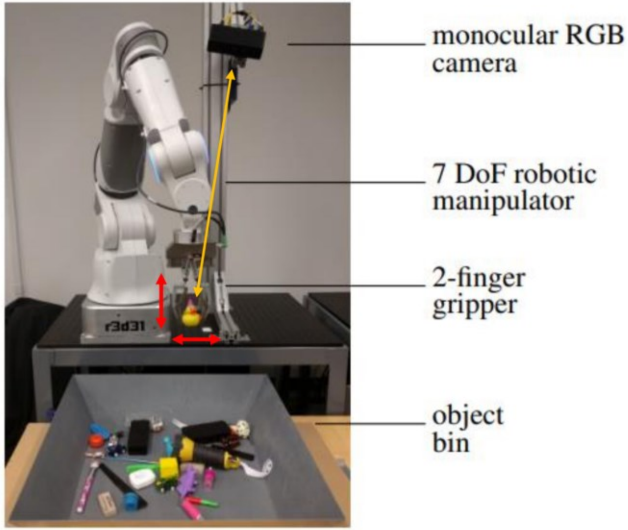
KTH VETENSKAP OCH KONST

KTH ROYAL INSTITUTE OF TECHNOLOGY

# Reinforcement learning

## Introduction I

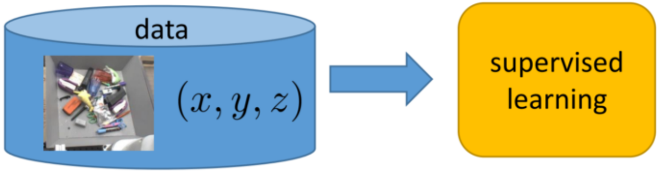
# Intuition to RL




**Task:**

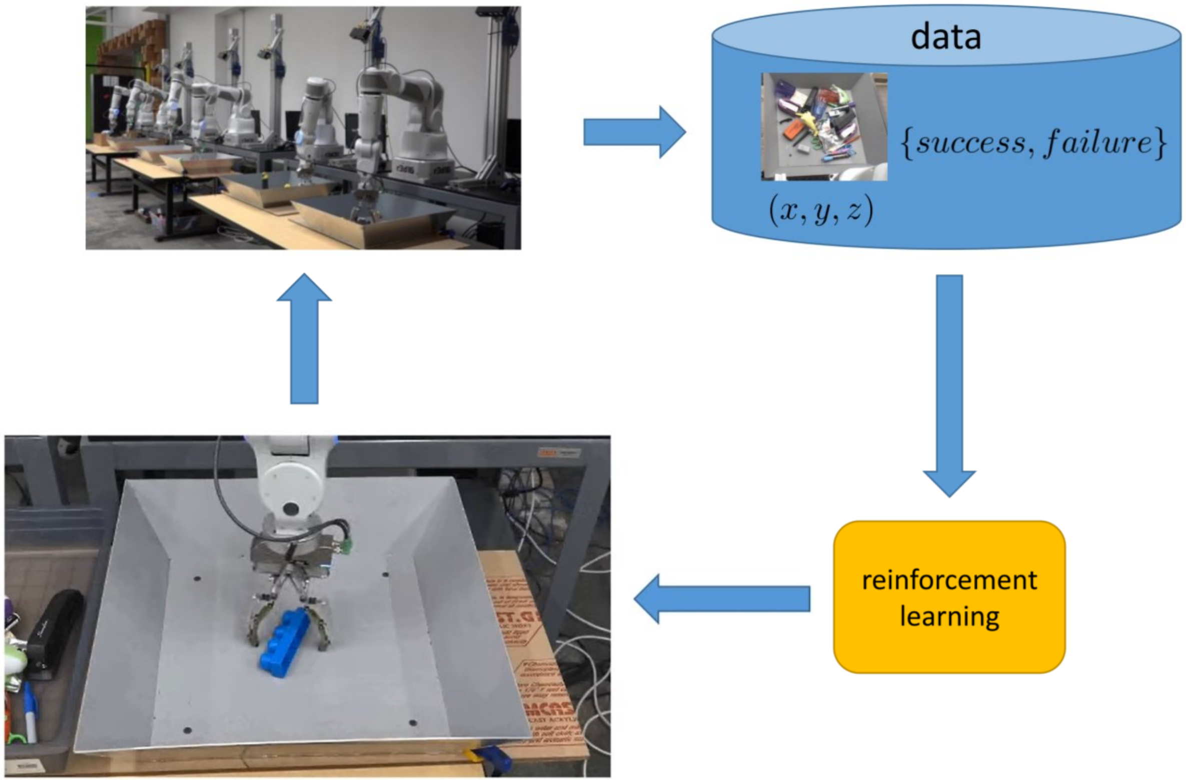

Option 1: understand and solve

Option 2: solve a ML problem



# Learning from experience

# Learning from experience





This lecture is partially based on the materials of the course CS 285 by Sergey Levine, University of California, Berkeley

# What is it?

Approach for learning decision making and control from experience.

# RL vs supervised learning

## Standard (supervised) machine learning:

given $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$

learn to predict $y$ from $\mathbf{x}$ $\qquad f(\mathbf{x}) \approx y$

## Usually assumes:

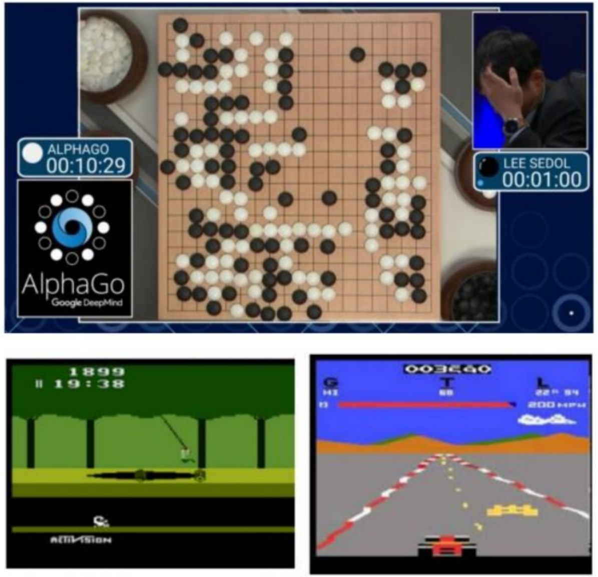
- i.i.d. data
- known ground truth outputs in training

## Reinforcement learning:

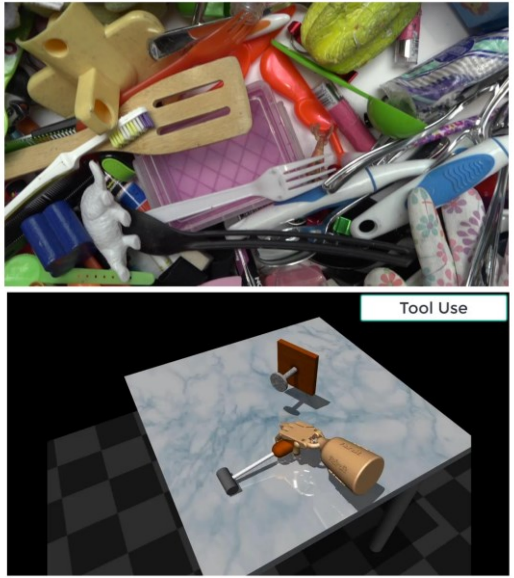
- Data is **not** i.i.d.: previous outputs influence future inputs!
- Ground truth answer is not known, only know if we succeeded or failed
  - more generally, we know the reward

# Applications

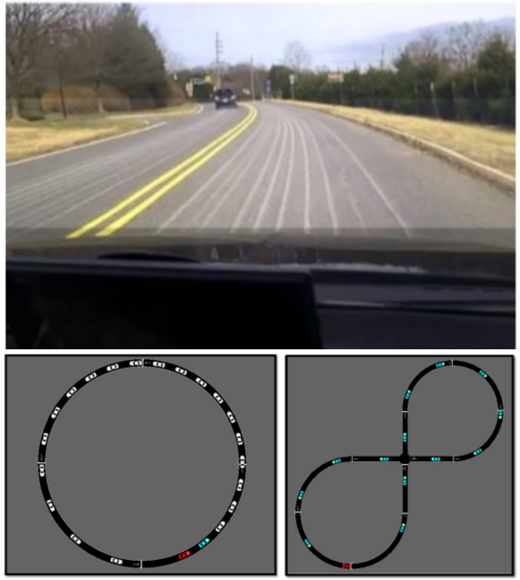
*Games*




*Robotics*




*Navigation*

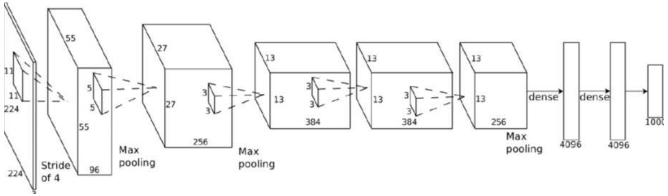


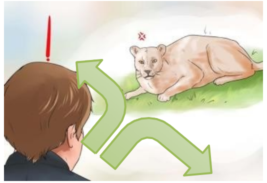
# Brief Introduction to Reinforcement Learning

**-- definitions and problem formulation**

Some materials of the course are from http://rail.eecs.berkeley.edu/deeprlcourse/

# Terminology & notation



| $\mathbf{o}_t$ | $\pi_\theta(\mathbf{a}_t \vert \mathbf{o}_t)$ | $\mathbf{a}_t$ |
|---|---|---|

$\mathbf{s}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{a}_t$ – action

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)

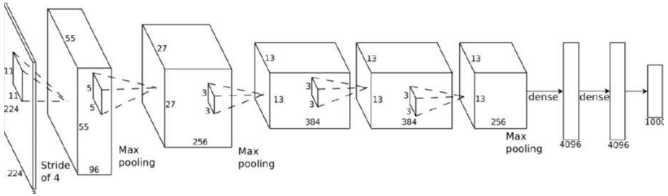
$\mathbf{o}_t$ – observation

$\mathbf{s}_t$ – state

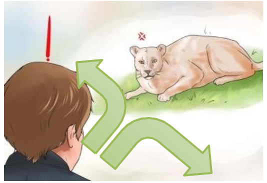

This lecture is partially based on the materials of the course CS 285 by Sergey Levine, University of California, Berkeley

# Terminology & notation

$\mathbf{o}_t$



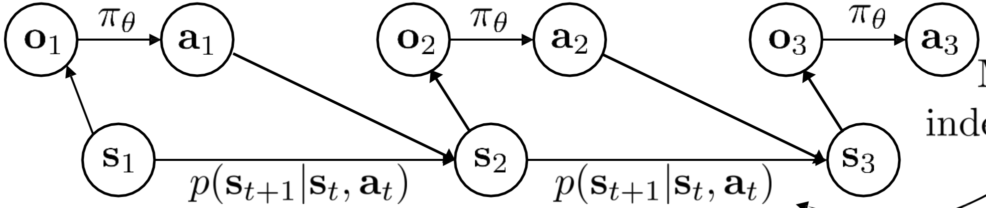

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$



$\mathbf{a}_t$

$\mathbf{s}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{a}_t$ – action

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)

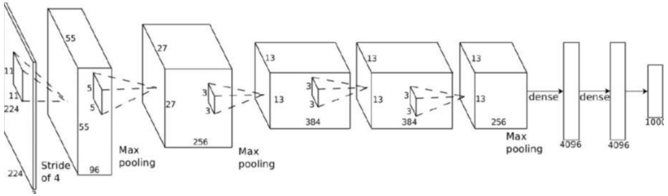



Markov property
independent of $\mathbf{s}_{t-1}$


This lecture is partially based on the materials of the course CS 285 by Sergey Levine, University of California, Berkeley

# Reward functions

$\mathbf{o}_t$




$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

$\mathbf{a}_t$

which action is better or worse?

$r(\mathbf{s}, \mathbf{a})$: reward function

tells us which states and actions are better

$\mathbf{s}$, $\mathbf{a}$, $r(\mathbf{s}, \mathbf{a})$, and $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ define Markov decision process

high reward

low reward

# Definitions: Markov chain

$$\mathcal{M} = \{\mathcal{S}, \mathcal{T}\}$$

$\mathcal{S}$ – state space — states $s \in \mathcal{S}$ (discrete or continuous)

$\mathcal{T}$ – transition operator — $p(s_{t+1}|s_t)$

let $\mu_{t,i} = p(s_t = i)$ — $\vec{\mu}_t$ is a vector of probabilities

let $\mathcal{T}_{i,j} = p(s_{t+1} = i | s_t = j)$ — then $\vec{\mu}_{t+1} = \mathcal{T}\vec{\mu}_t$

$$\mathbf{s}_1 \xrightarrow{\;p(\mathbf{s}_{t+1}|\mathbf{s}_t)\;} \mathbf{s}_2 \xrightarrow{\;p(\mathbf{s}_{t+1}|\mathbf{s}_t)\;} \mathbf{s}_3$$

Markov property
independent of $\mathbf{s}_{t-1}$

# Stationary distributions

The *stationary distribution* of a Markov chain describes the distribution over the set of states after a sufficiently long time, such that the state distribution does not change anymore:

$$\mu = \mathcal{T}\mu$$

A stationary state distribution exists, if a Markov chain is *ergodic, i.e.,* if it is possible to eventually get from every state to every other state with positive probability, and *aperiodic*

# Definitions: Markov decision process (MDP)

$$\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, r\}$$

$\mathcal{S}$ – state space — states $s \in \mathcal{S}$ (discrete or continuous)

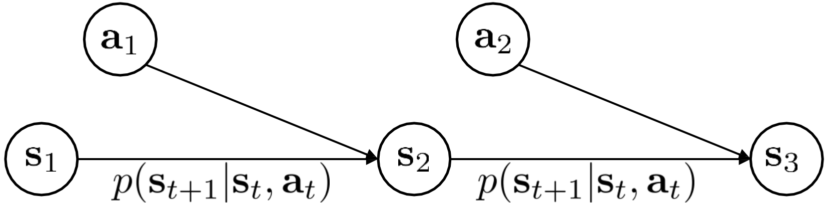
$\mathcal{A}$ – action space — actions $a \in \mathcal{A}$ (discrete or continuous)

$\mathcal{T}$ – transition operator (now a tensor!)

let $\mu_{t,j} = p(s_t = j)$

let $\xi_{t,k} = p(a_t = k)$

let $\mathcal{T}_{i,j,k} = p(s_{t+1} = i | s_t = j, a_t = k)$



# Definitions: Markov decision process (MDP)

$$\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, r\}$$

$\mathcal{S}$ – state space — states $s \in \mathcal{S}$ (discrete or continuous)

$\mathcal{A}$ – action space — actions $a \in \mathcal{A}$ (discrete or continuous)

$\mathcal{T}$ – transition operator (now a tensor!)

$r$ – reward function — $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$

$r(s_t, a_t)$ – reward

# Definitions: partially observed MDP

$\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{E}, r\}$

$\mathcal{S}$ – state space — states $s \in \mathcal{S}$ (discrete or continuous)

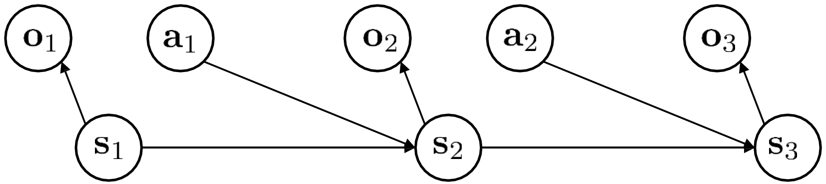
$\mathcal{A}$ – action space — actions $a \in \mathcal{A}$ (discrete or continuous)

$\mathcal{O}$ – observation space — observations $o \in \mathcal{O}$ (discrete or continuous)

$\mathcal{T}$ – transition operator (like before)

$\mathcal{E}$ – emission probability $p(o_t|s_t)$

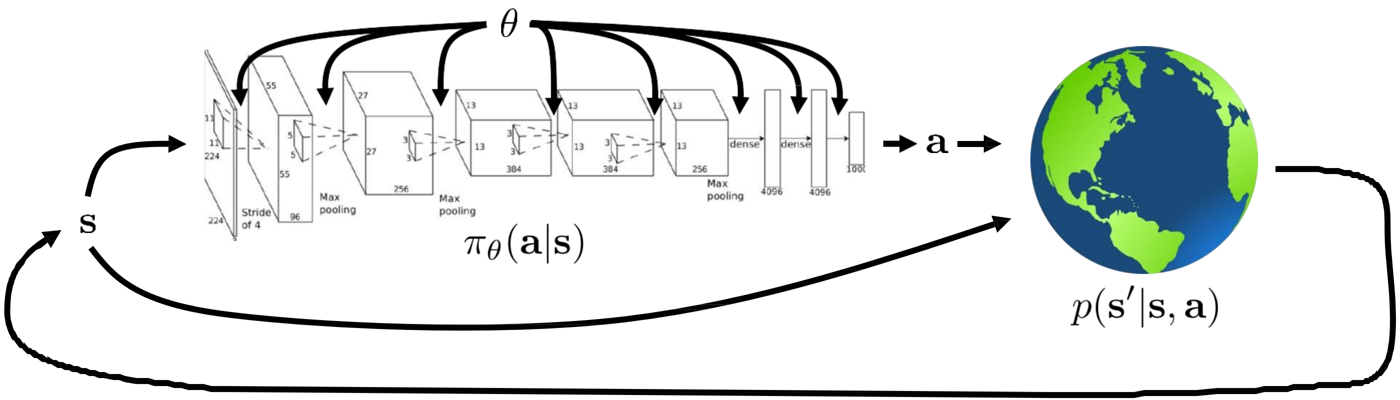
$r$ – reward function — $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$





This lecture is partially based on the materials of the course CS 285 by Sergey Levine, University of California, Berkeley

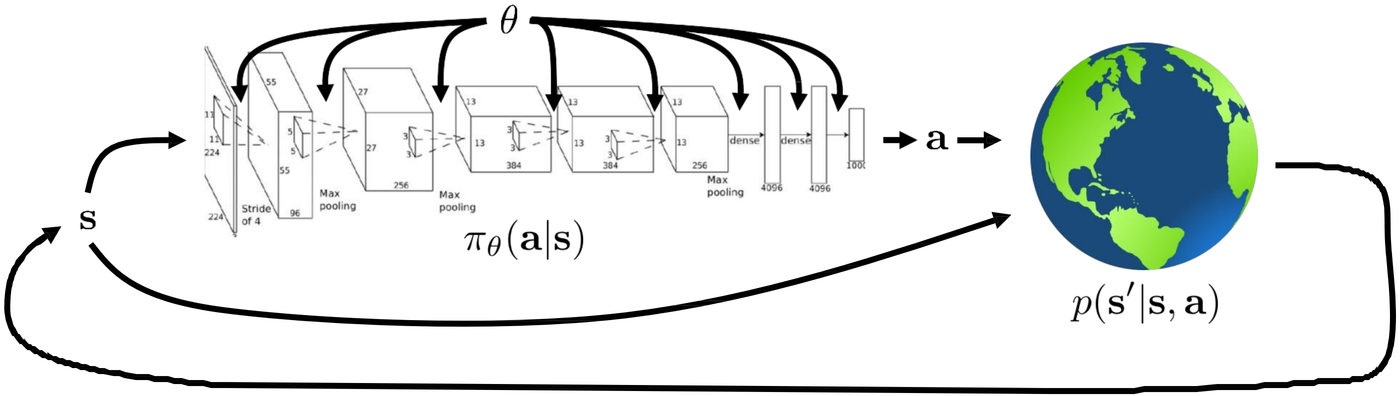
# The goal of reinforcement learning



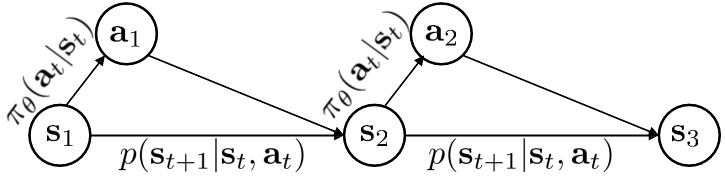

$$\underbrace{p_\theta(\mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T, \mathbf{a}_T)}_{p_\theta(\tau)} = p(\mathbf{s}_1) \prod_{t=1}^{T} \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$$

$$\theta^\star = \arg\max_\theta E_{\tau \sim p_\theta(\tau)} \left[ \sum_t r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

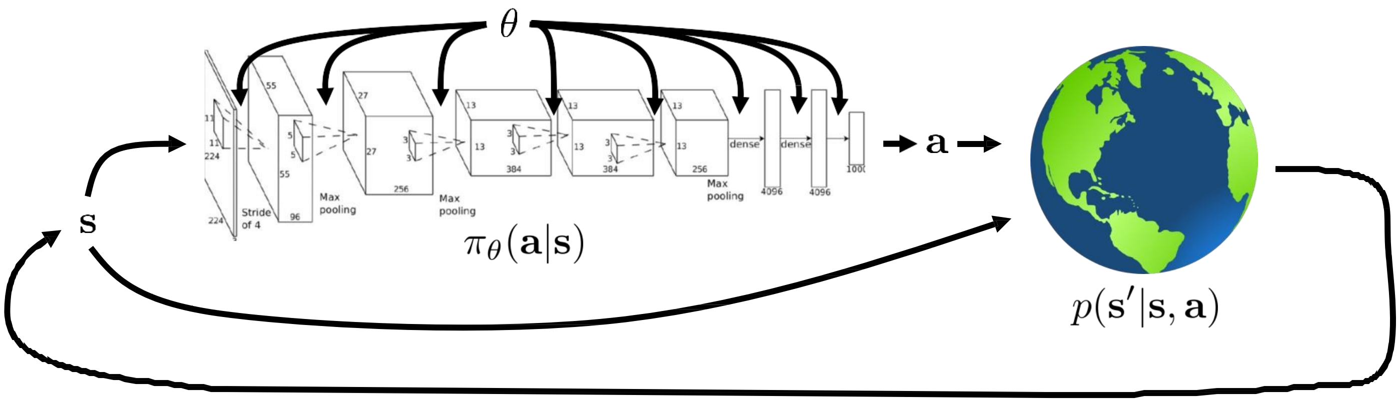
# The goal of reinforcement learning




$$\underbrace{p_\theta(\mathbf{s}_1,\mathbf{a}_1,\ldots,\mathbf{s}_T,\mathbf{a}_T)}_{p_\theta(\tau)} = p(\mathbf{s}_1)\prod_{t=1}^{T}\underbrace{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t)}_{\text{Markov chain on } (\mathbf{s},\mathbf{a})}$$

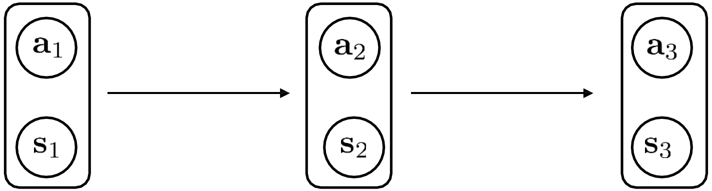


# The goal of reinforcement learning




$$\underbrace{p_\theta(\mathbf{s}_1,\mathbf{a}_1,\ldots,\mathbf{s}_T,\mathbf{a}_T)}_{p_\theta(\tau)} = p(\mathbf{s}_1)\prod_{t=1}^{T}\underbrace{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t)}_{\text{Markov chain on }(\mathbf{s},\mathbf{a})}$$

$$p((\mathbf{s}_{t+1},\mathbf{a}_{t+1})|(\mathbf{s}_t,\mathbf{a}_t)) =$$
$$p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t)\pi_\theta(\mathbf{a}_{t+1}|\mathbf{s}_{t+1})$$



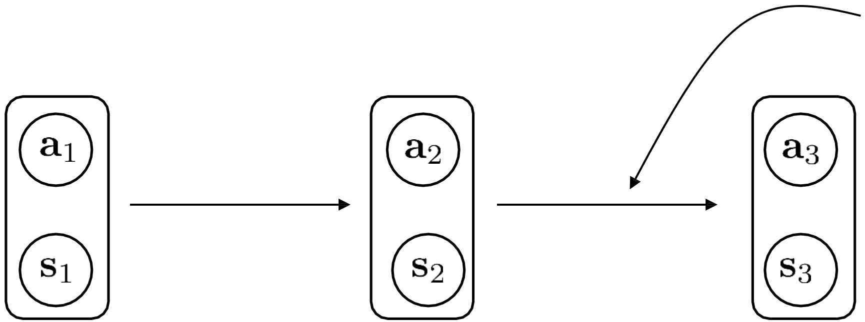
# Finite horizon case: state-action marginal

$$\theta^\star = \arg\max_\theta E_{\tau\sim p_\theta(\tau)}\left[\sum_t r(\mathbf{s}_t,\mathbf{a}_t)\right]$$

$$= \arg\max_\theta \sum_{t=1}^{T} E_{(\mathbf{s}_t,\mathbf{a}_t)\sim p_\theta(\mathbf{s}_t,\mathbf{a}_t)}[r(\mathbf{s}_t,\mathbf{a}_t)]$$

$p_\theta(\mathbf{s}_t,\mathbf{a}_t)$ state-action marginal

$$p((\mathbf{s}_{t+1},\mathbf{a}_{t+1})|(\mathbf{s}_t,\mathbf{a}_t)) =$$
$$p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{a}_t)\pi_\theta(\mathbf{a}_{t+1}|\mathbf{s}_{t+1})$$



# Infinite horizon case: stationary distribution

$$\theta^\star = \arg\max_\theta \sum_{t=1}^{T} E_{(\mathbf{s}_t,\mathbf{a}_t)\sim p_\theta(\mathbf{s}_t,\mathbf{a}_t)}[r(\mathbf{s}_t,\mathbf{a}_t)]$$

what if $T = \infty$?

does $p(\mathbf{s}_t, \mathbf{a}_t)$ converge to a *stationary* distribution?

$$\mu = \mathcal{T}\mu$$

stationary = the same before and after transition

$$(\mathcal{T} - \mathbf{I})\mu = 0$$

(always exists under some regularity conditions)

$\mu = p_\theta(\mathbf{s}, \mathbf{a})$ stationary distribution


This lecture is partially based on the materials of the course CS 285 by Sergey Levine, University of California, Berkeley

# Infinite horizon case: stationary distribution

$$\theta^\star = \arg\max_\theta \frac{1}{T}\sum_{t=1}^{T} E_{(\mathbf{s}_t,\mathbf{a}_t)\sim p_\theta(\mathbf{s}_t,\mathbf{a}_t)}[r(\mathbf{s}_t,\mathbf{a}_t)] \to E_{(\mathbf{s},\mathbf{a})\sim p_\theta(\mathbf{s},\mathbf{a})}[r(\mathbf{s},\mathbf{a})]$$

(in the limit as $T \to \infty$)

what if $T = \infty$?

does $p(\mathbf{s}_t, \mathbf{a}_t)$ converge to a *stationary* distribution?

$$\mu = \mathcal{T}\mu$$

stationary = the same before and after transition

$$(\mathcal{T} - \mathbf{I})\mu = 0$$

(always exists under some regularity conditions)

$\mu = p_\theta(\mathbf{s}, \mathbf{a})$ stationary distribution

# Value-action (Q) and value functions

## Definition: Q-function

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

## Definition: value function

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

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

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

# Using Q-functions and value functions

Idea 1: if we have policy $\pi$, and we know $Q^\pi(\mathbf{s}, \mathbf{a})$, then we can *improve* $\pi$:

set $\pi'(\mathbf{a}|\mathbf{s}) = 1$ if $\mathbf{a} = \arg\max_{\mathbf{a}} Q^\pi(\mathbf{s}, \mathbf{a})$

this policy is at least as good as $\pi$ (and probably better)!

and it doesn't matter what $\pi$ is

Idea 2: compute gradient to increase probability of good actions $\mathbf{a}$:

if $Q^\pi(\mathbf{s}, \mathbf{a}) > V^\pi(\mathbf{s})$, then $\mathbf{a}$ is *better than average* (recall that $V^\pi(\mathbf{s}) = E[Q^\pi(\mathbf{s}, \mathbf{a})]$ under $\pi(\mathbf{a}|\mathbf{s})$)

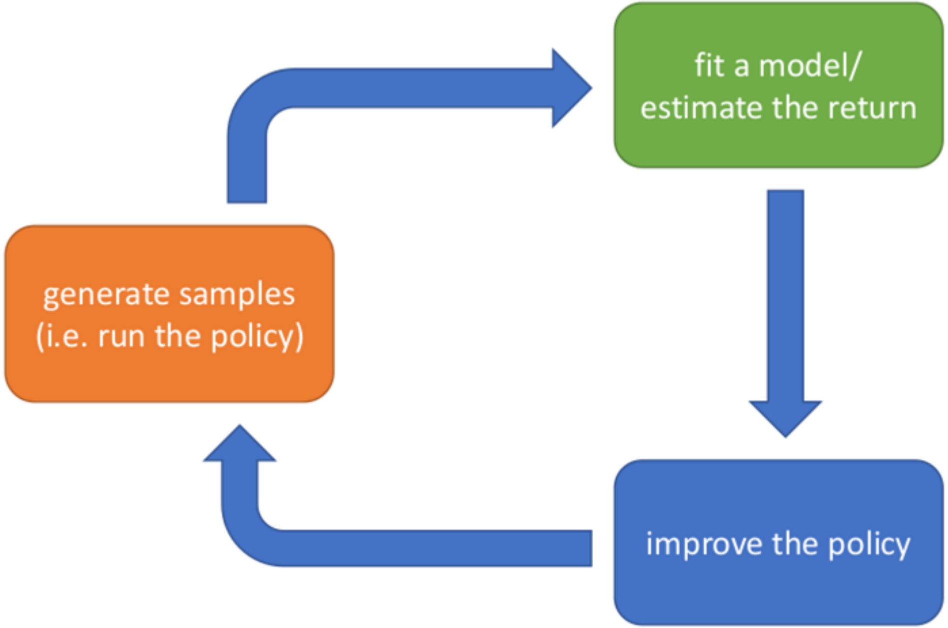
modify $\pi(\mathbf{a}|\mathbf{s})$ to increase probability of $\mathbf{a}$ if $Q^\pi(\mathbf{s}, \mathbf{a}) > V^\pi(\mathbf{s})$

**These ideas represent a base for many RL methods!**

# Overview of RL algorithms

Some materials of the course are from http://rail.eecs.berkeley.edu/deeprlcourse/

# The anatomy of a reinforcement learning algorithm

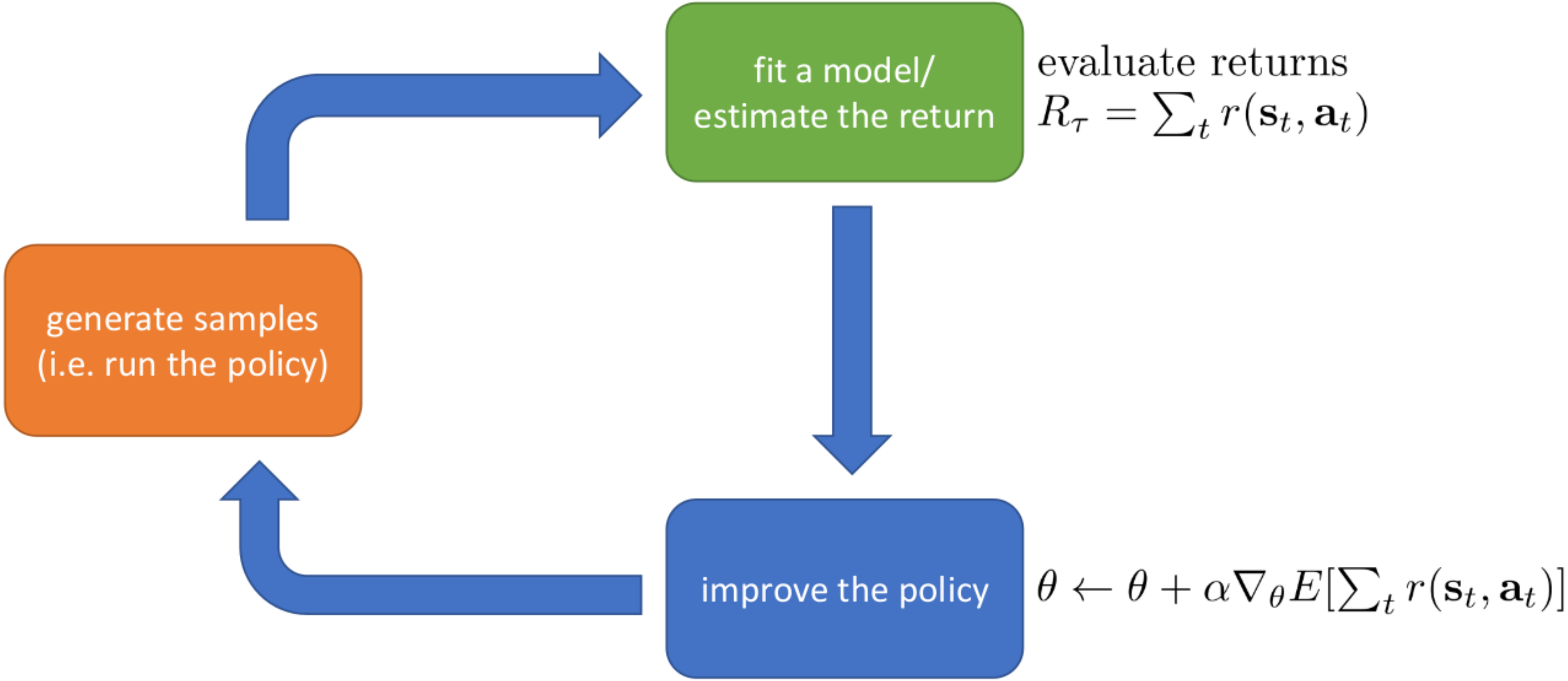


# Types of RL algorithms

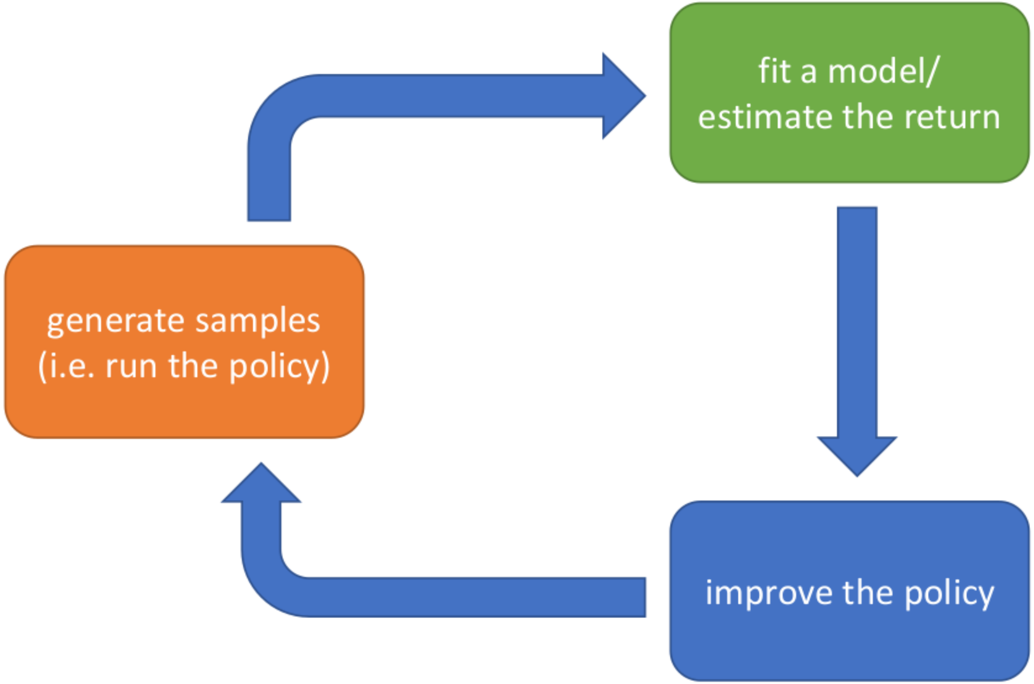
$$\theta^{\star} = \arg\max_{\theta} E_{\tau \sim p_{\theta}(\tau)} \left[ \sum_t r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

- Policy gradients: directly differentiate the above objective
- Value-based: estimate value function or Q-function of the optimal policy (no explicit policy)
- Actor-critic: estimate value function or Q-function of the current policy, use it to improve policy
- Model-based RL: estimate the transition model, and then…
  - Use it for planning (no explicit policy)
  - Use it to improve a policy
  - Something else

# Policy gradients



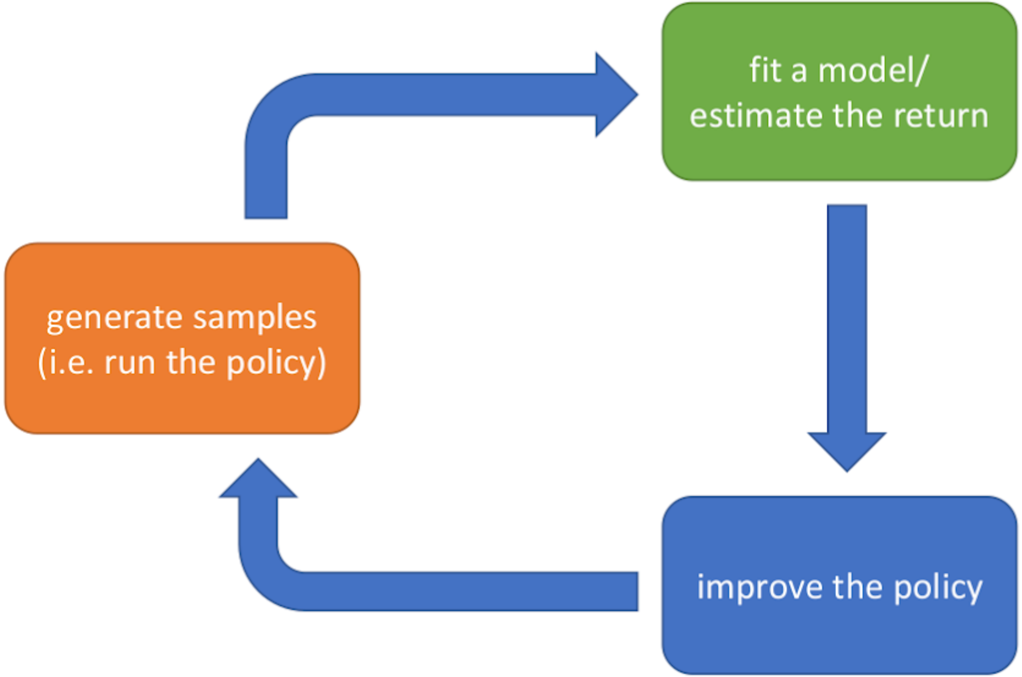
# Value function-based algorithms




fit $V(\mathbf{s})$ or $Q(\mathbf{s}, \mathbf{a})$

set $\pi(\mathbf{s}) = \arg\max_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a})$

# Actor-critic algorithms

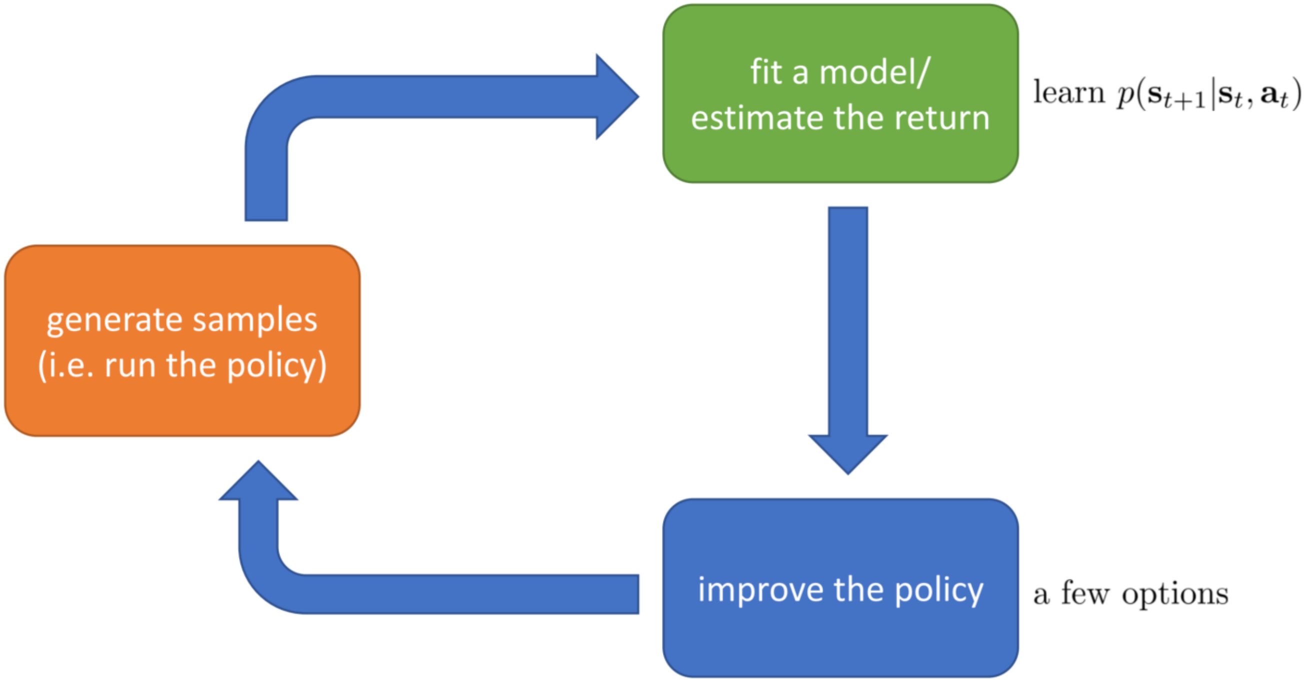



fit $V(\mathbf{s})$ or $Q(\mathbf{s}, \mathbf{a})$

evaluate returns using $V$ or $Q$!

$\theta \leftarrow \theta + \alpha \nabla_\theta E[\sum_t r(\mathbf{s}_t, \mathbf{a}_t)]$

# Model-based RL algorithms

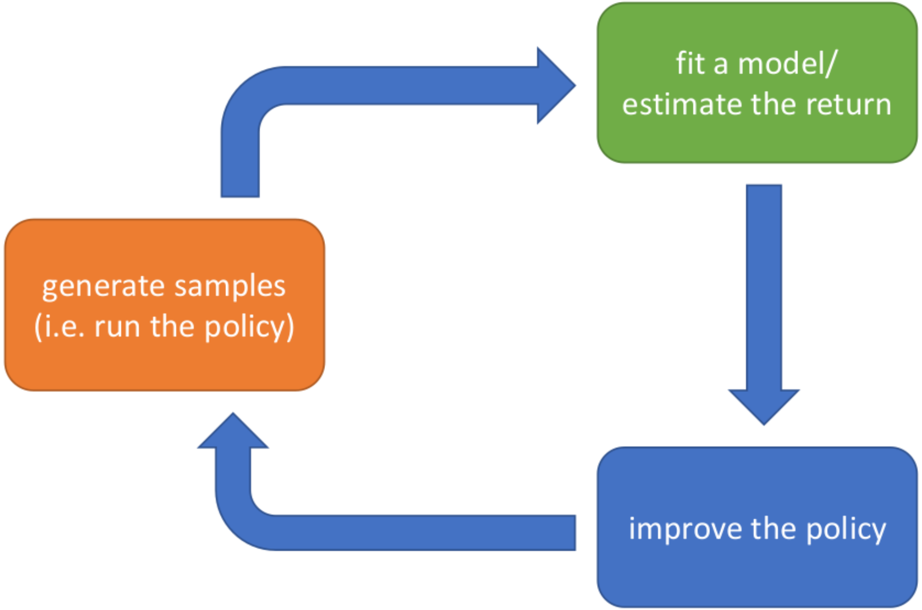




This lecture is partially based on the materials of the course CS 285 by Sergey Levine, University of California, Berkeley

# Why different algorithms?

- Different tradeoffs
  - Sample efficiency
  - Stability & ease of use
- Different assumptions
  - Stochastic or deterministic?
  - Continuous or discrete?
  - Episodic or infinite horizon?
- Different things are easy or hard in different settings
  - Easier to represent the policy?
  - Easier to represent the model?



# Sample efficiency

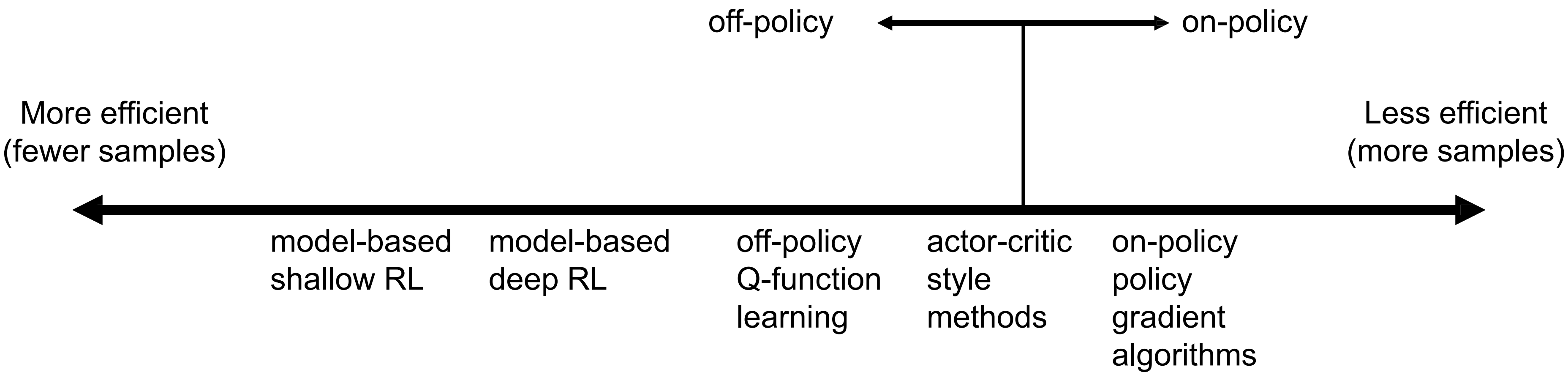
*How many samples do we need to get a good policy?*

generate samples
(i.e. run the policy)

- **Off** policy: able to improve the policy without generating new samples from that policy (using old samples)

- **On** policy: each time the policy is changed, even as a gradient step, we need to generate new samples

# Sample efficiency




Q: Why would we use a *less* efficient algorithm?

A: Wall clock time is not the same as efficiency – simulation can be efficient!

# Stability – not what you expect

- Does it converge?
- And if it converges, to what?
- And does it converge every time?

**Supervised Learning vs Reinforcement Learning**

- Supervised learning: almost *always* gradient descent
- Reinforcement learning: often *not* gradient descent
  - Q-learning: fixed point iteration
  - Model-based RL: model is not optimized for expected reward
  - Policy gradient: *is* gradient ascent, but often the least efficient!

# Stability – not what you expect

- Model-based RL
  - Model minimizes fitting error
  - Hard for large state space

- Value function fitting
  - At best, minimizes error of fit (“Bellman error”)
    - Not the same as expected reward
  - At worst, doesn’t optimize anything
    - Many popular deep RL value fitting algorithms are not guaranteed to converge to *anything* with nonlinearly parameterized function approximators (DNN)

- Policy gradient
  - directly performs gradient ascent on the true objective

# Assumptions

- Common assumption #1: full observability
  - Generally assumed by value function fitting methods
- Common assumption #2: episodic vs lifelong 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

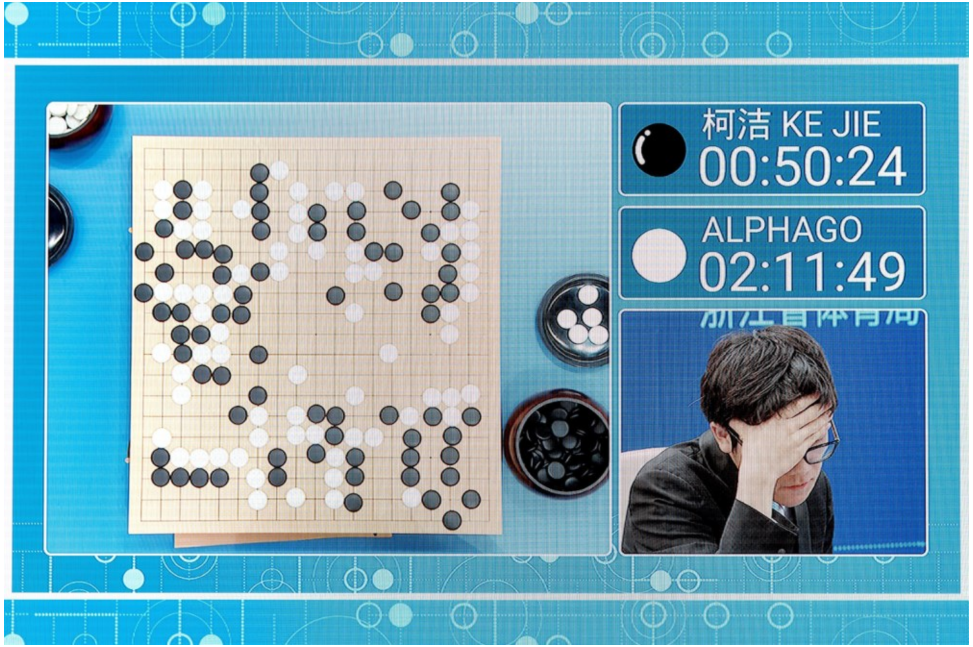
# Examples

Example 1: full observability, “continuous” state, discrete action, cheap to interact


https://deepmind.com/research/publications/playing-atari-deep-reinforcement-learning

# Examples

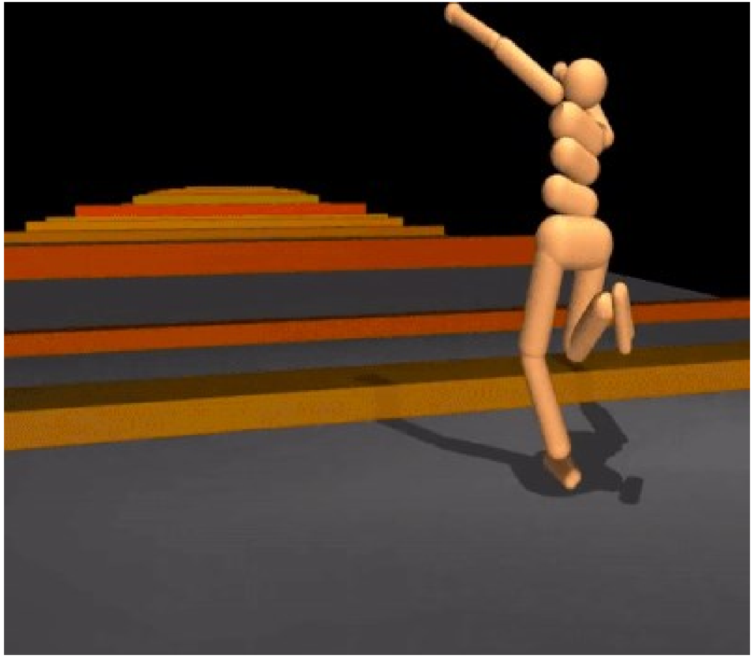
Example 2: full observability, "discrete" state, discrete action, cheap to interact



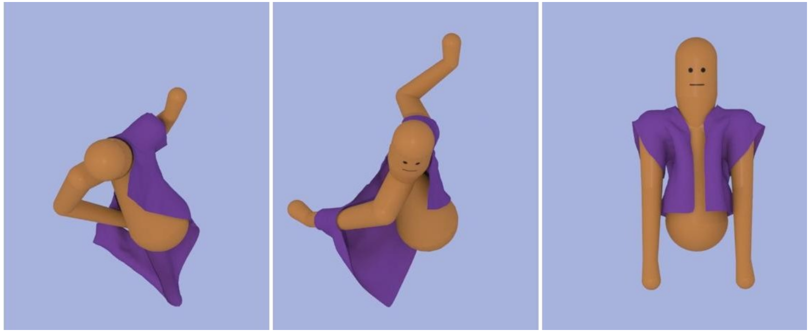

Source:Wired, Business 05.24.17

# Examples

Example 3: partial observability, continuous state, continuous action, “cheap” to interact



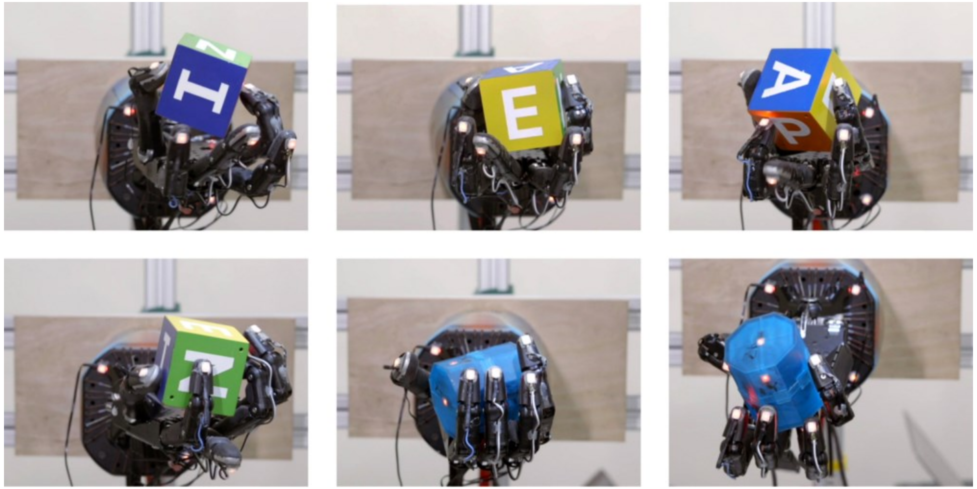
https://deepmind.com/blog/article/producing-flexible-behaviours-simulated-environments



Clegg et al. ToG 2018

# Examples

Example 4: full/partial observability, continuous state, continuous action, expensive to interact



https://openai.com/blog/learning-dexterity/


Stanford Stanley, DARPA Challenge