Lecture 25: Embodied vision

Today

- Formalisms for intelligent agents (environment, state, action, policy)
- Imitation learning
- Reinforcement learning
  - Markov Decision Processes
  - Policy gradient algorithm
- Just a high-level overview. See Sutton & Barto book [http://incompleteideas.net/book/RLbook2018.pdf] for much more complete treatment

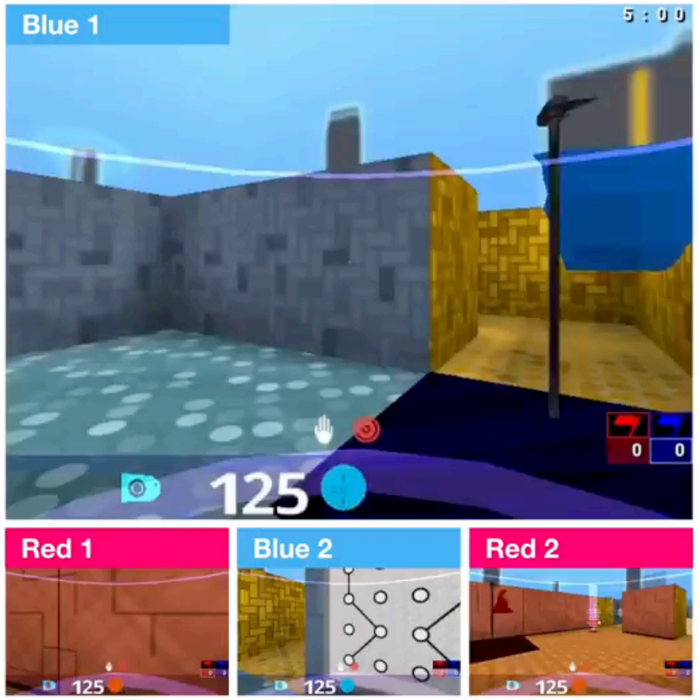
Source: Isola, Torralba, Freeman

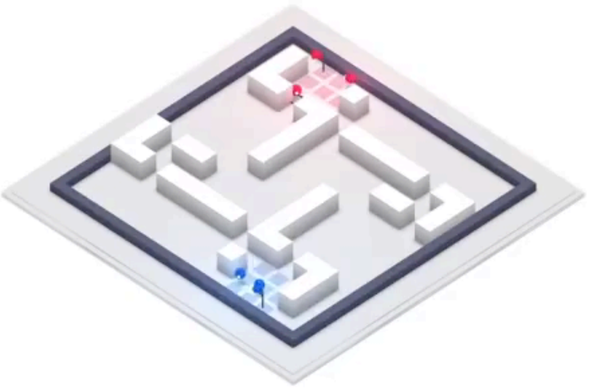
Announcements

- Sign up for final presentation timeslot by tonight!
- Email us ASAP if no time works for your group

[Silver et al., 2016]

Agent observation raw pixels





Indoor map overview

[Jaderberg et al. 2018]

Source: Isola, Torralba, Freeman

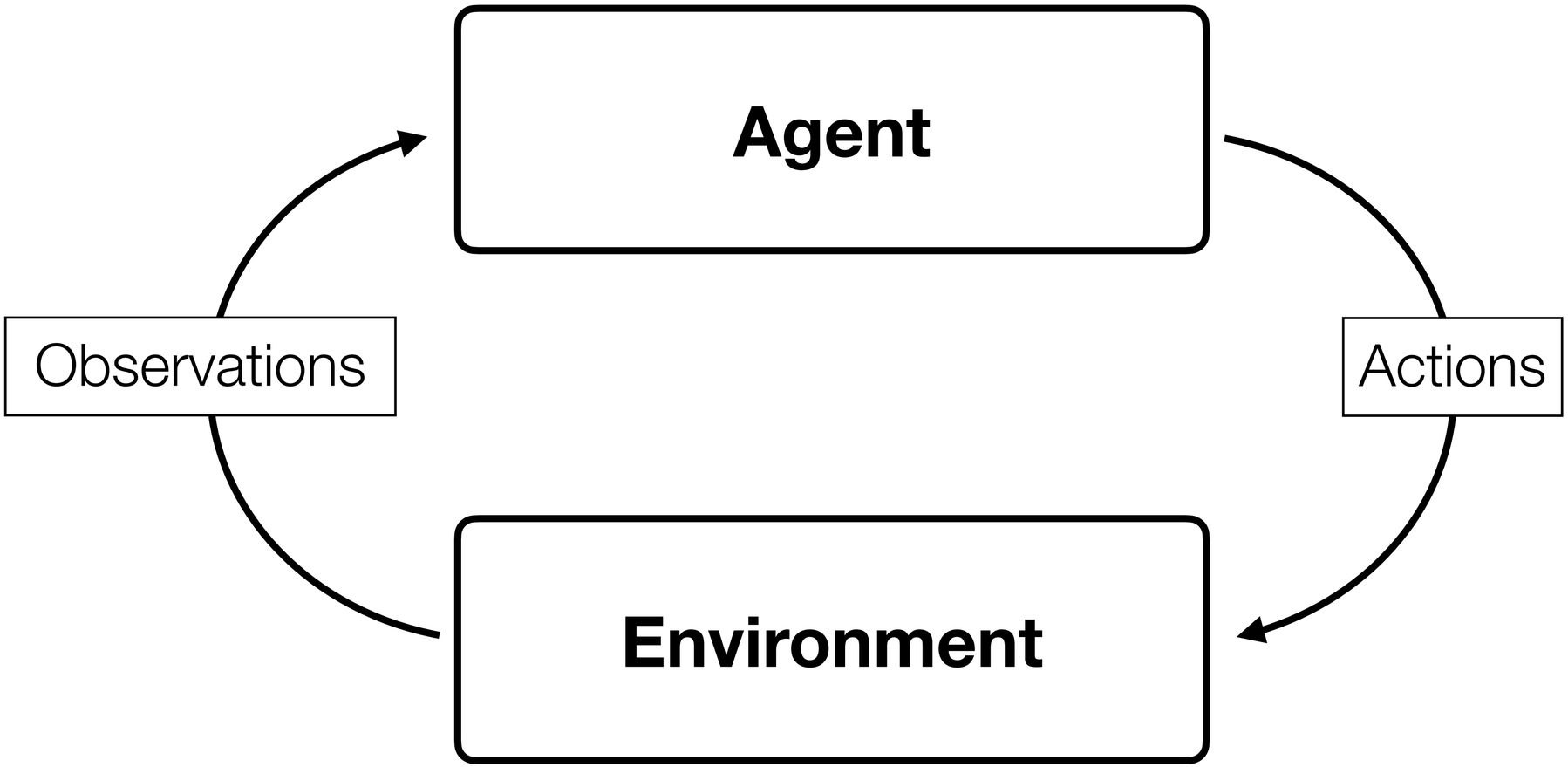
The whole purpose of visual perception, in humans,
is to make good motor decisions.

“We move in order to see and we see in order to move” — J. J. Gibson

We are **sensorimotor** systems.

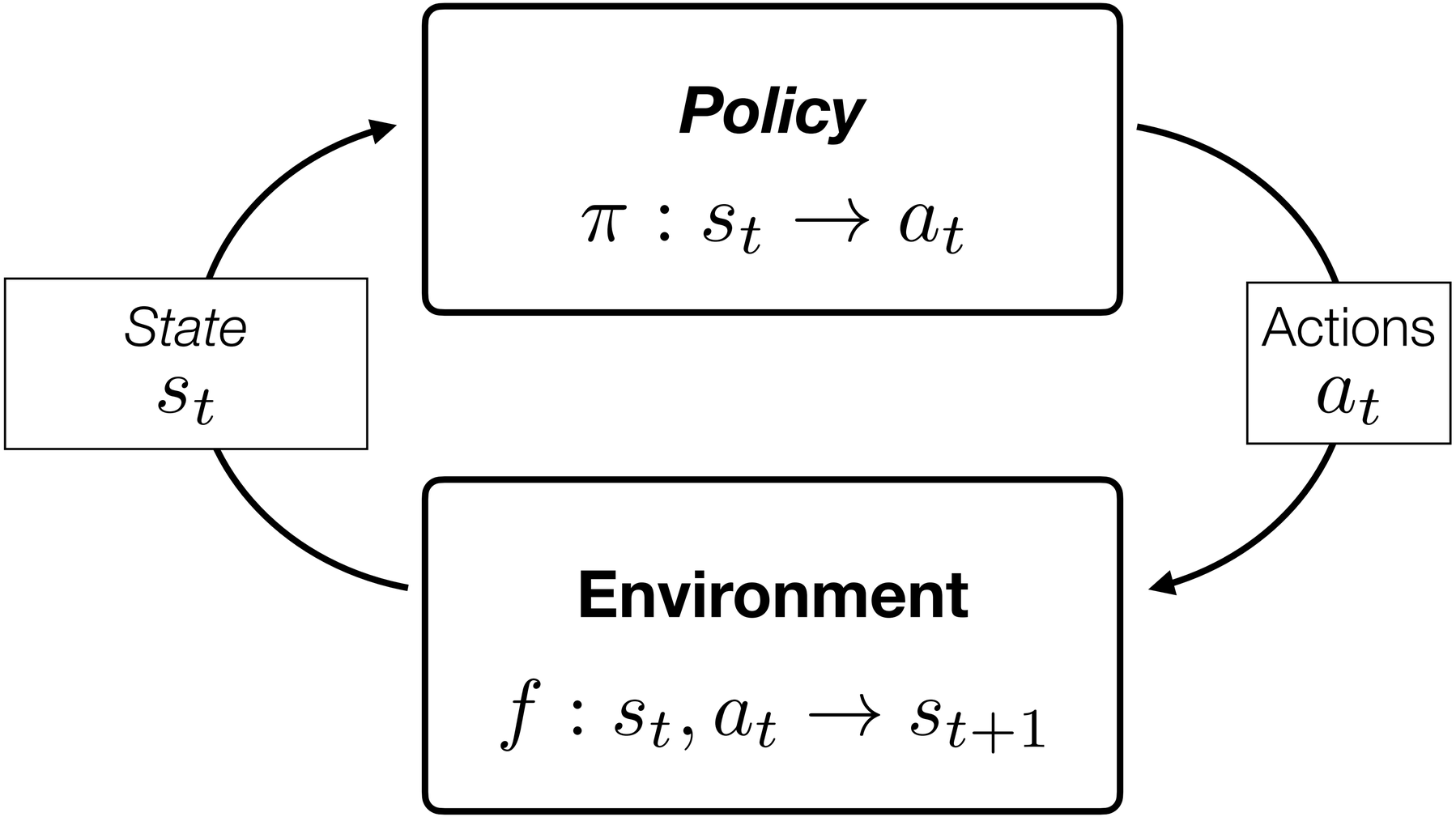
Source: Isola, Torralba, Freeman

Intelligent agents



Source: Isola, Torralba, Freeman

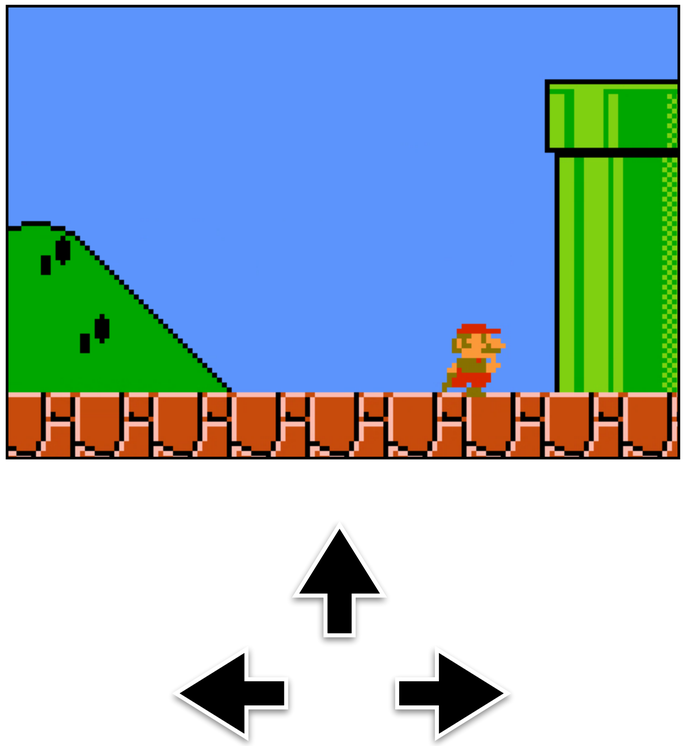
Intelligent agents



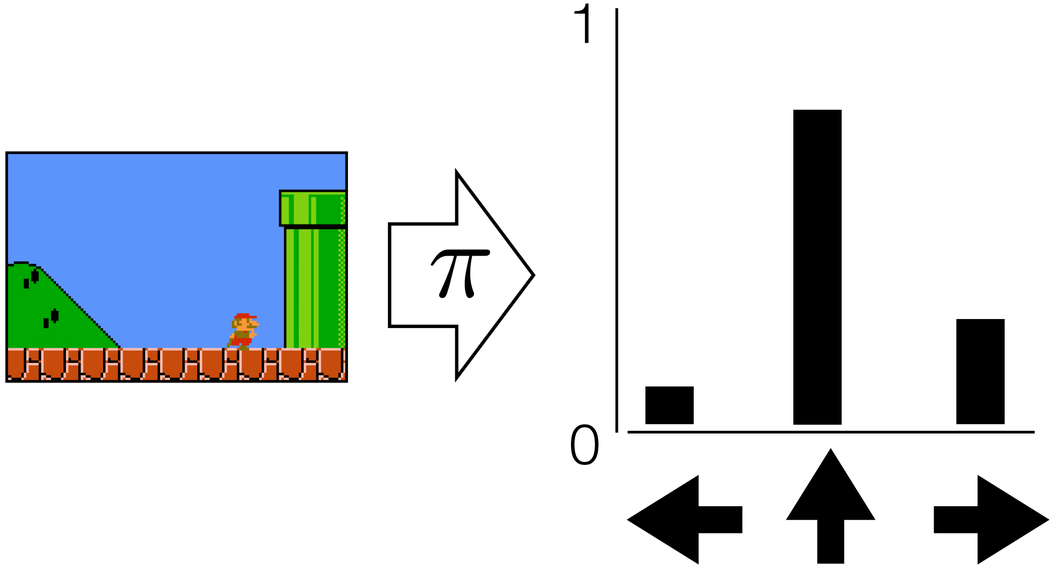
Source: Isola, Torralba, Freeman

How to represent a state? How to represent policy?

state: pixels!



policy: action classifier



Source: Isola, Torralba, Freeman

Learning from examples

(aka **supervised learning**)

Training data

$$\{x_1, y_1\}$$
$$\{x_2, y_2\}$$
$$\{x_3, y_3\}$$
$$\ldots$$

$\rightarrow$ Learner $\rightarrow$ $f : X \rightarrow Y$

$$f^* = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} \mathcal{L}(f(x_i), y_i)$$

Source: Isola, Torralba, Freeman

Imitation learning

(still just **supervised learning**, applied to learn *policies*)

Training data

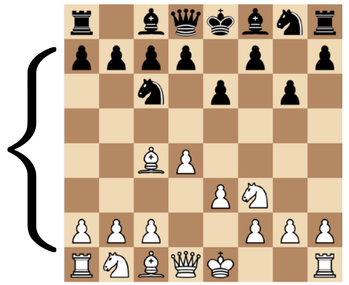
$$\{s_1, a_1\}$$
$$\{s_2, a_2\} \quad \rightarrow \quad \boxed{\text{Learner}} \quad \rightarrow \quad \pi : s \rightarrow a$$
$$\{s_3, a_3\}$$
$$\ldots$$

$$\pi^* = \arg\min_{\pi \in \Pi} \sum_{i=1}^{N} \mathcal{L}(\pi(s_i), a_i)$$

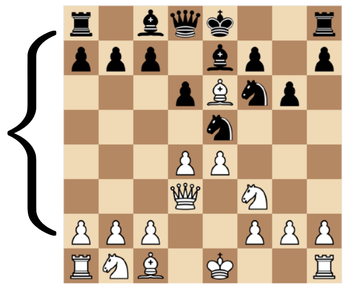
Source: Isola, Torralba, Freeman

Imitation learning

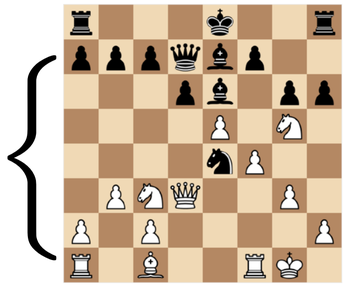
s a



d3 ♗b4+



♘f3 ♞c6



0-0 ♗xc3

⋮

→

Learner

Objective

$$\pi(s) = \texttt{softmax}(g_\theta(s))$$

$$\mathcal{L}(a, \pi(s)) = H(a, \pi(s))$$

Hypothesis space

Convolutional neural net

Optimizer

Stochastic gradient descent

→ π

Source: Isola, Torralba, Freeman

End-to-end Driving via Conditional Imitation Learning

Felipe Codevilla, Matthias Mueller, Alexey Dosovitskiy, Antonio Lopez, Vladlen Koltun

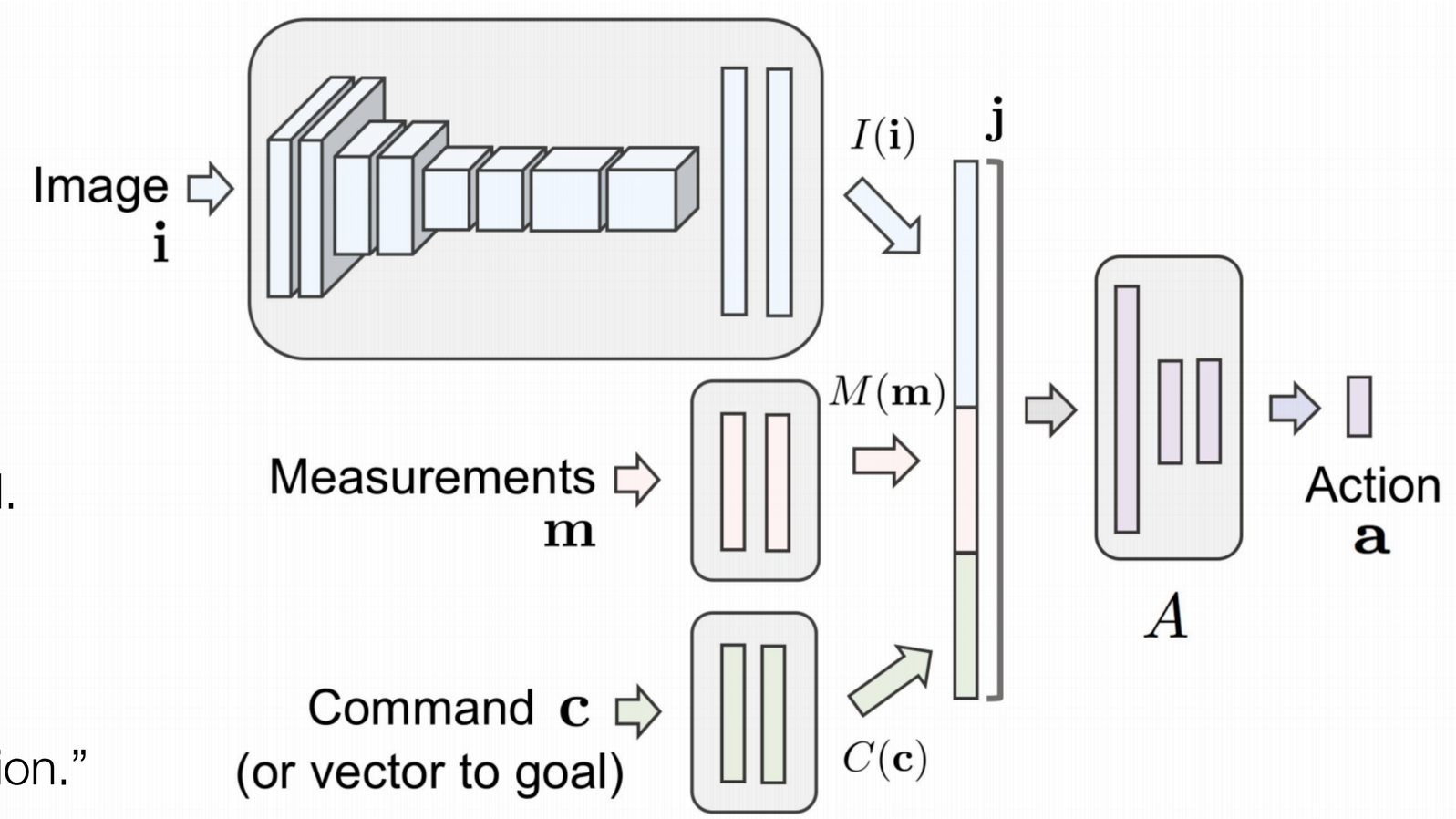
Submitted to ICRA 2018

We train and evaluate robotic vehicles in the physical world (top) and in simulated urban environments (bottom)

From images to actions

Auxiliary measurements, e.g. speed.

Goal capturing expert's intentions,
e.g. "Turn right at the next intersection."



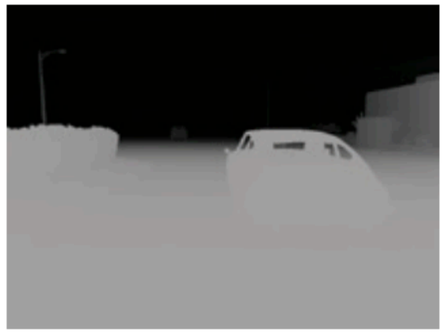
[See Codevilla et al., "End-to-end driving via conditional imitation learning", 2018]

From images to actions

Segmentation

Albedo

Depth

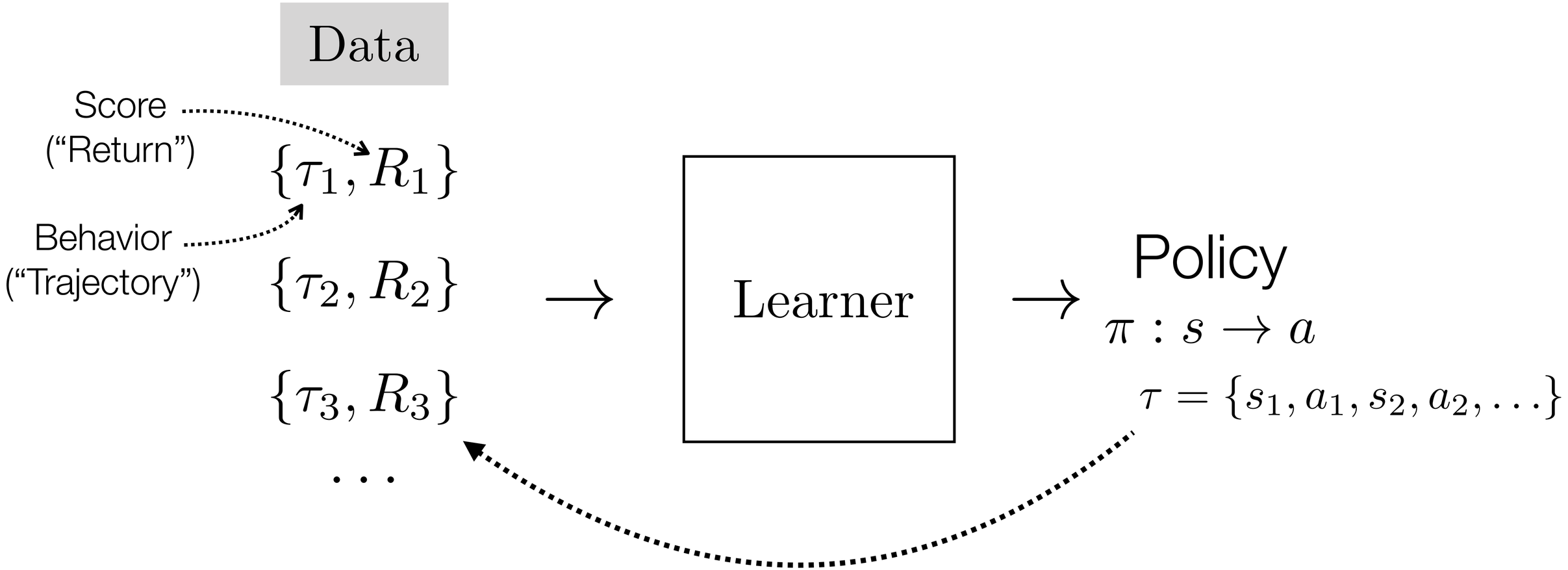


Optical flow

- Can use mid-level representations like depth, motion.
- Or do transfer learning from pretained net

[See Zhou et al., “Does computer vision matter for action?”, 2019]

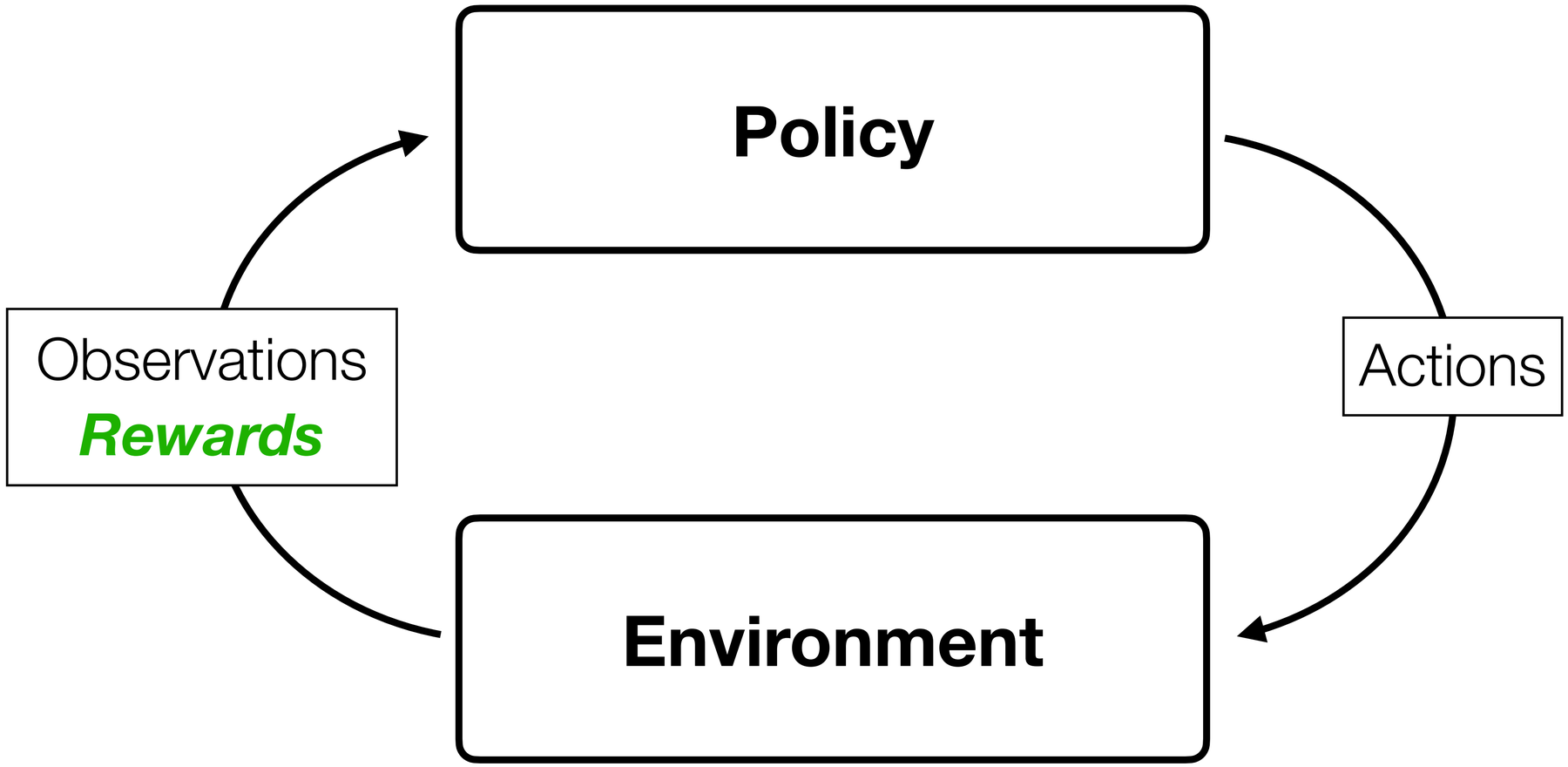
Reinforcement learning



What's a good policy? (what's the learning objective?)

Source: Isola, Torralba, Freeman

Reinforcement learning



Learn a policy that takes actions that maximize **reward**

Source: Isola, Torralba, Freeman

Imitation learning

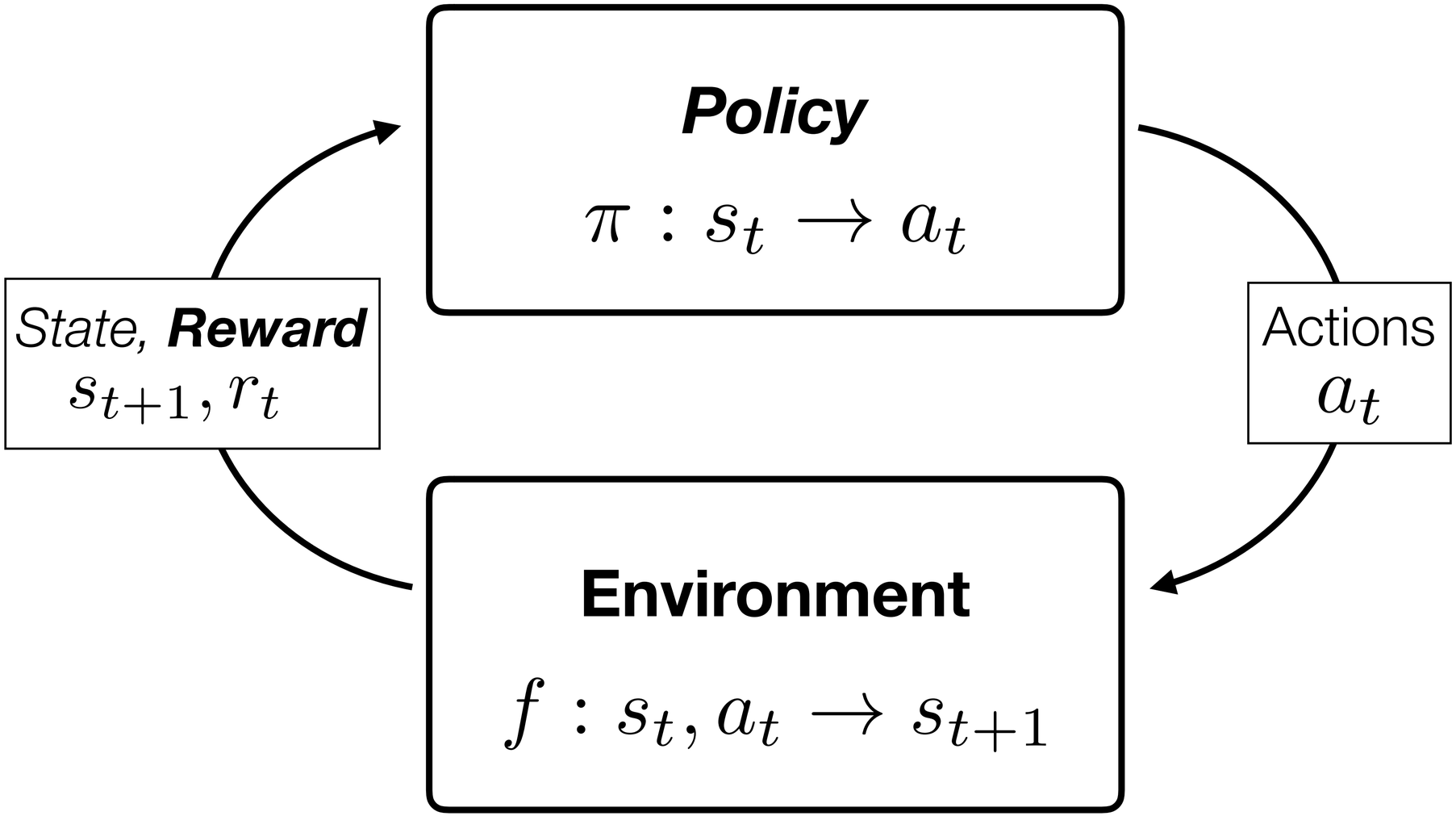
Hand-curated training data
+ Instructive examples
+ Follows a curriculum
- Expensive
- Limited to teacher's knowledge

Reinforcement learning

No training data, have to play around and collect the data *yourself*
+ No need for labeled data
+ Can learn things no human knows how to do
- Less instructive
- No curriculum
- Have to explore

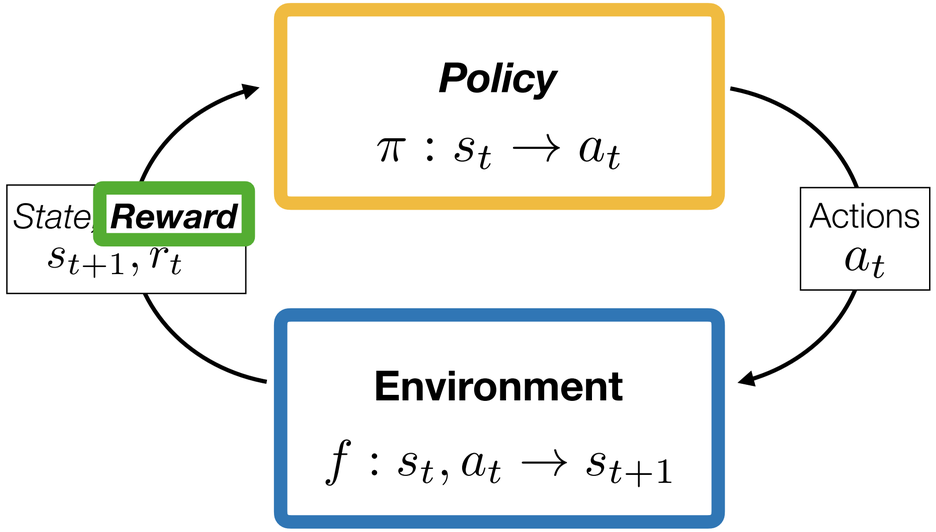
Source: Isola, Torralba, Freeman

Reinforcement learning



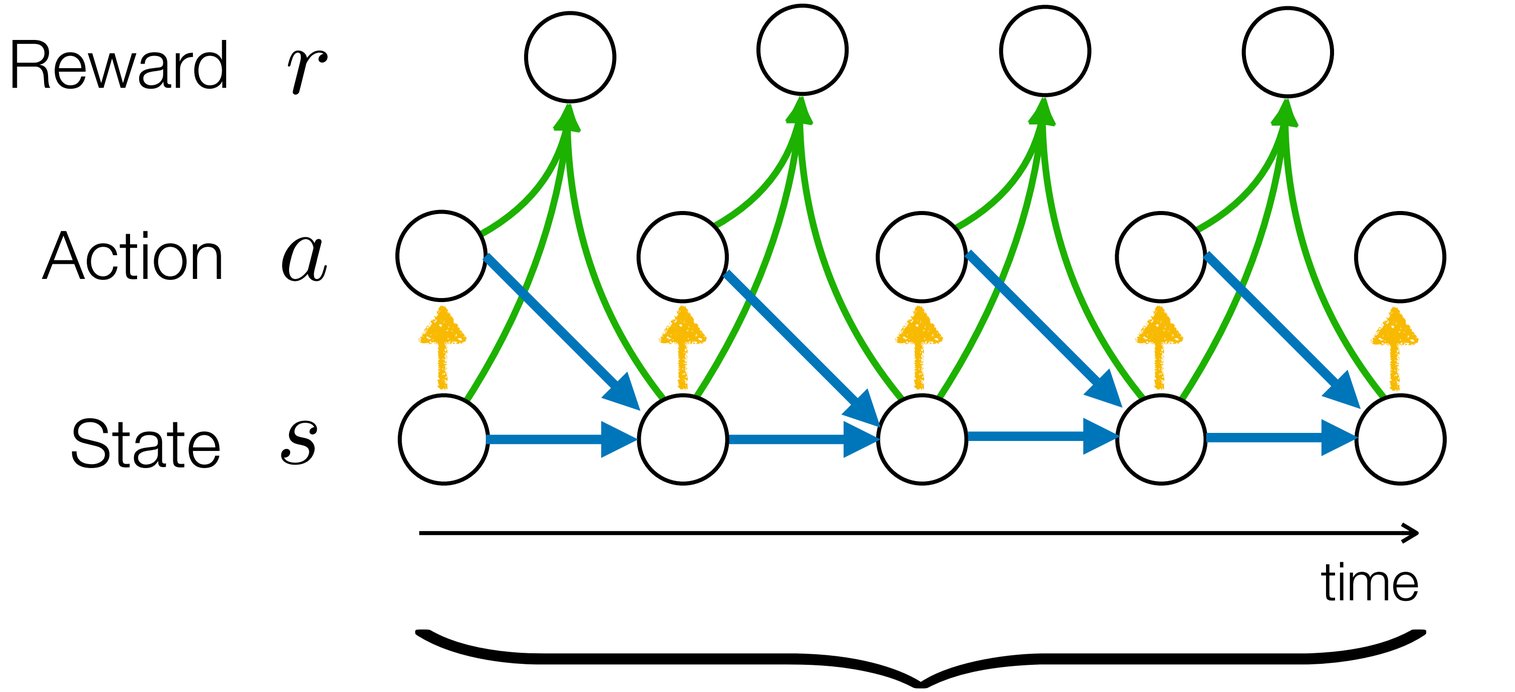
Source: Isola, Torralba, Freeman

Reinforcement learning



Markov decision process (MDP)

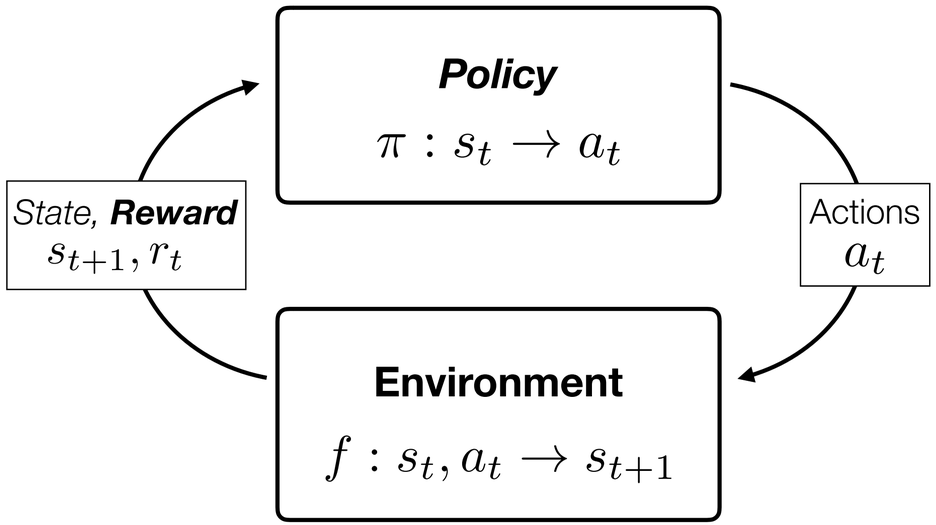
Learned



A sample from the MPD is called a **Trajectory** $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots)$

Source: Isola, Torralba, Freeman

Reinforcement learning



Trajectory $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots)$

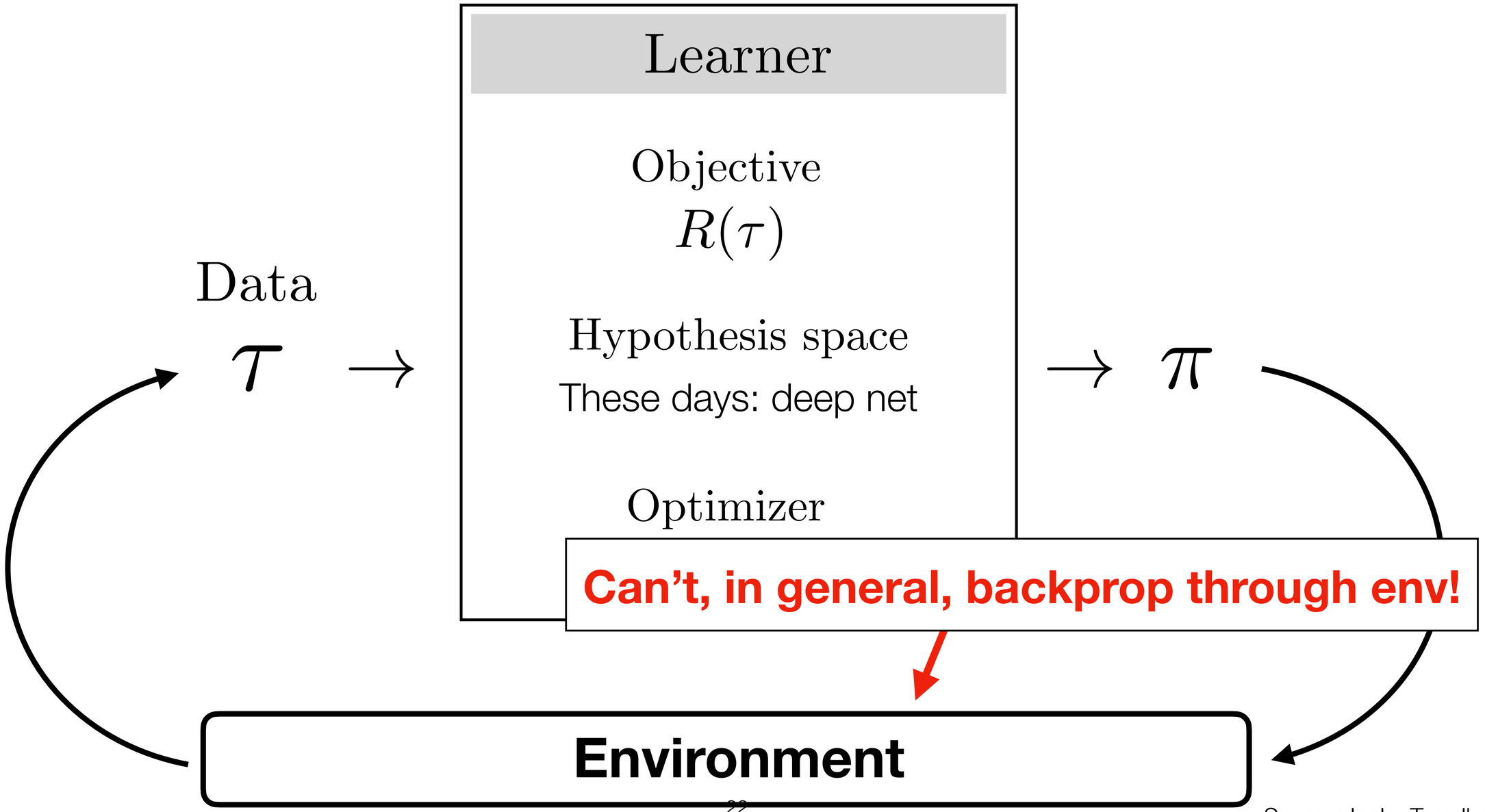
Discounted Returns $R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t, \quad \gamma \in (0, 1)$

Learn a policy that takes actions that maximize expected reward

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau \sim \pi}[R(\tau)]$$

Source: Isola, Torralba, Freeman

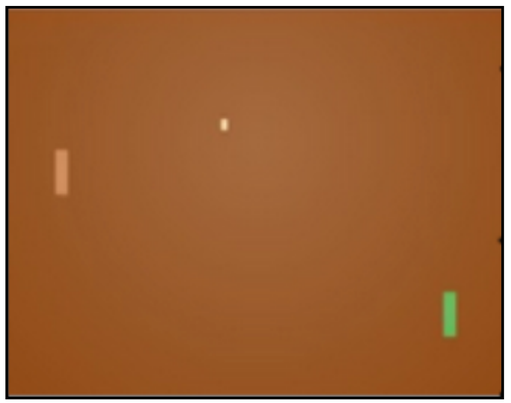
Reinforcement learning



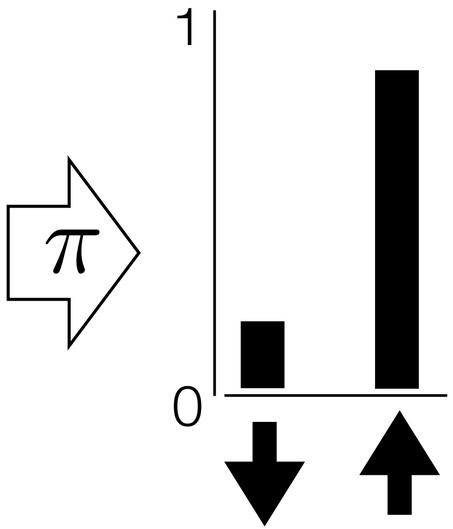
Source: Isola, Torralba, Freeman

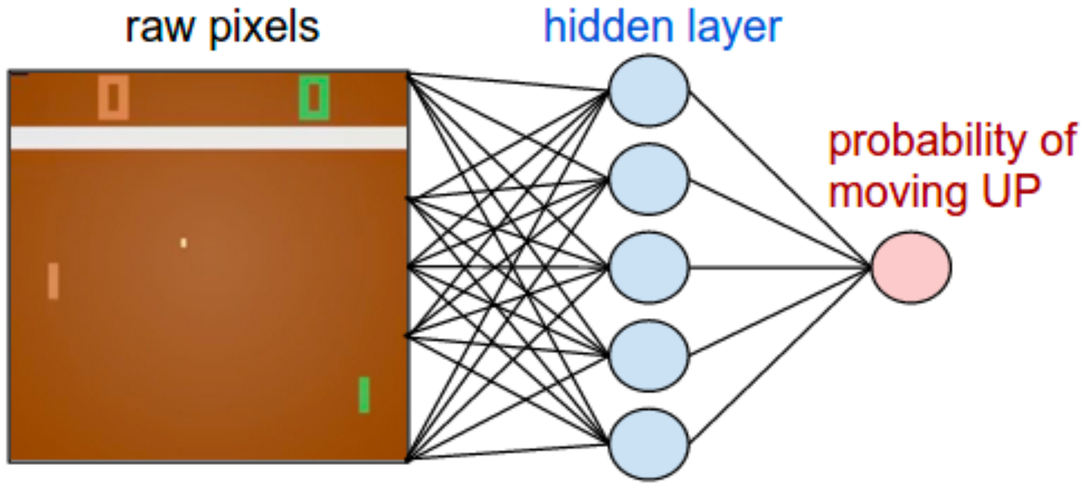
Environment is not differentiable! — How to optimize?

Policy gradients: Run a policy for a while. See what actions led to high rewards. Increase their probability.



Pong

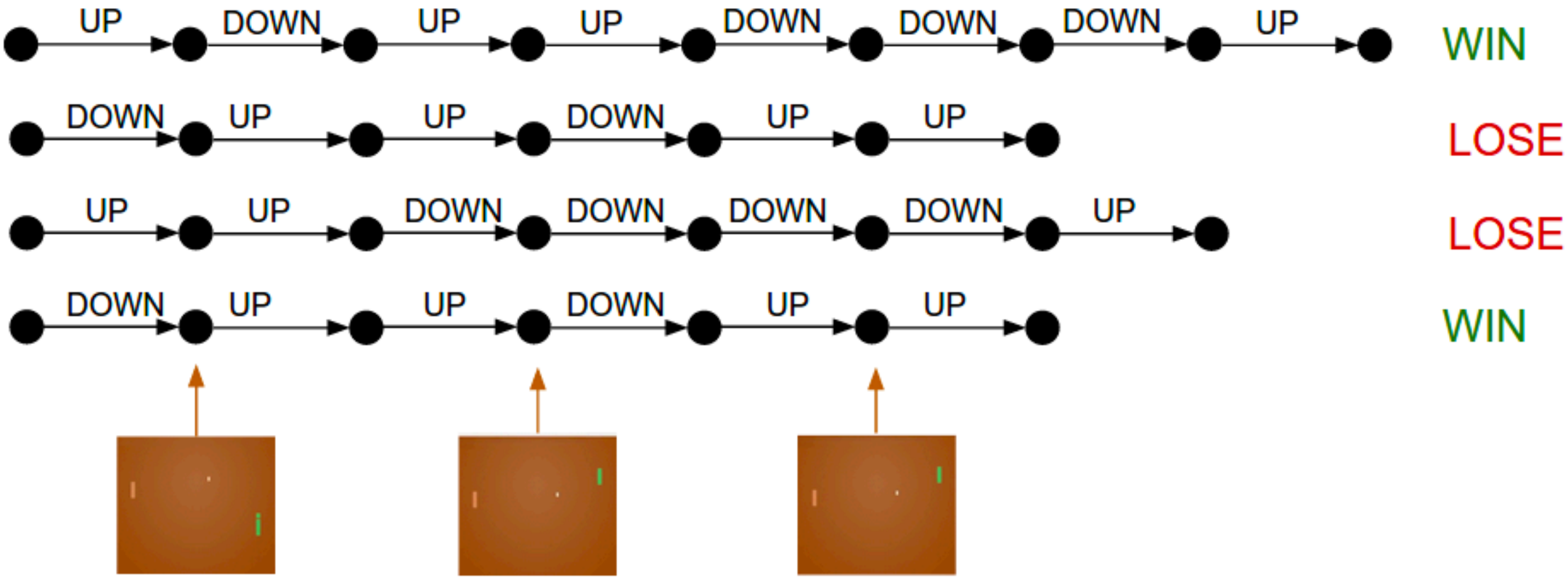




[Adapted from Andrej Karpathy: http://karpathy.github.io/2016/05/31/rl/]

Source: Isola, Torralba, Freeman

Policy gradients: Run a policy for a while. See what actions led to high rewards. Increase their probability.



[Adapted from Andrej Karpathy: http://karpathy.github.io/2016/05/31/rl/]

Source: Isola, Torralba, Freeman

Prediction $\hat{\mathbf{y}}$

$$f_\theta : X \to \mathbb{R}^K$$

Ground truth label $\mathbf{y}$

Loss

$$H(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{k=1}^{K} y_k \log \hat{y}_k$$

$\mathbf{x}$

f

dolphin
cat
grizzly bear
angel fish
chameleon
clown fish
iguana
elephant
⋮

0 1

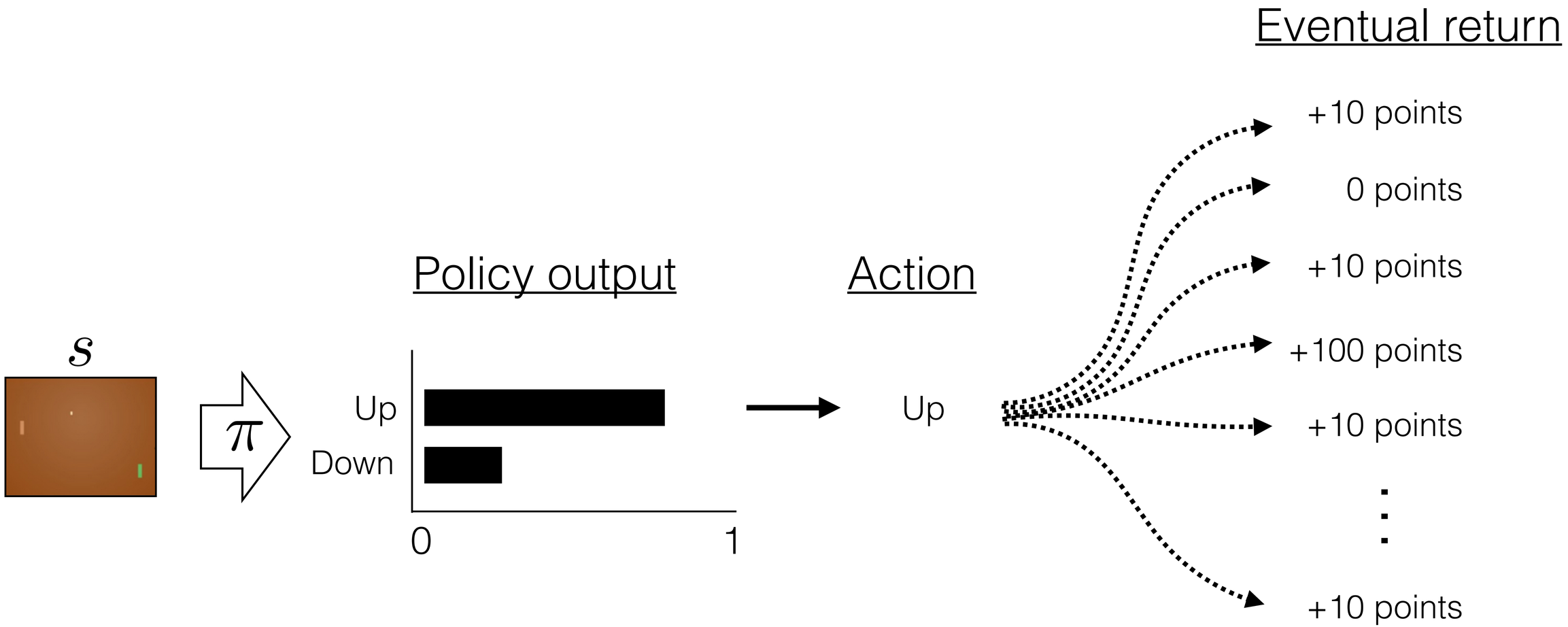
$\odot$

dolphin
cat
grizzly bear
angel fish
chameleon
clown fish
iguana
elephant
⋮

0 1

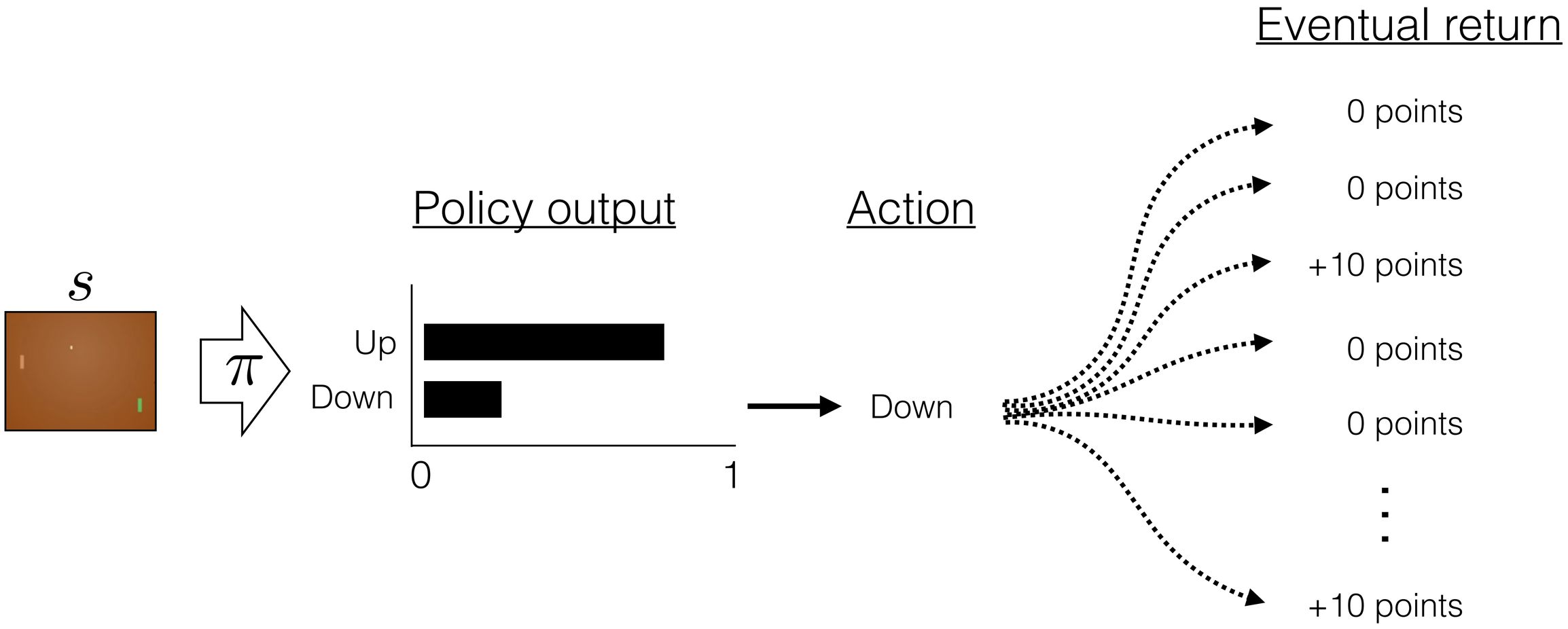
0

Source: Isola, Torralba, Freeman



$\pi(a|s)$ = probability of choosing action *a* given state *s*

Source: Isola, Torralba, Freeman



$\pi(a|s)$ = probability of choosing action *a* given state *s*

Source: Isola, Torralba, Freeman

Policy gradient

- Want to take derivatives of expected reward w.r.t. the policy parameters.

$$\frac{\partial}{\partial\theta}\mathbb{E}_{\tau\sim\pi_\theta}[R(\tau)] = \frac{\partial}{\partial\theta}\int_\tau p(\tau|\theta)R(\tau)d\tau$$

$$= \int_\tau p(\tau|\theta)\left[\frac{\partial}{\partial\theta}\log(p(\tau|\theta))\right]R(\tau)d\tau$$

$$= \mathbb{E}_{\tau\sim\pi_\theta}\left[\frac{\partial}{\partial\theta}\log(p(\tau|\theta))R(\tau)\right]$$

- Do actions with high rewards more often, and low rewards less often
- This is called the REINFORCE algorithm.
- Converges very slowly, in comparison to supervised learning with gradients

Policy gradient

- What happens in a rollout? Recall we're maximizing $\mathbb{E}_{\tau \sim \pi_\theta}\left[\frac{\partial}{\partial \theta} \log(p(\tau|\theta)) R(\tau)\right]$

$$\frac{\partial}{\partial \theta} \log(p(\tau|\theta)) = \frac{\partial}{\partial \theta} \log\left[p(s_0) \prod_{t=0}^{T} \pi_\theta(a_t|s_t) \prod_{t=1}^{T} p(s_t|s_{t-1}, a_{t-1})\right]$$

$$= \frac{\partial}{\partial \theta} \log \prod_{t=0}^{T} \pi_\theta(a_t|s_t)$$

$$= \sum_{t=0}^{T} \frac{\partial}{\partial \theta} \log \pi_\theta(a_t|s_t)$$

- *All* actions become more likely if the reward is high.
- Doesn't do *credit assignment*.

Slide adapted from R. Grosse and J. Ba

Policy gradient

- Algorithm: do SGD on policy

1. Sample a rollout, e.g. play the game with current policy

$$\tau = (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$$

2. Compute reward, e.g. what was our game score?

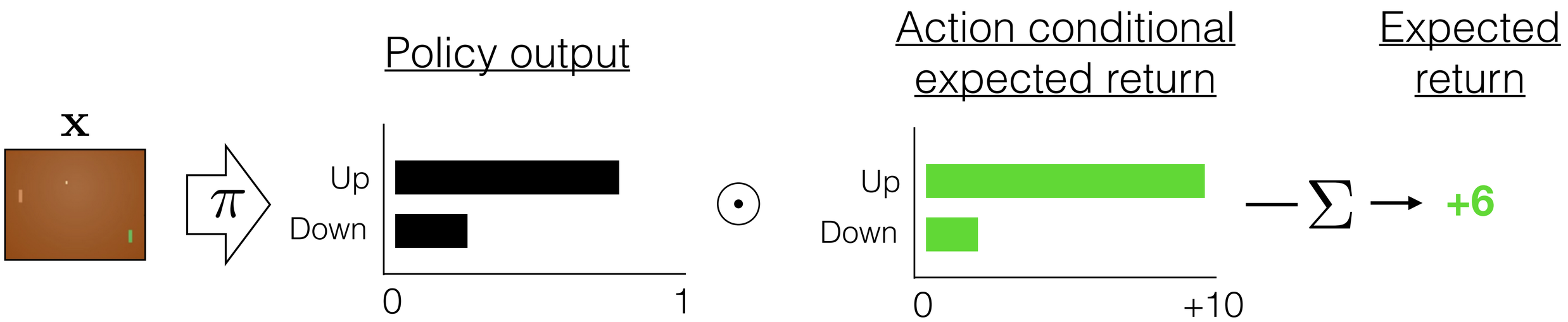
$$r(\tau) = \sum_{t=0}^{T} R(s_t)$$

3. Do a gradient update:

$$\theta \leftarrow \theta + \alpha r(\tau) \frac{\partial}{\partial \theta} \pi_\theta(a_t | s_t)$$

Slide adapted from R. Grosse and J. Ba

Approximated via sampling



$$\nabla_\theta \mathbb{E}_{\tau\sim\pi_\theta}[R(\tau)] = \mathbb{E}_{\tau\sim\pi_\theta}[R(\tau)\nabla_\theta \log \pi_\theta]$$ ← **Estimate gradient using REINFORCE and do gradient descent**

Source: Isola, Torralba, Freeman

Environment is not differentiable! — How to optimize?

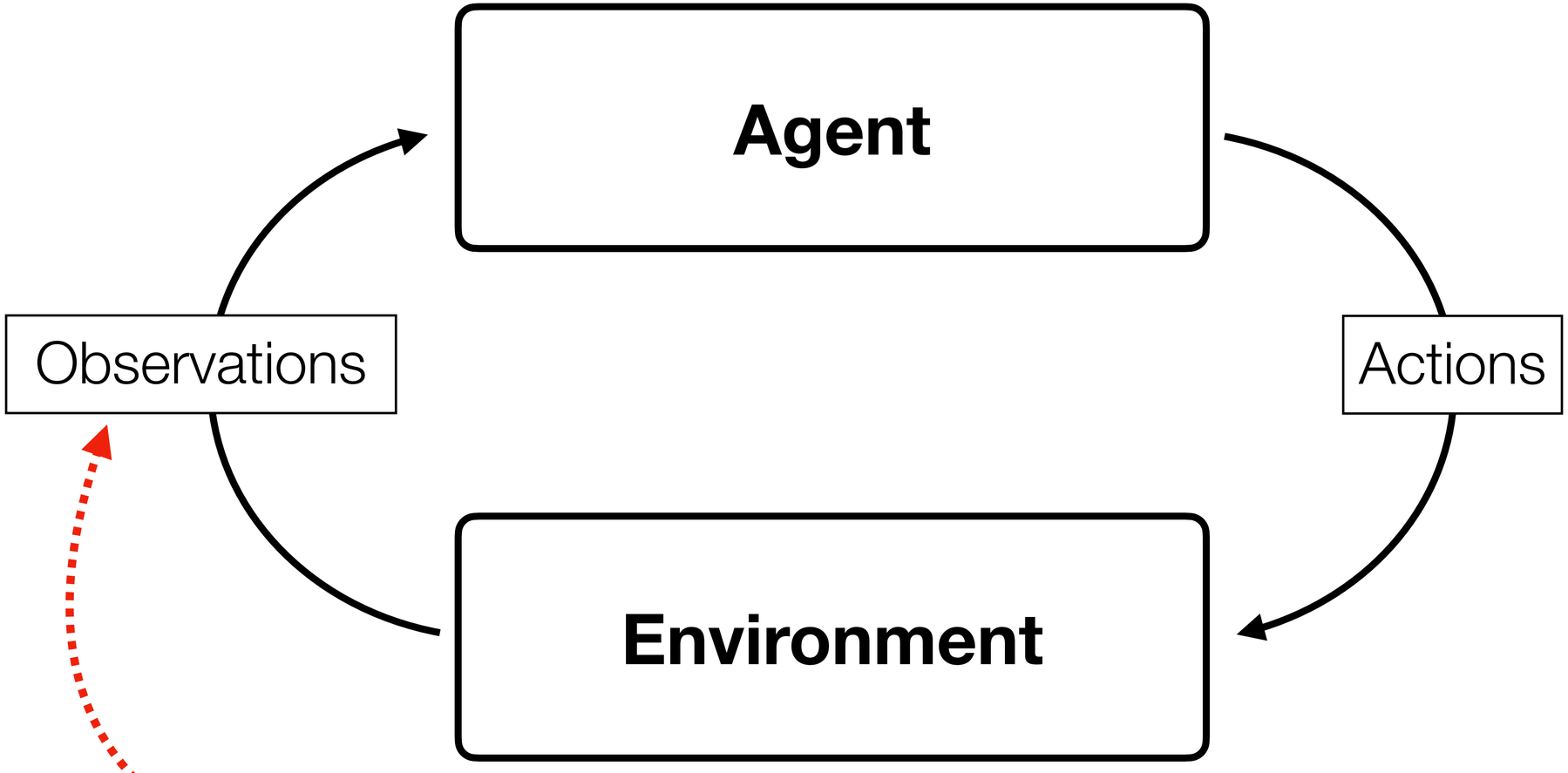
Policy gradients

1. Start with an arbitrary initial policy
2. **Rollout** this *stochastic* policy a bunch of times, sampling different random actions each time
3. Update your policy to place higher probability on actions that led to higher returns

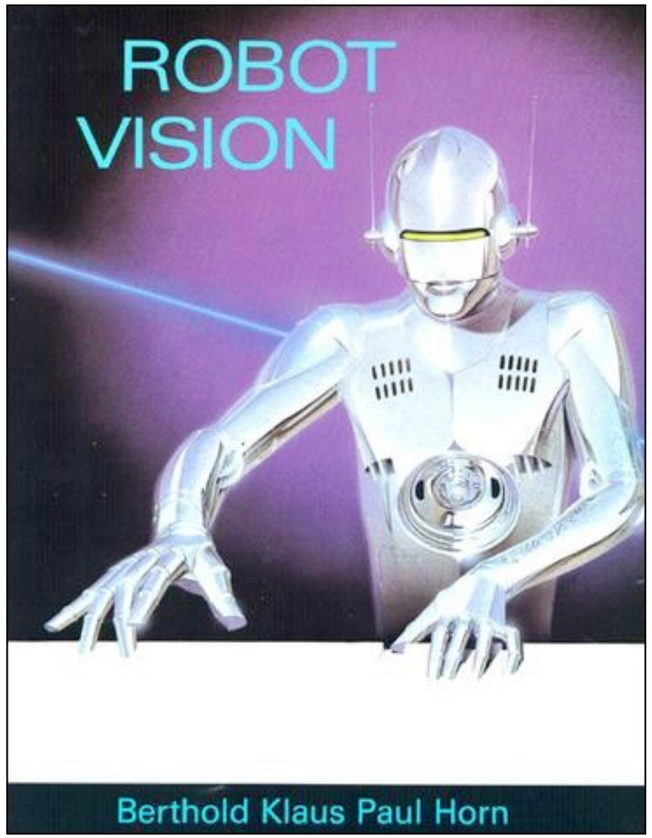
Mathematically, this approximates gradient ascent on policy parameters, so as to maximize reward.

Source: Isola, Torralba, Freeman

Intelligent agents



Why vision?



Source: Isola, Torralba, Freeman

Why vision?

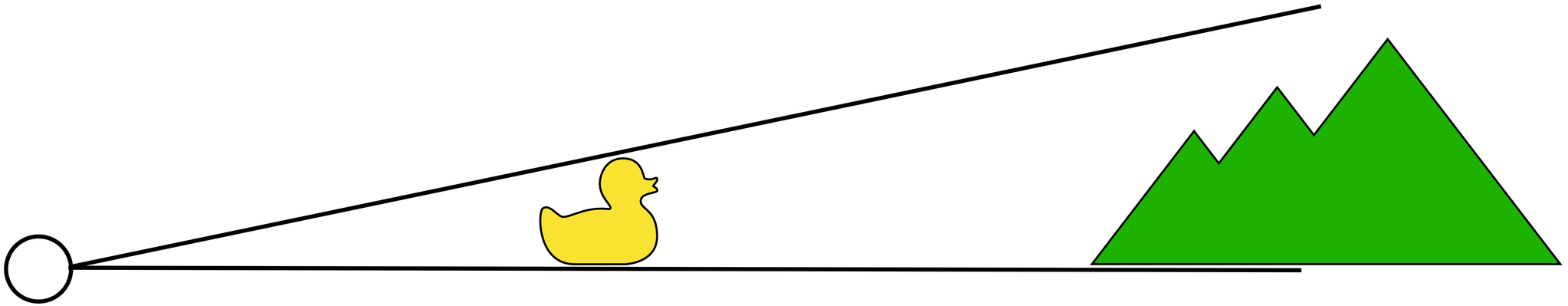
1. Human-like intelligence (and animal-like), relies heavily on vision

We already know it works well!

Source: Isola, Torralba, Freeman

Why vision?

2. Eyes are good sensors



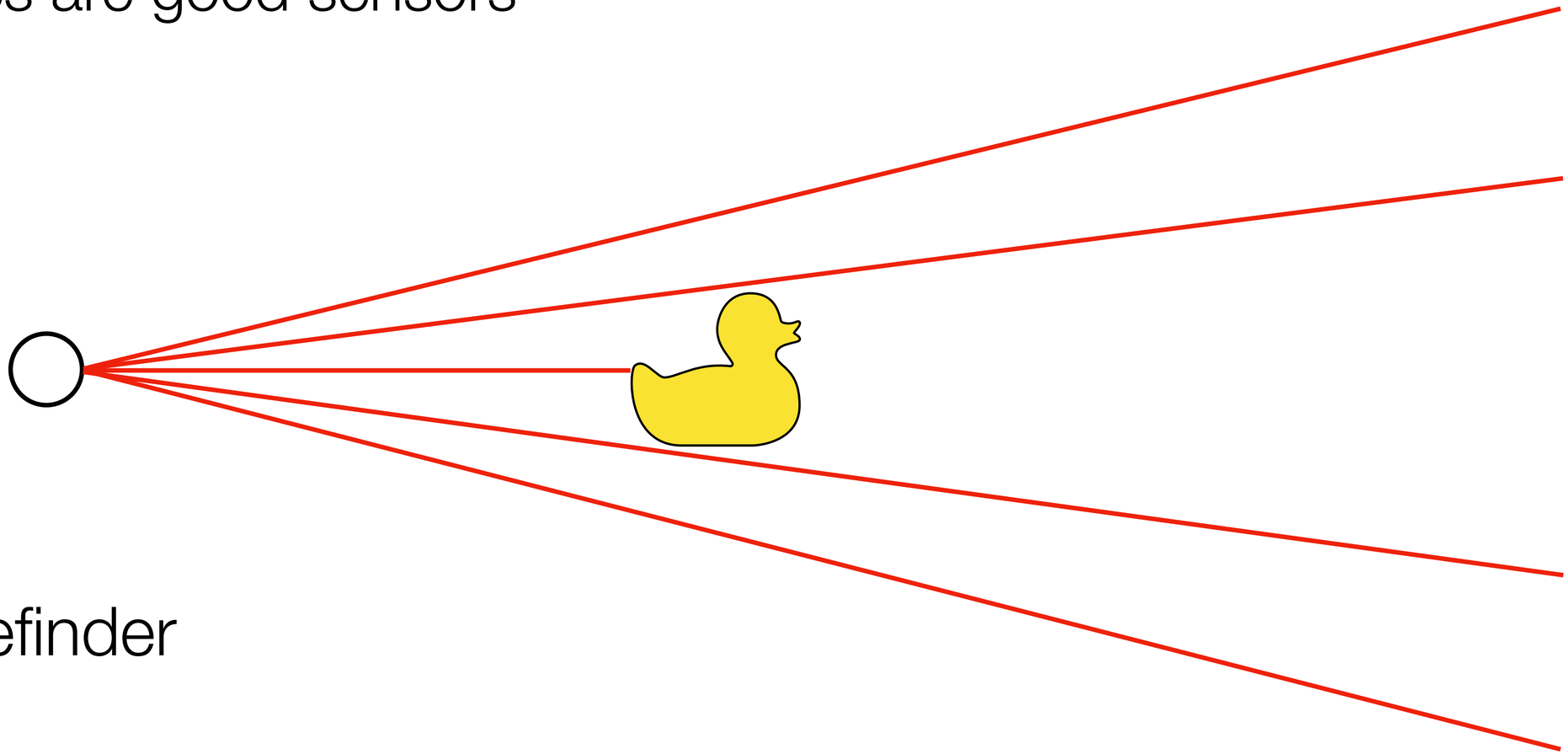
Farther away things look smaller

Get details on stuff that we can immediately interact with,
rough summary of more distant context

Source: Isola, Torralba, Freeman

Why vision?

2. Eyes are good sensors

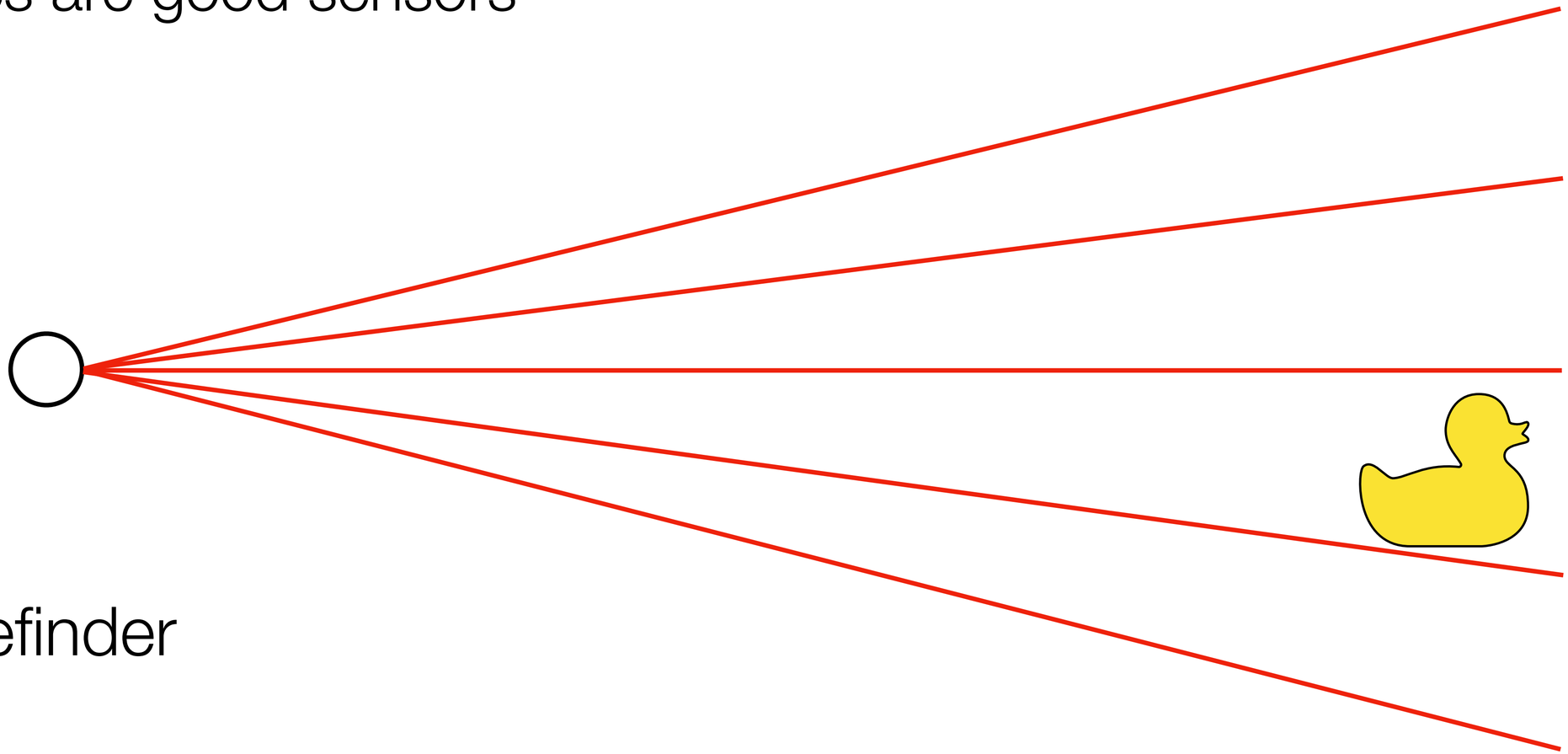


Laser rangefinder

Source: Isola, Torralba, Freeman

Why vision?

2. Eyes are good sensors

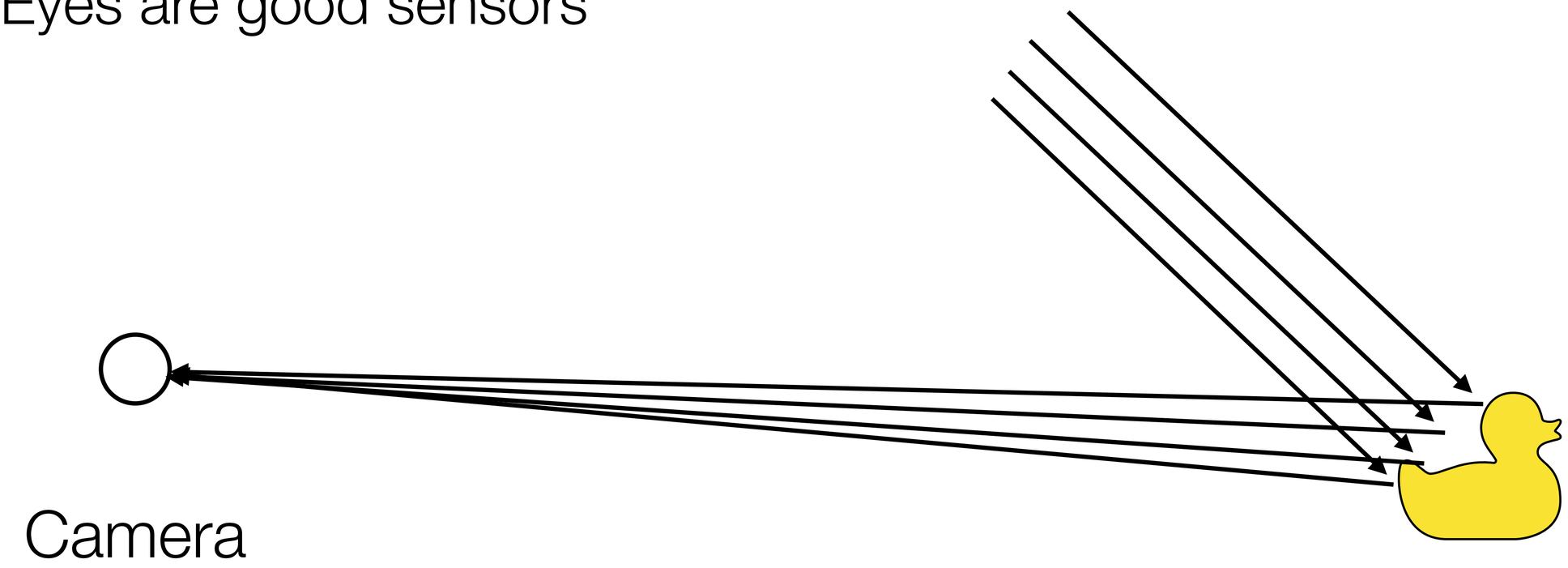


Laser rangefinder

Source: Isola, Torralba, Freeman

Why vision?

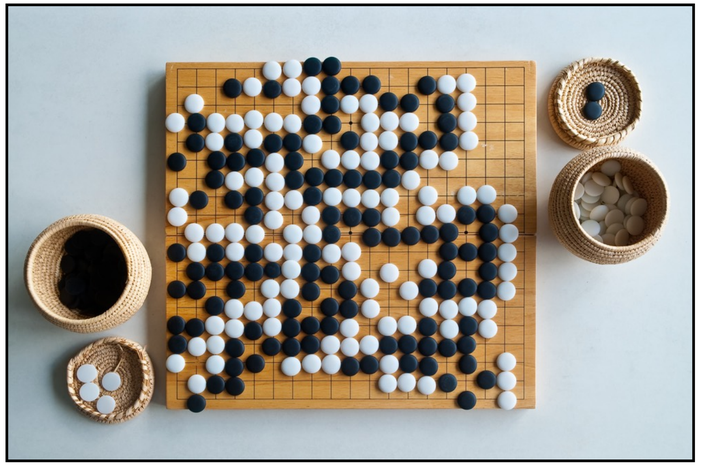
2. Eyes are good sensors

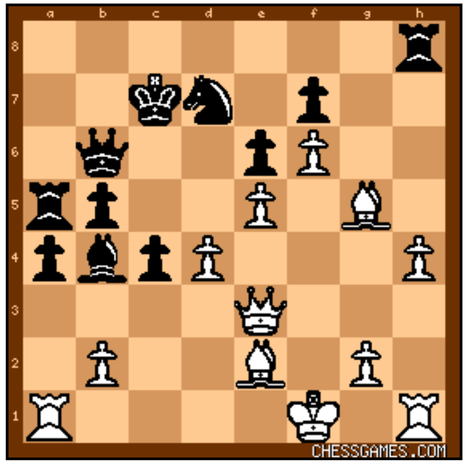


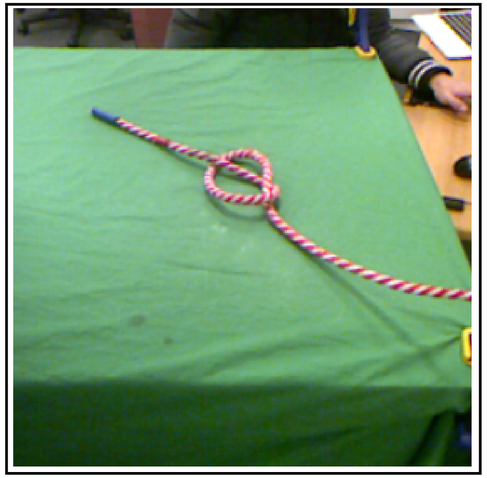
Source: Isola, Torralba, Freeman

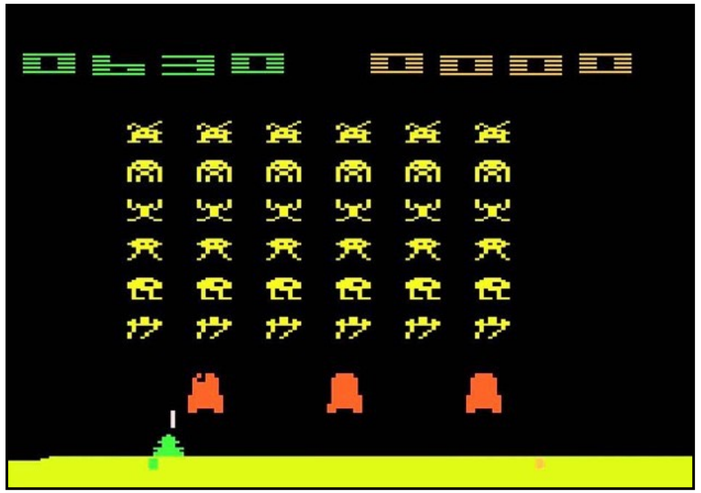
Why vision?

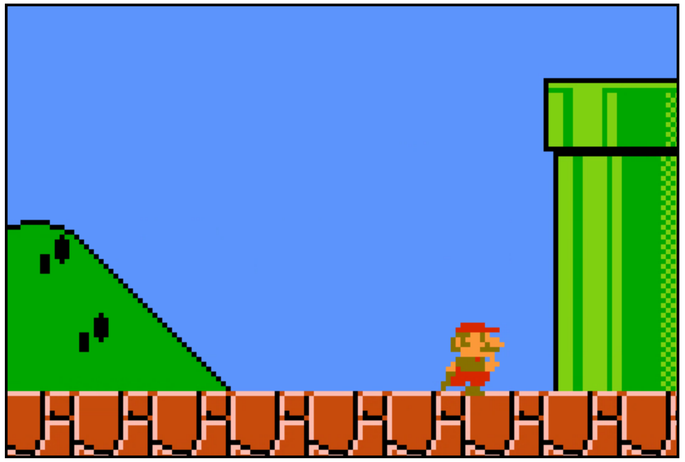
3. Universal interface







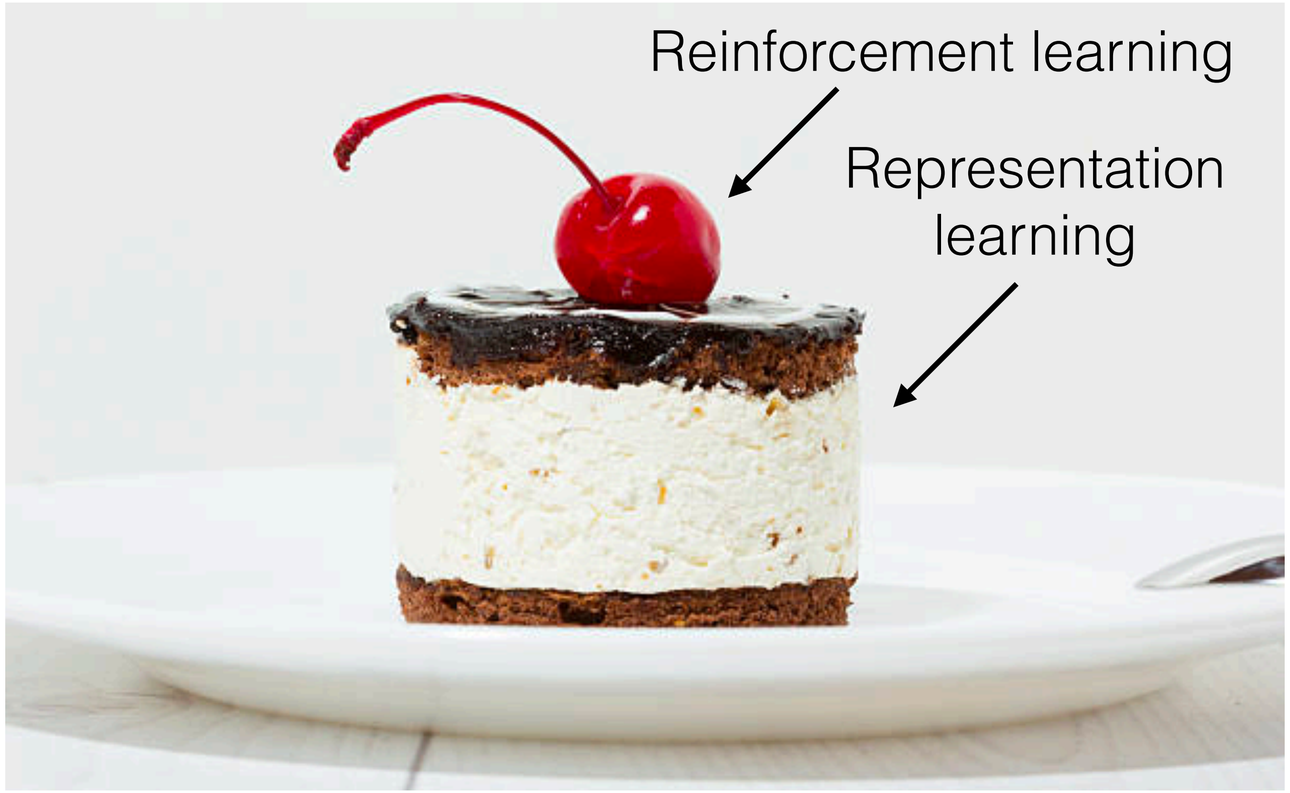




Source: Isola, Torralba, Freeman

Model-based intelligence

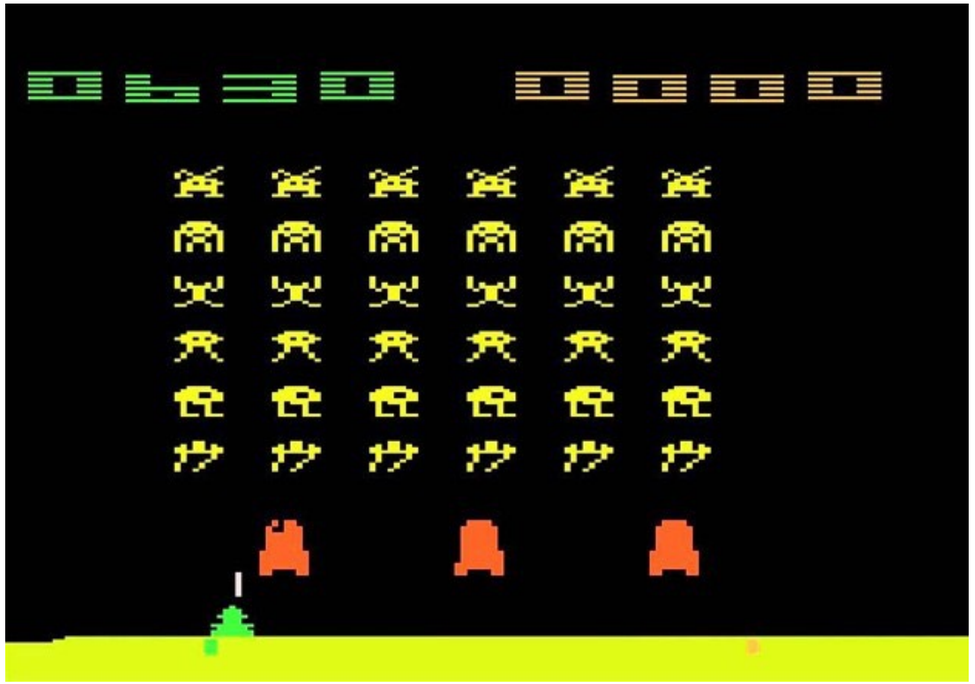
If vision can give us a good representation/model of the world, then planning and control should be easy.



Yann LeCun's "cake"

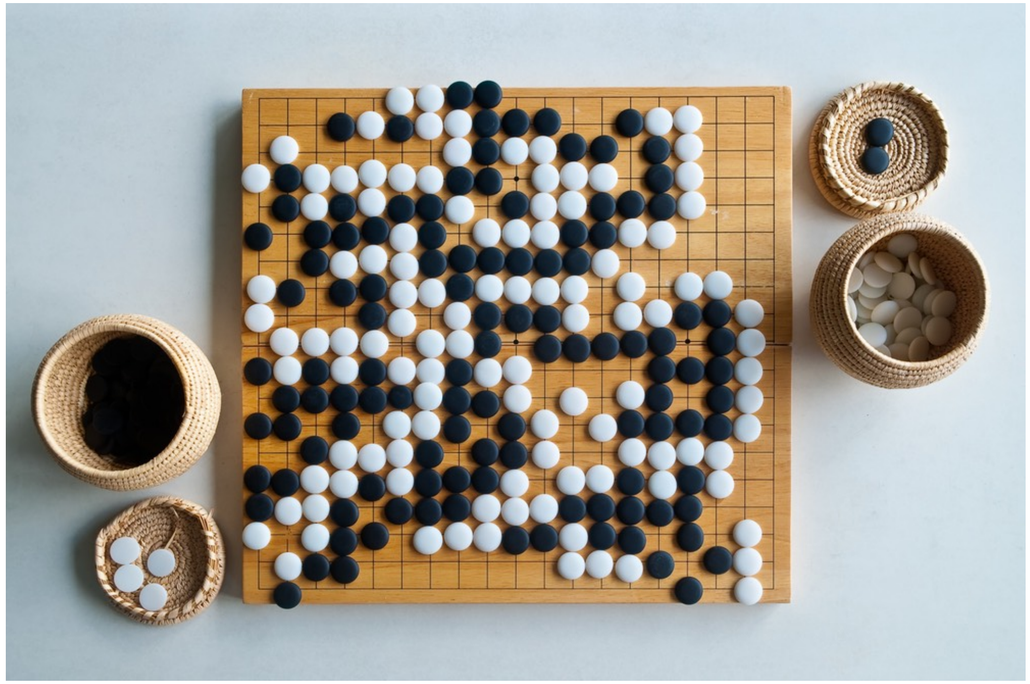
Source: Isola, Torralba, Freeman

Atari Games

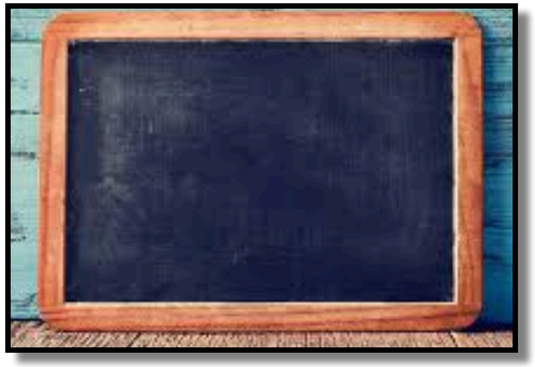


~10-50 million interactions!

AlphaGo



21 million games!



[Slide adapted from Pulkit Agrawal]

Source: Isola, Torralba, Freeman

Next class: more visual reinforcement learning