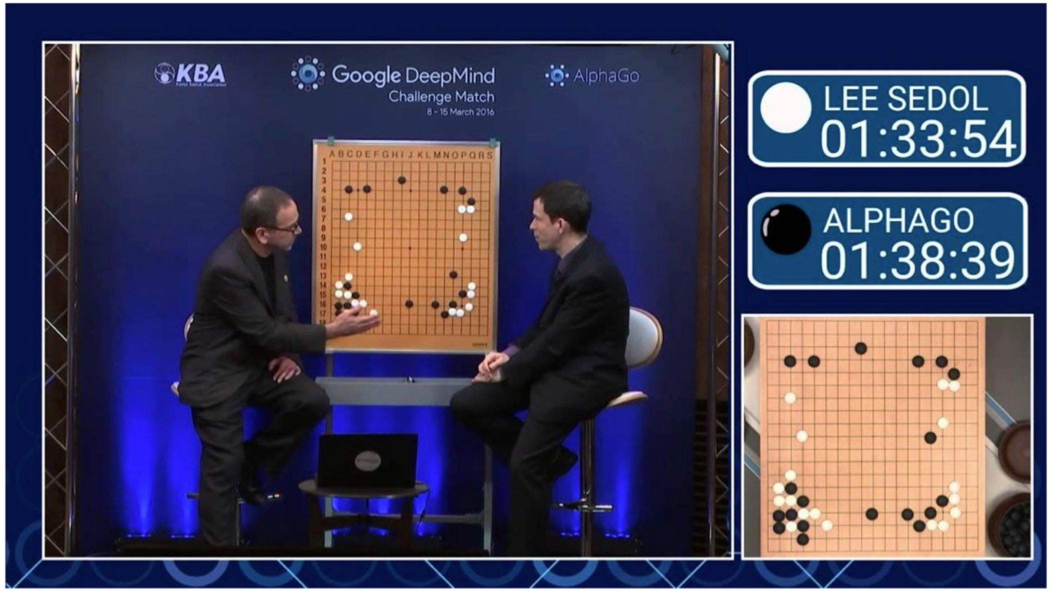
Courtesy: CS 234 course, Stanford

Planning vs learning

- Two fundamental problems in sequential decision making
  - Reinforcement learning:
    - The environment is initially unknown
    - The agent interacts with the environment
    - The agent improves its policy
  - Planning:
    - A model of the environment is known
    - The agent performs computations with its model (without any external interaction)
    - The agent improves its policy
    - a.k.a. deliberation, reasoning, introspection, pondering, thought, search

Why should we study deep reinforcement learning?

Impressive because no person had thought of it!



"Move 37" in Lee Sedol AlphaGo match: reinforcement learning "discovers" a move that surprises everyone

Impressive because it looks like something a person might draw!

vibrant portrait painting of Salvador Dalí with a robotic half face

a shiba inu wearing a beret and black turtleneck

a close up of a handpalm with leaves growing from it

an espresso machine that makes coffee from human souls, artstation

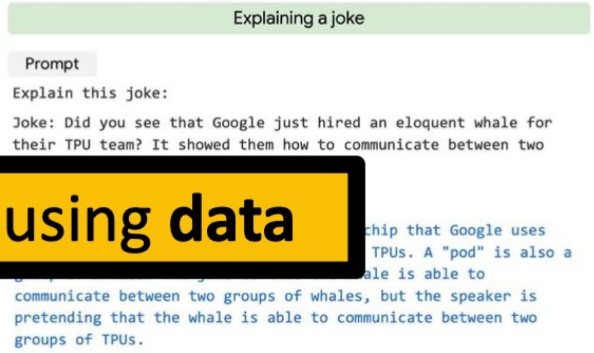
panda mad scientist mixing sparkling chemicals, artstation

a corgi's head depicted as an explosion of a nebula

Courtesy: CS 285 course, Berkeley

Data-driven AI vs. RL

Data-Driven AI

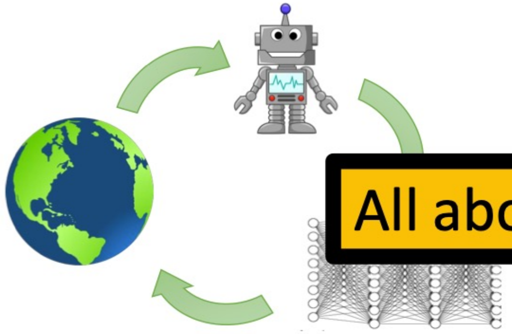


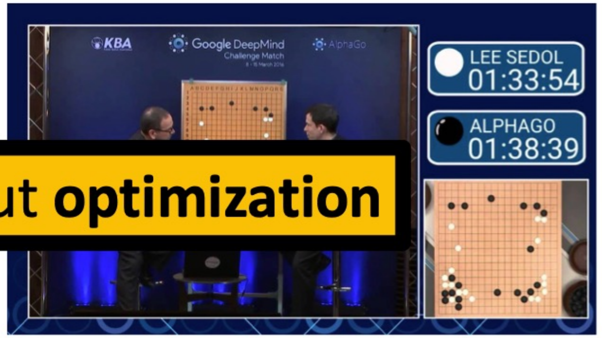
All about using **data**

+ learns about the real world from data

- doesn't try to do **better** than the data

Reinforcement Learning





All about **optimization**

+ optimizes a goal with emergent behavior

- but need to figure out how to use at scale!

Data without **optimization** doesn't allow us to solve new problems in new ways

Courtesy: CS 285 course, Berkeley

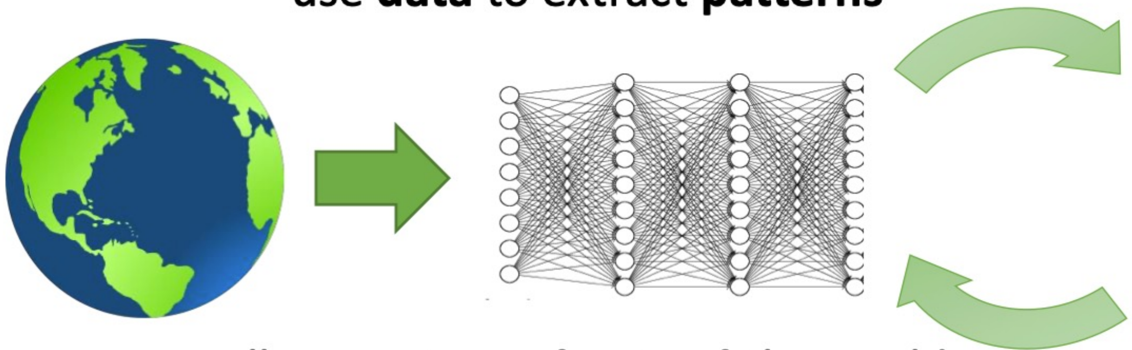
A Bitter Lesson (Richard Sutton)

"We have to learn the bitter lesson that building in how we think we think does not work in the long run. The two methods that seem to scale arbitrarily … are learning and search

http://www.incompleteideas.net/IncIdeas/BitterLesson.html

Learning

use **data** to extract **patterns**



allows us to **understand** the world

Search

use **computation** to extract **inferences**

optimization

some optimization process that uses (typically iterative) computation to make rational decisions

leverages that **understanding** for **emergence**

Data without **optimization** doesn't allow us to solve new problems in new ways

Optimization without **data** is hard to apply to the real world outside of simulators

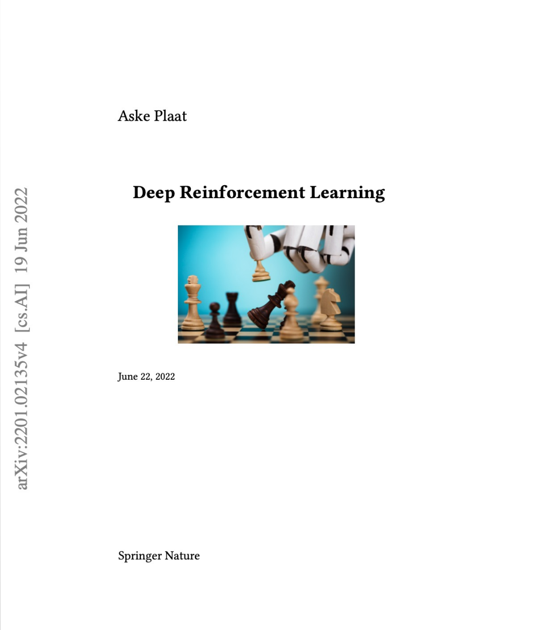
Courtesy: CS 285 course, Berkeley

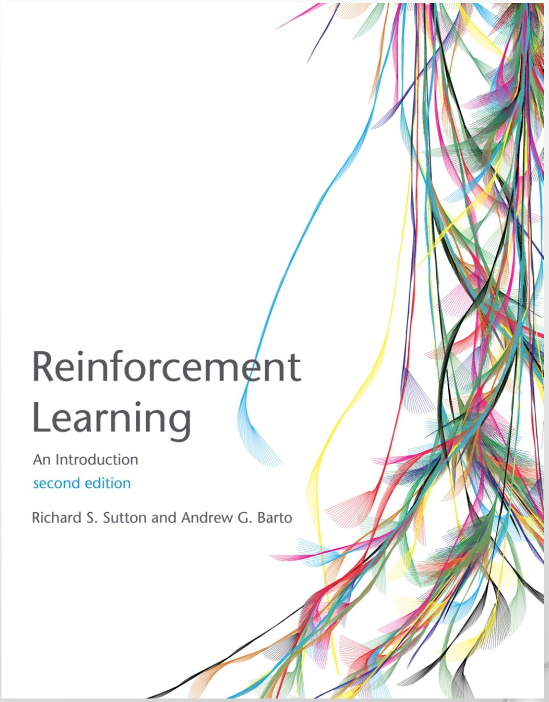
Superintelligence

- The models are trained based on human annotations and preferences.
- Can they get smarter than humans?

References

- Reinforcement Learning: An Introduction by R. Sutton and A. Barto, 2nd Edition, 2020.
- Deep Reinforcement Learning by A. Plaat, 2022.
- Original papers of some methods.





Teaching Assistants

- Arash Alikhani (Head TA)
- Soroush Vafaei Tabar
- Amirmohammad Izadi

Prereqs.

- Stochastic Processes (Prob. And Stats, Markov Processes, Estimation Theory, Information Theory)
- Optimization (Lagrange Multipliers)
- Deep Learning (Concepts and Pytorch)