Lecture 22: Embodied vision

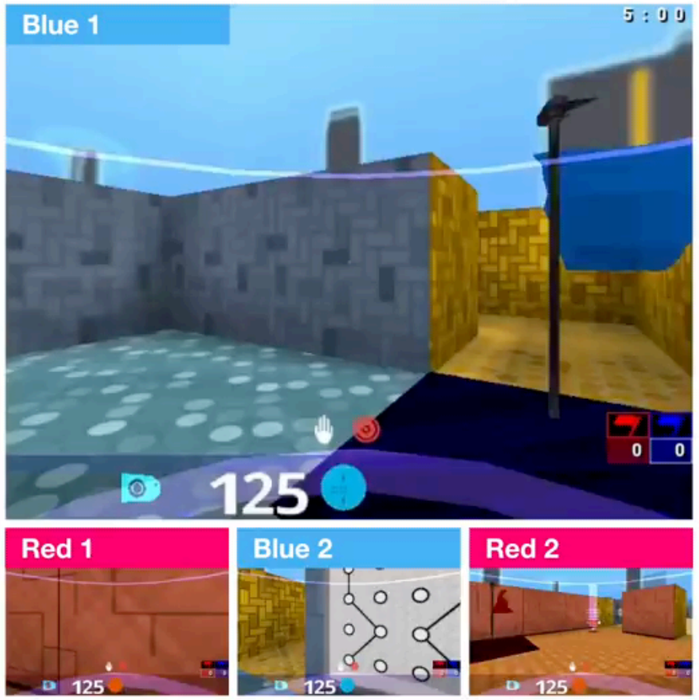
Today

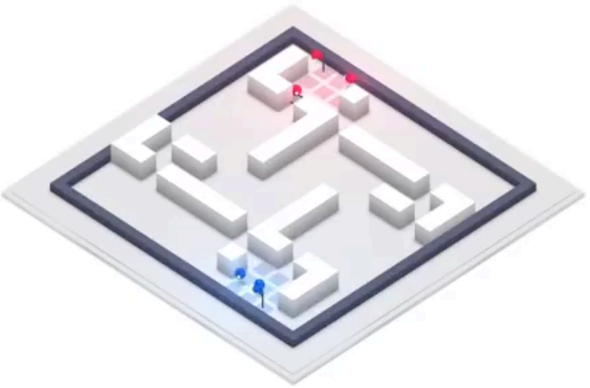
- Formalisms for intelligent agents (environment, state, action, policy)
- Imitation learning
- Reinforcement learning
  - Policy gradient algorithm
  - Q-learning
- This is just a very high-level overview
  - See Sutton & Barto [http://incompleteideas.net/book/RLbook2018.pdf] for more.

Source: Isola, Torralba, Freeman

[Silver et al., 2016]

Agent observation raw pixels





Indoor map overview

[Jaderberg et al. 2018]

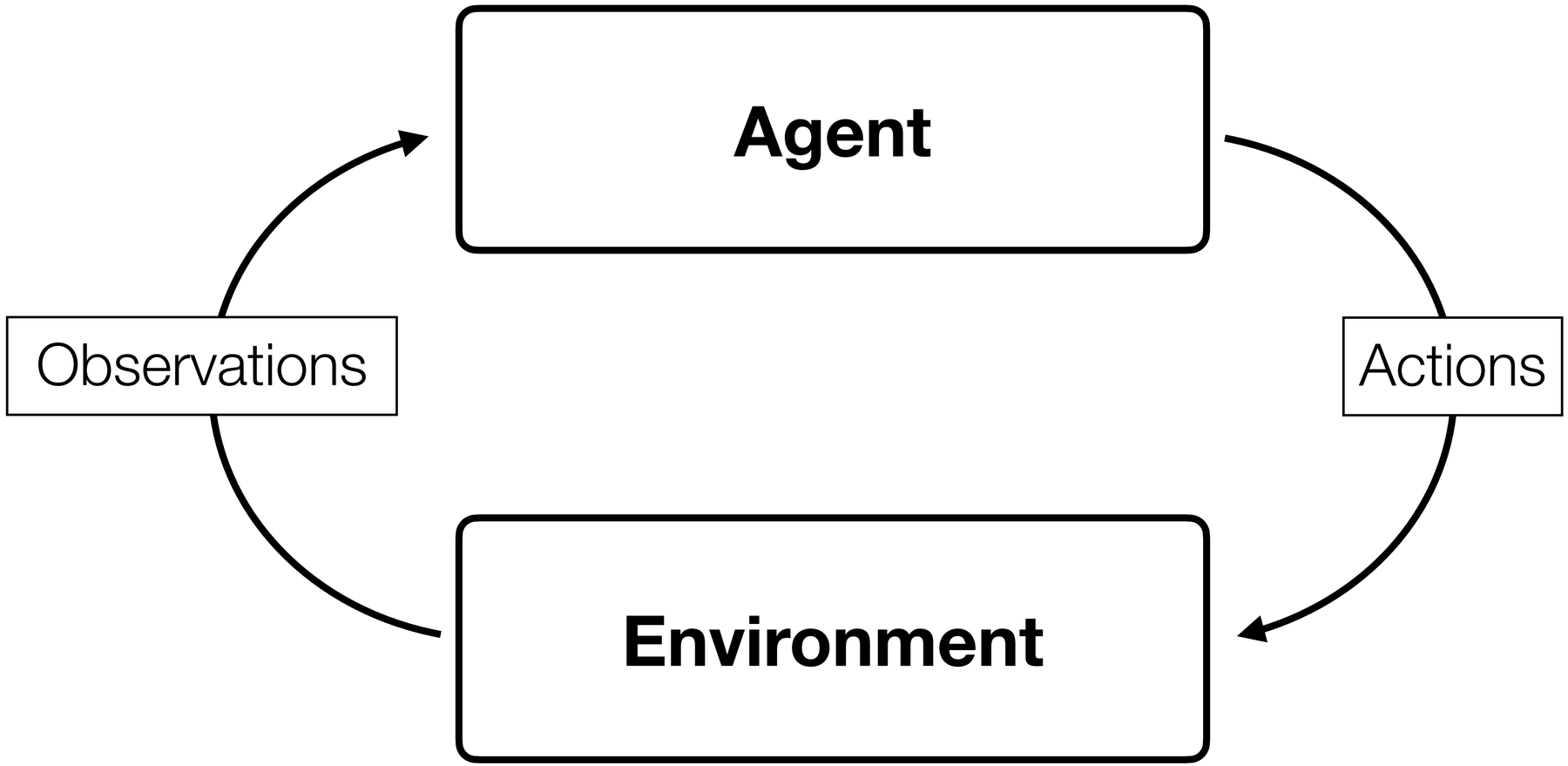
Source: Isola, Torralba, Freeman

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

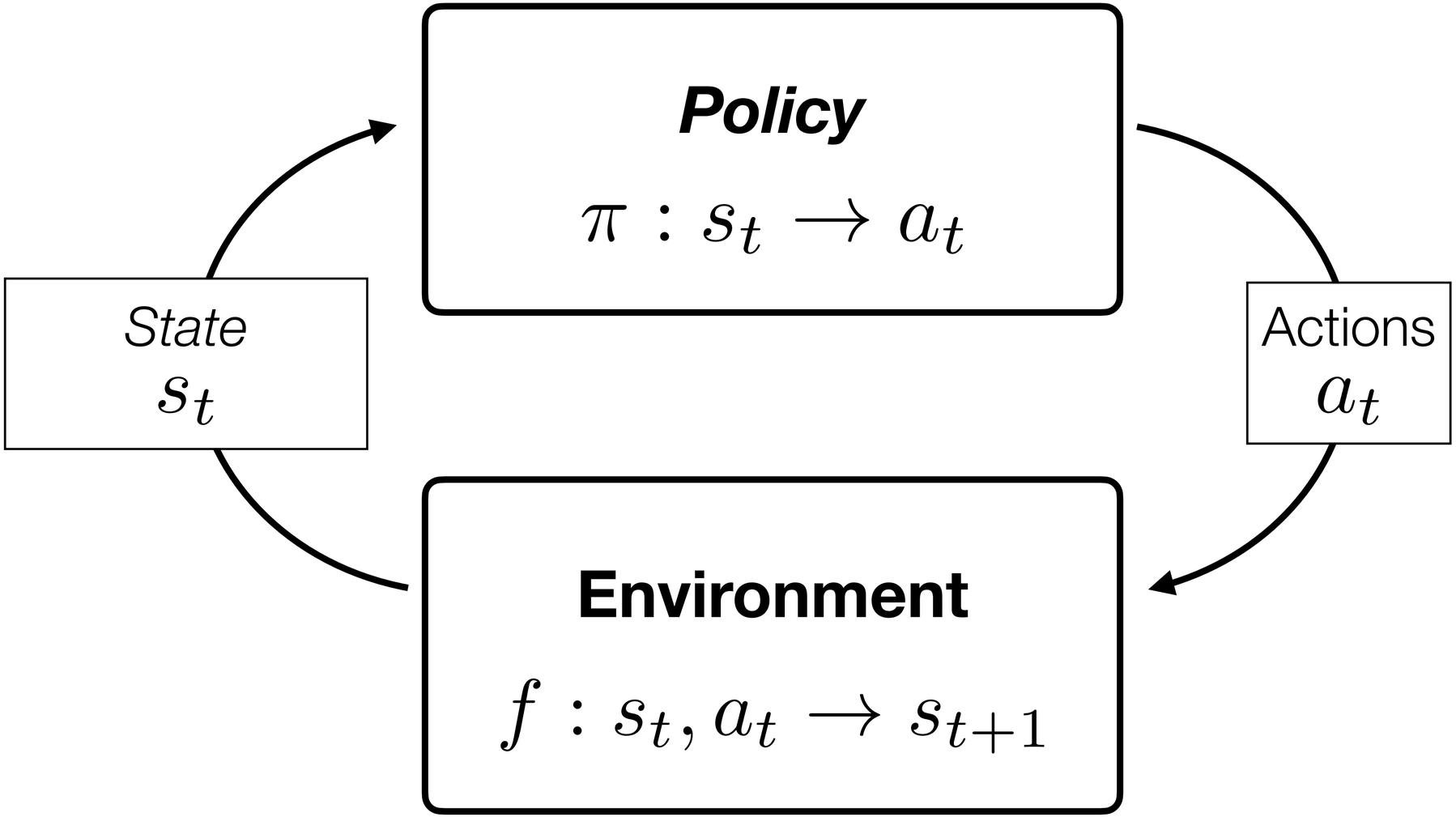
Source: Isola, Torralba, Freeman

Intelligent agents



Source: Isola, Torralba, Freeman

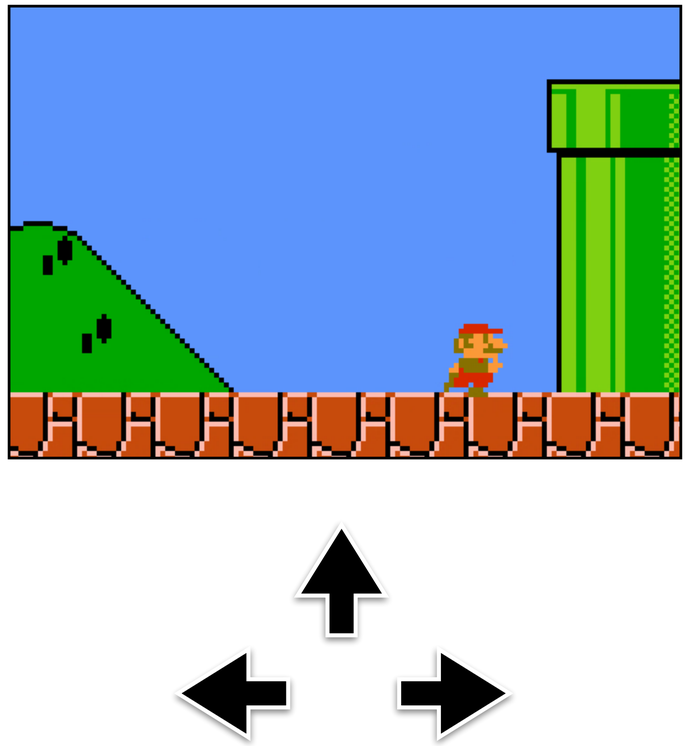
Intelligent agents



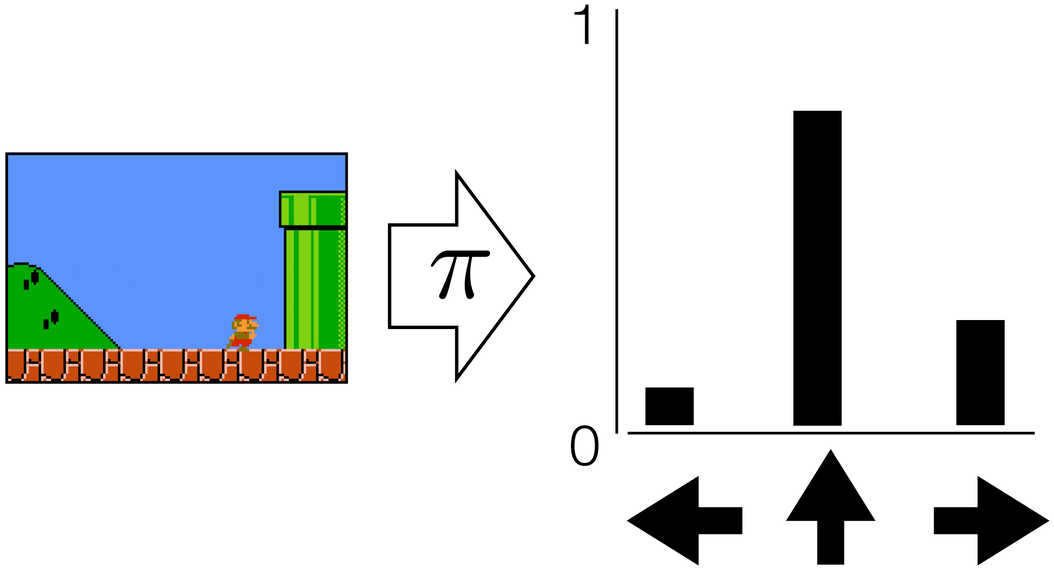
Source: Isola, Torralba, Freeman

How to represent a state? How to represent policy?

state: pixels!



policy: action classifier



Source: Isola, Torralba, Freeman

Can we use 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

(supervised learning, applied to learn *policies*)

Training data

(from an **expert**)

$\{s_1, a_1\}$

$\{s_2, a_2\}$

$\{s_3, a_3\}$

$\ldots$

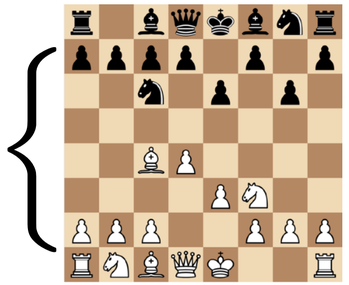
$\rightarrow$ Learner $\rightarrow$ $\pi : s \rightarrow a$

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

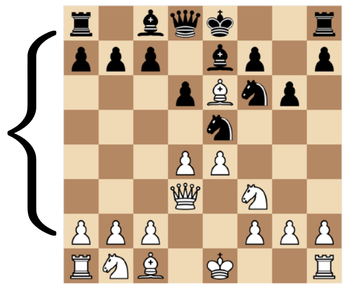
Source: Isola, Torralba, Freeman

Imitation learning

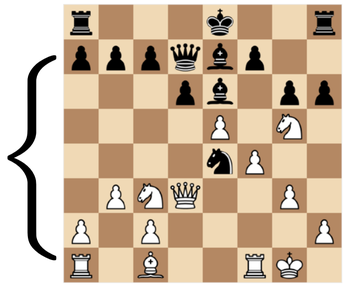
s a



d3 ♗b4+



♘f3 ♞c6



0-0 ♗xc3

⋮

→

Learner

Objective

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

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

Hypothesis space

Convolutional neural net

Optimizer

Stochastic gradient descent

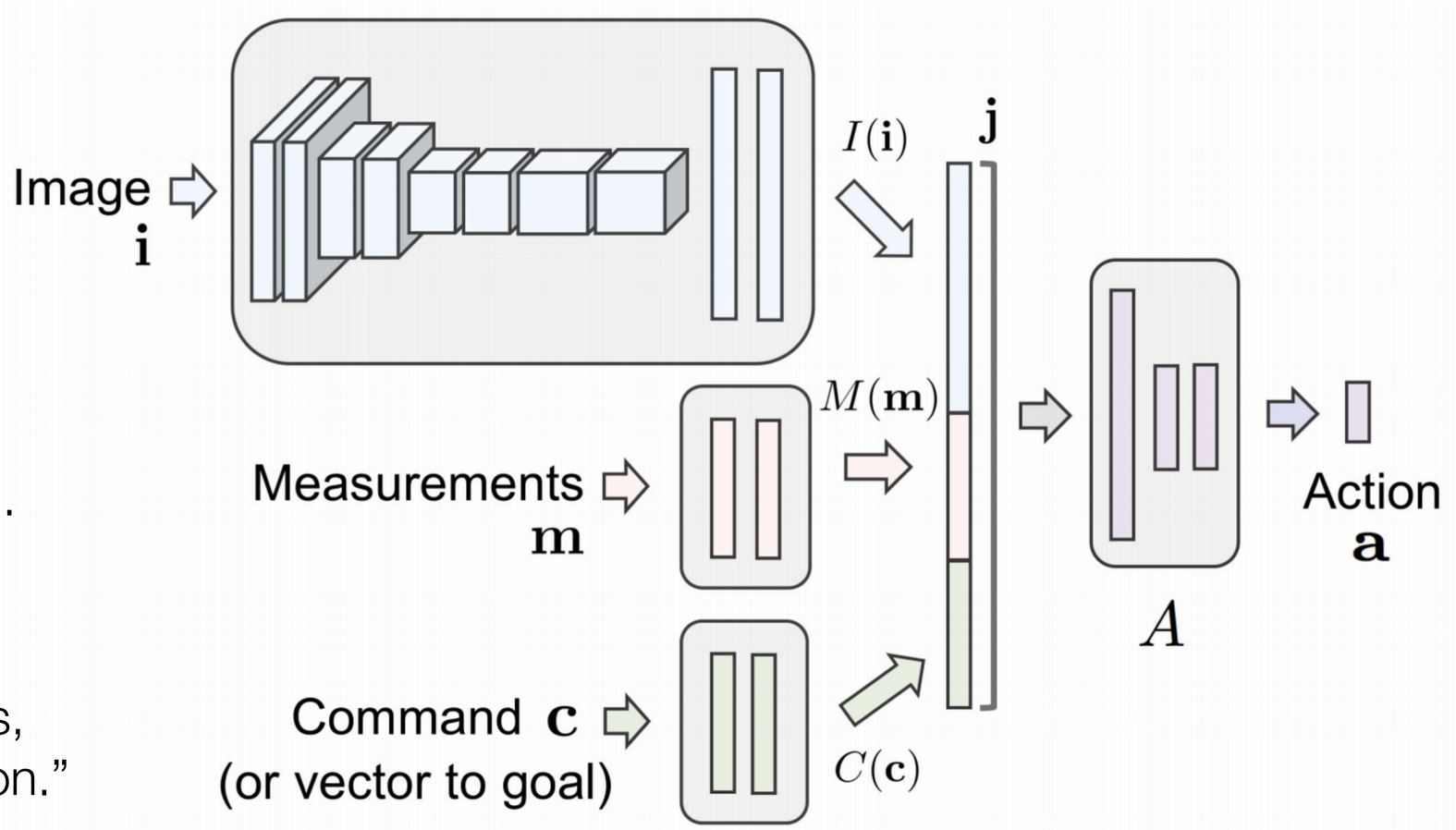
→ π

Source: Isola, Torralba, Freeman

From images to actions

+ auxiliary measurements, e.g. speed.

+ a 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]

End-to-end Driving via Conditional Imitation Learning

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

Submitted to ICRA 2018

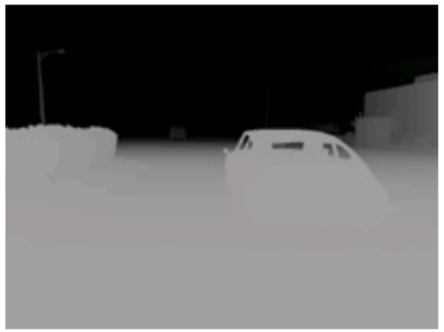

FPS: 15
Speed: 0 km/h
Gear: N
GAS
BRAKE
4x

Exploiting other knowledge

Segmentation

Albedo

Depth

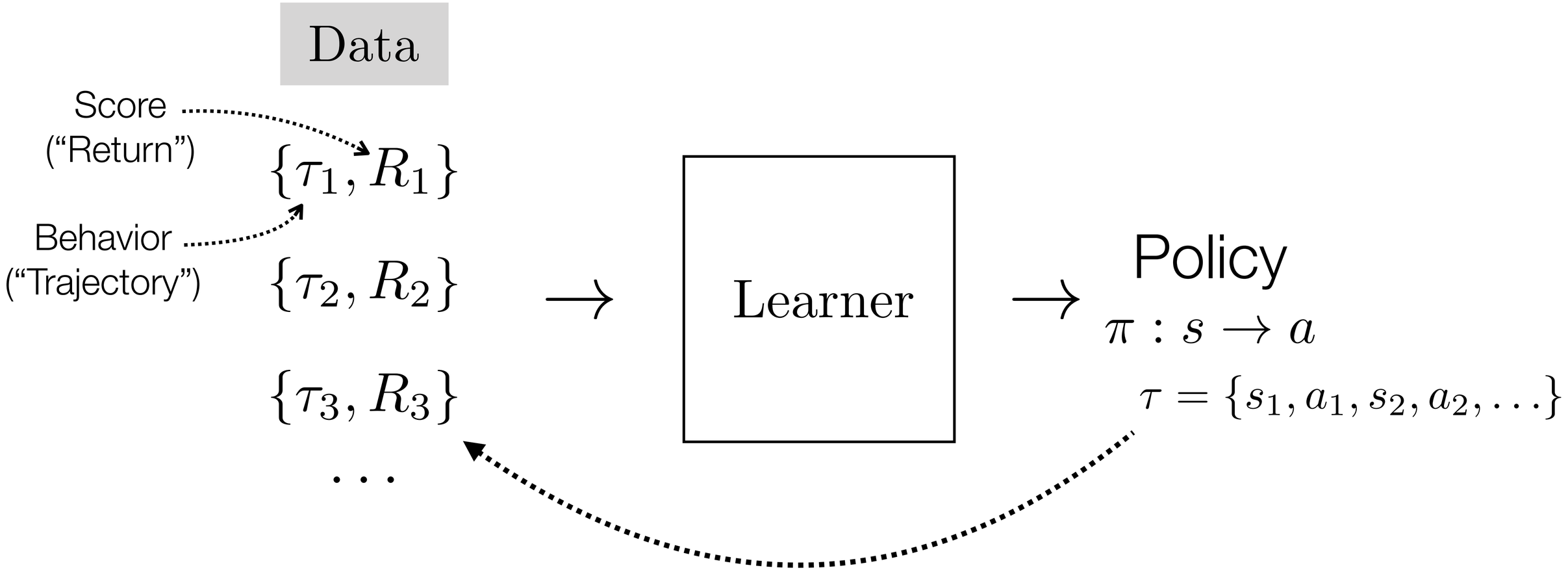


Optical flow

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

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

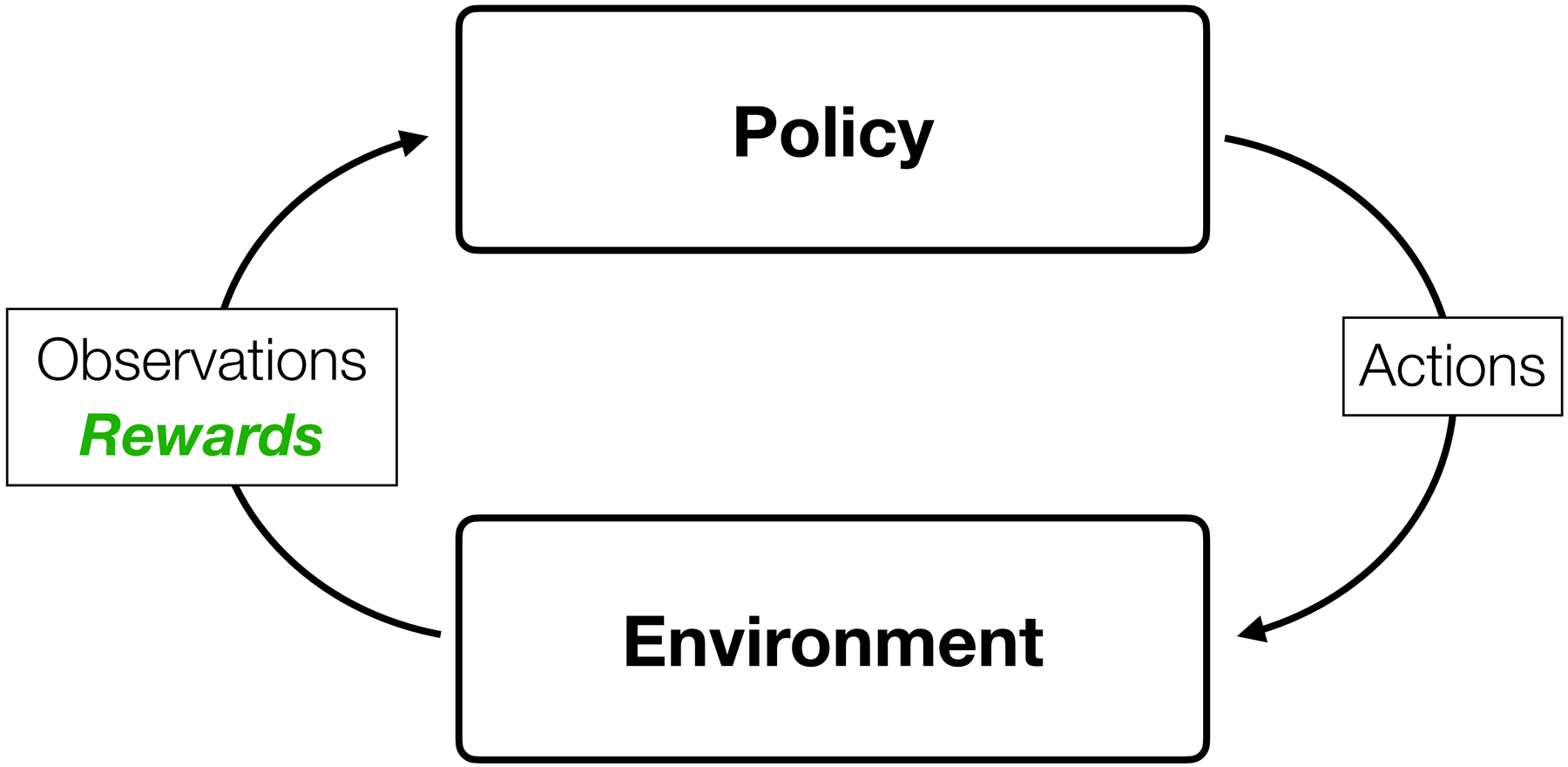
Reinforcement learning



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

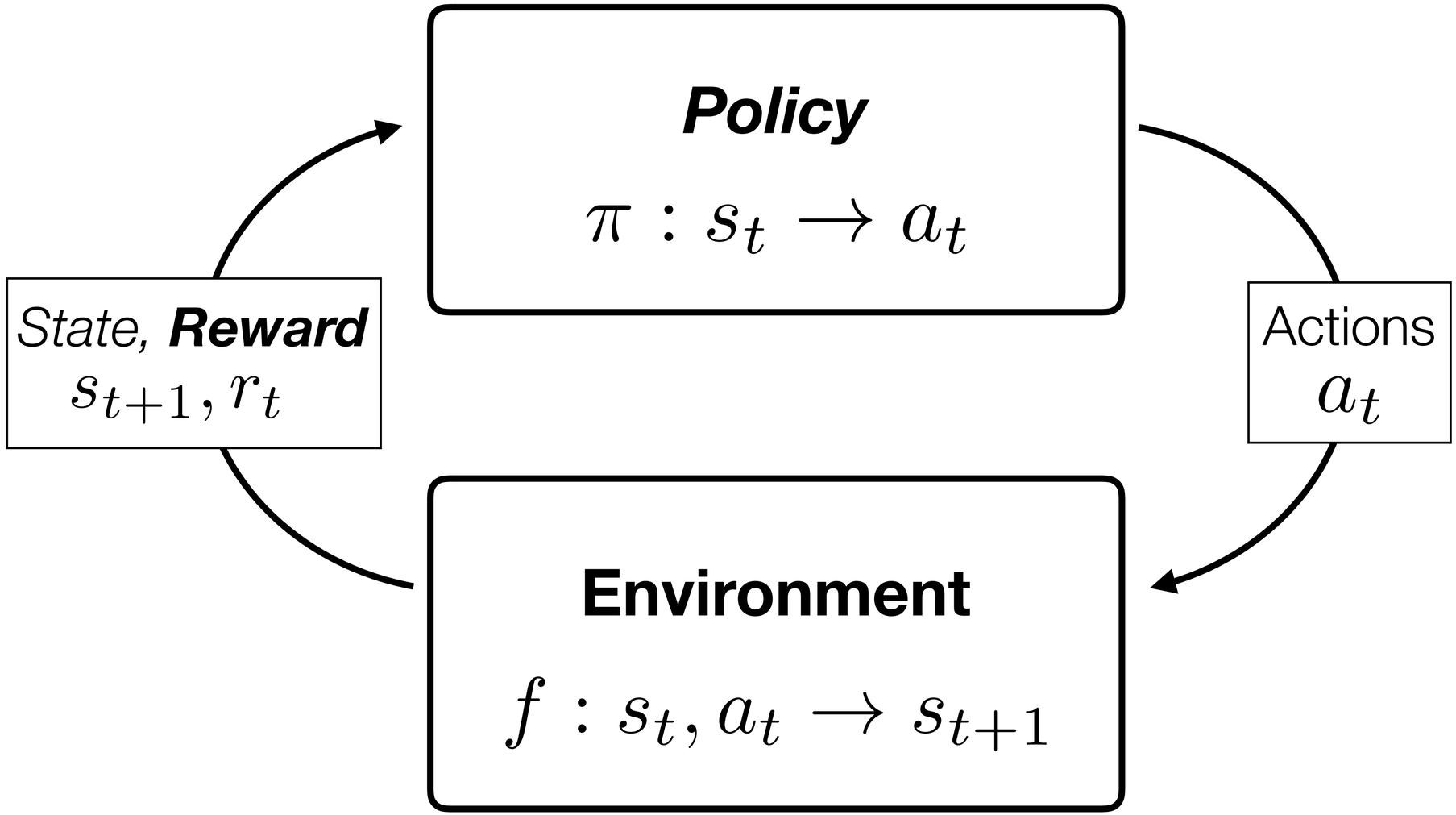
Hand-curated training data
+ Instructive examples
+ Follows a curriculum
- Expensive
- Limited to teacher's knowledge

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

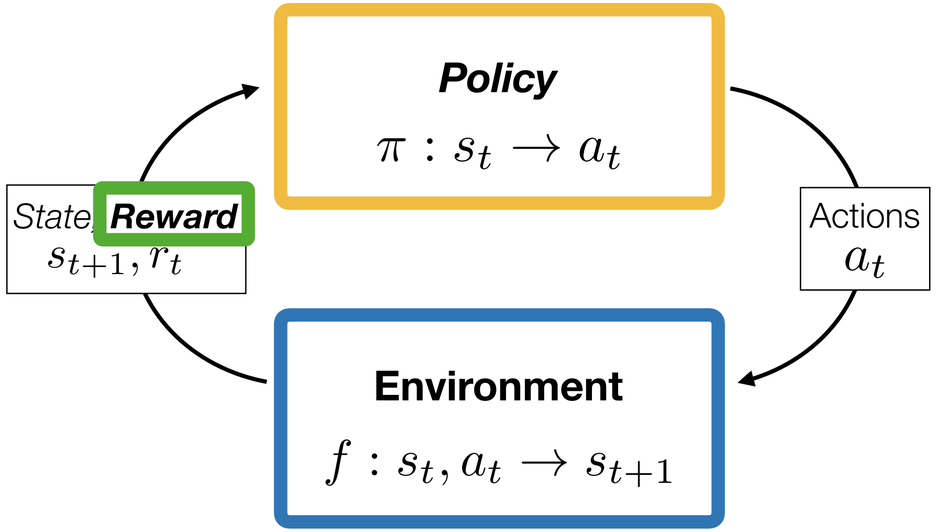
Source: Isola, Torralba, Freeman

Reinforcement learning

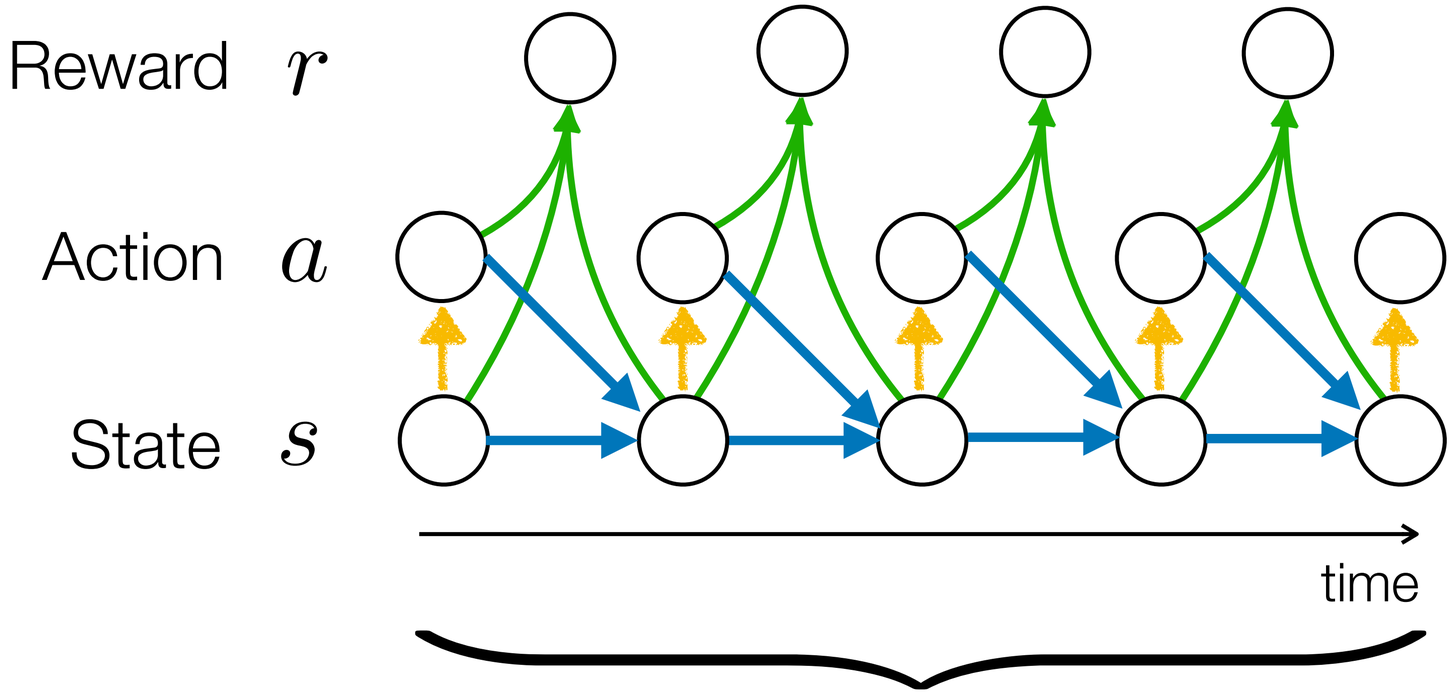


Source: Isola, Torralba, Freeman

Reinforcement learning



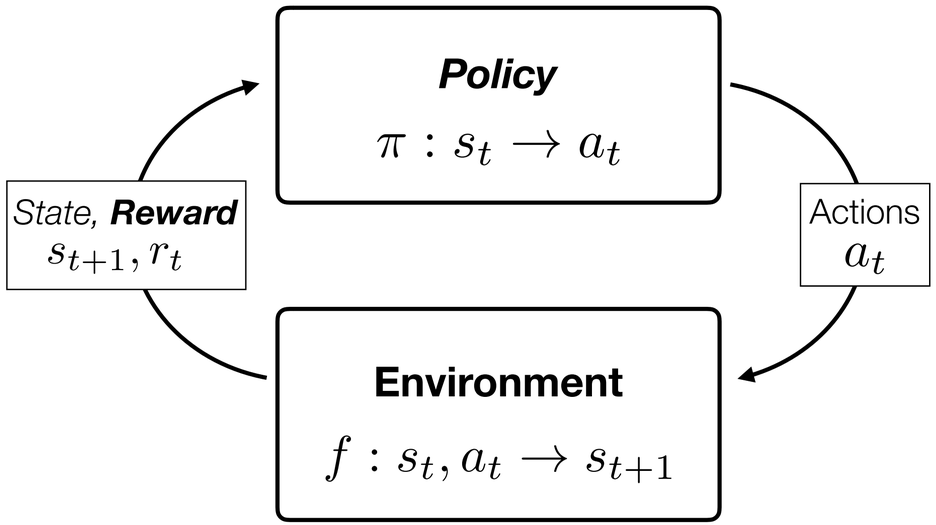
Markov decision process (MDP)



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

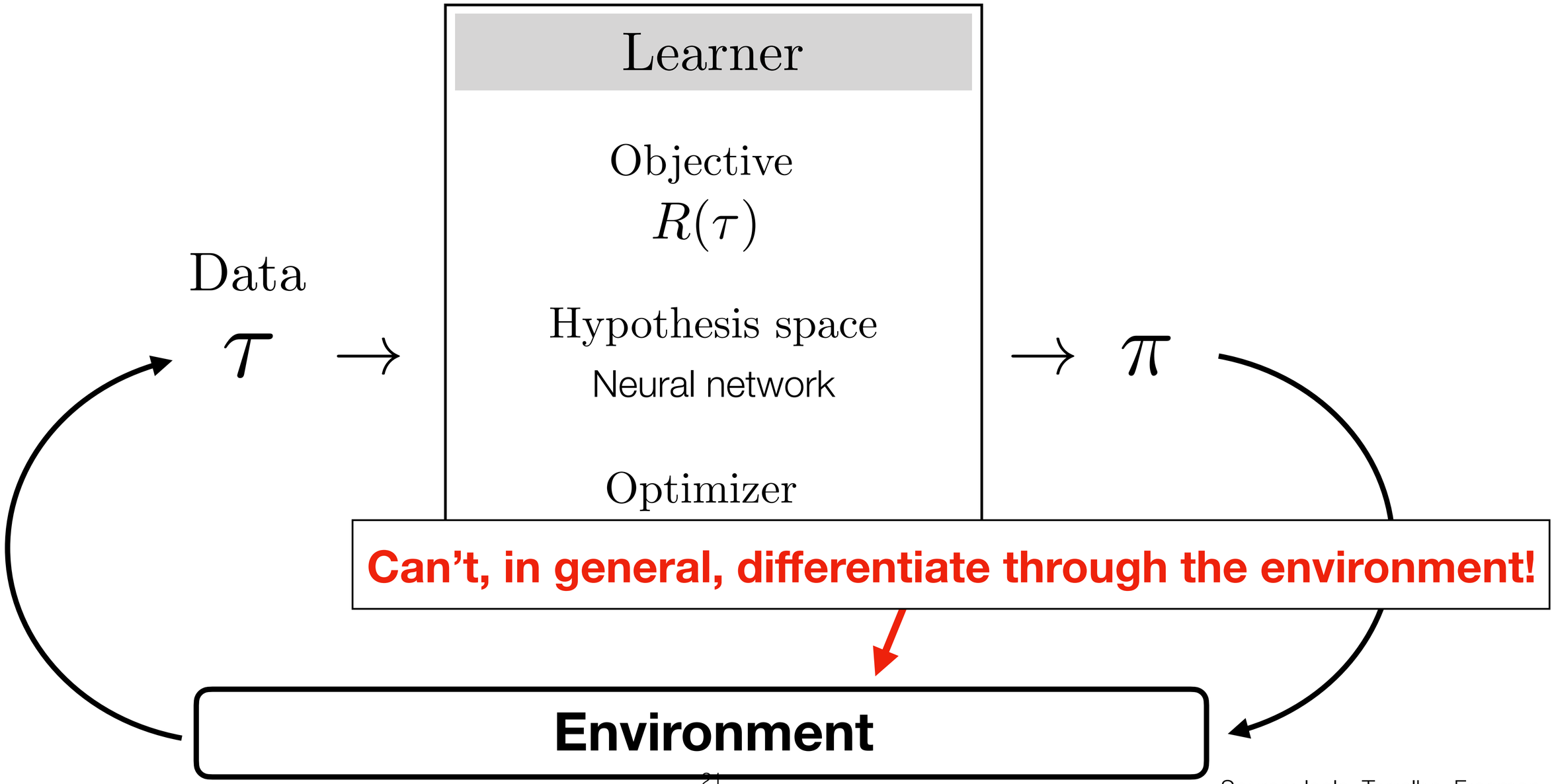
Discounted rewards $R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t, \quad \gamma \in (0, 1)$

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

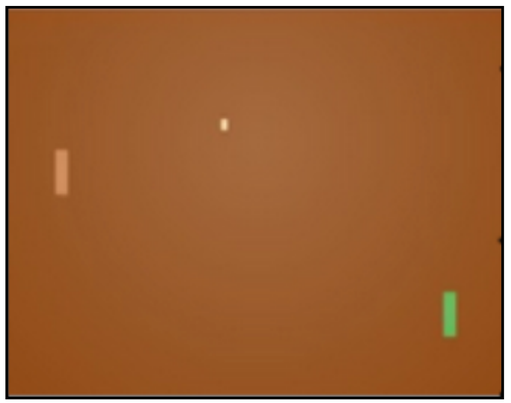
Reinforcement learning



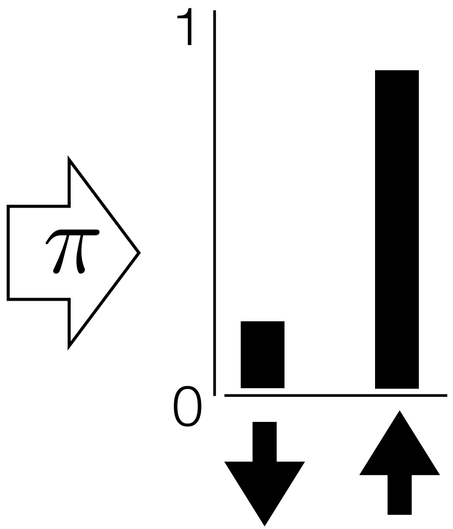
Source: Isola, Torralba, Freeman

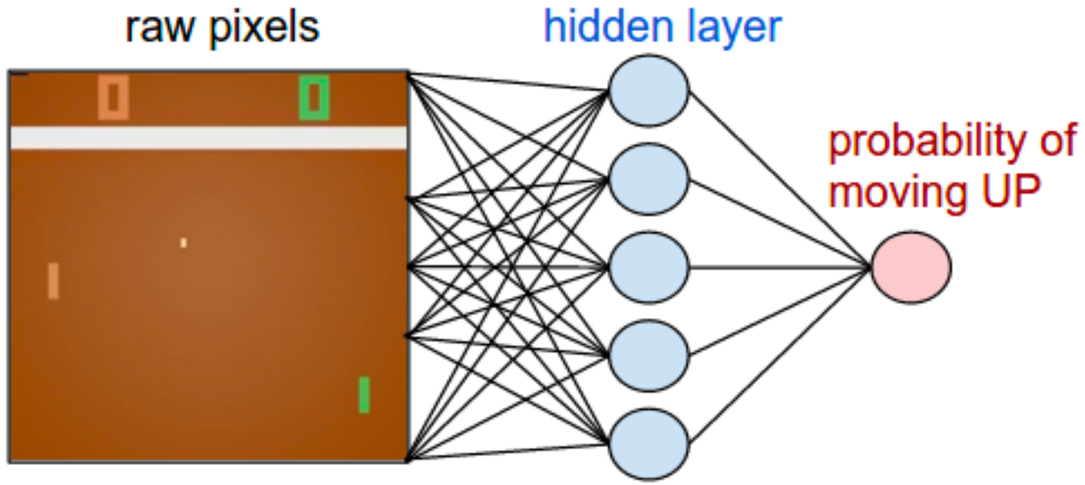
Environment is not differentiable! — How to optimize?

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



Pong

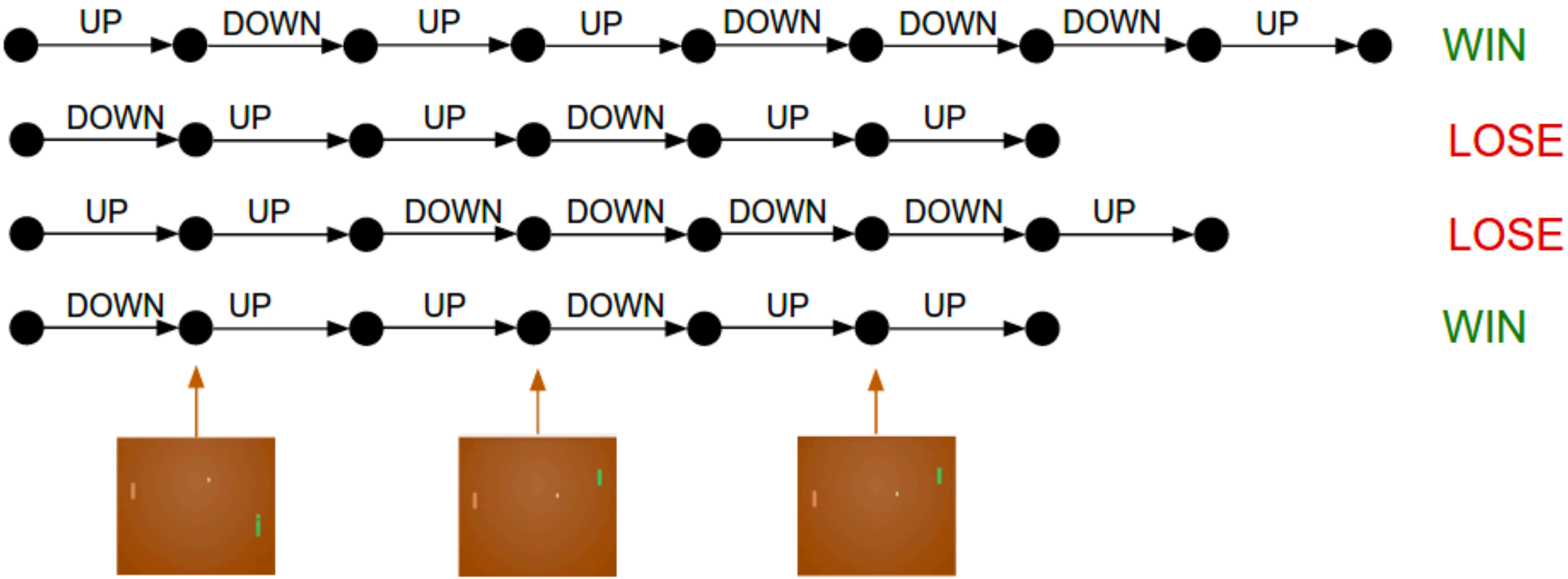




[Adapted from Andrej Karpathy: http://karpathy.github.io/2016/05/31/rl/]

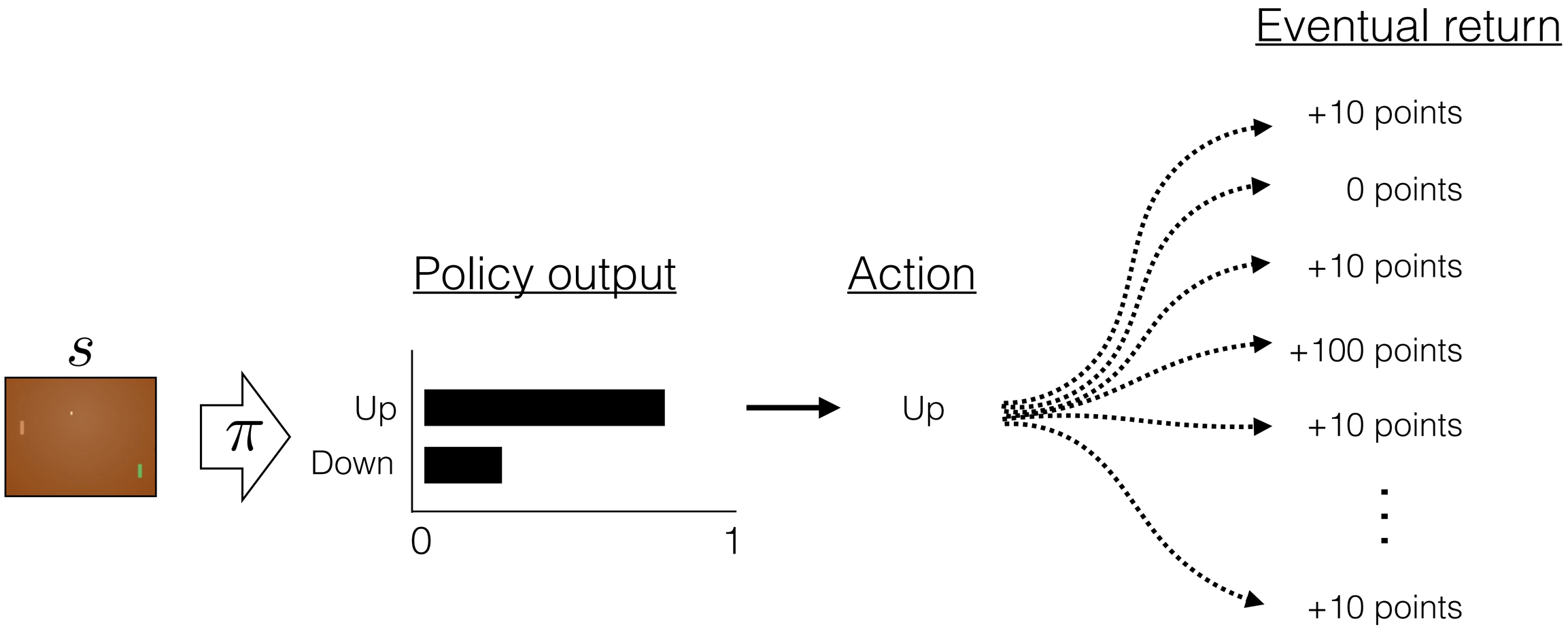
Source: Isola, Torralba, Freeman

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



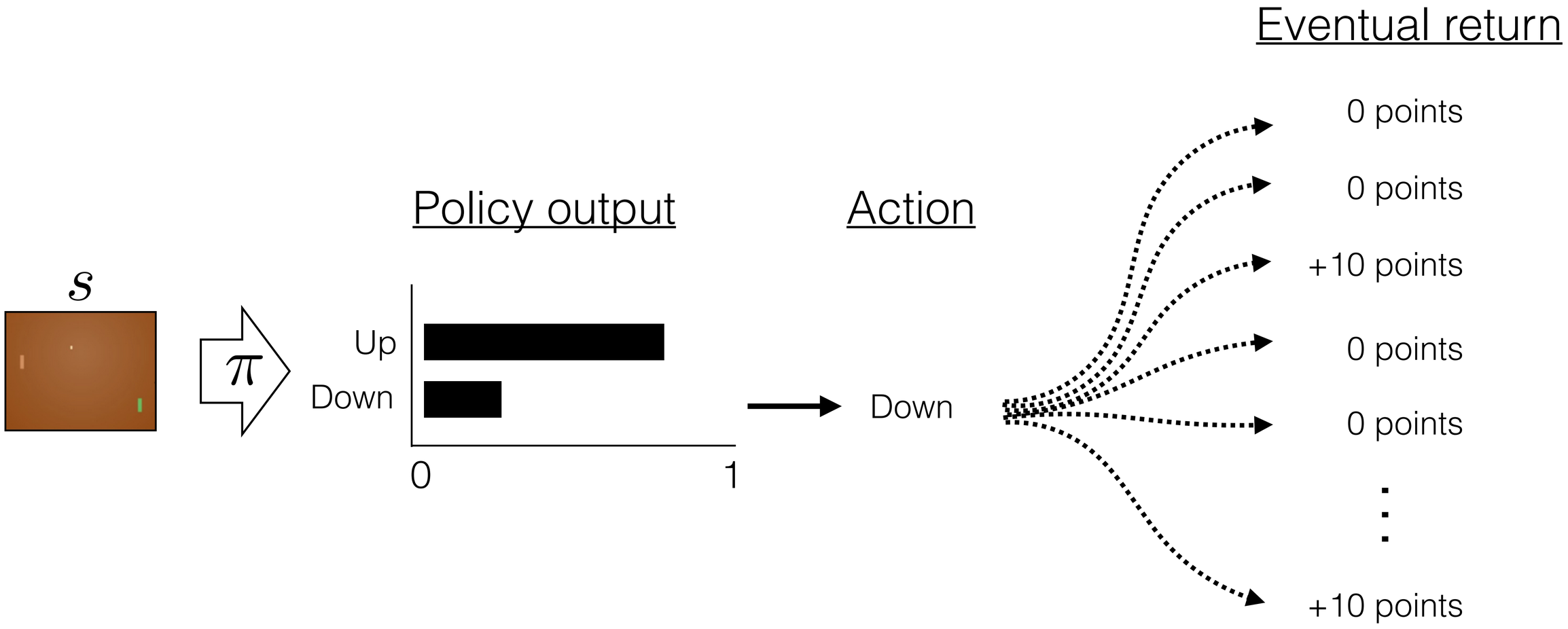
[Adapted from Andrej Karpathy: http://karpathy.github.io/2016/05/31/rl/]

Source: Isola, Torralba, Freeman



$\pi(a|s)$ = 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.
  - Estimate gradients, do gradient ascent

Policy gradient

Looks like 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?

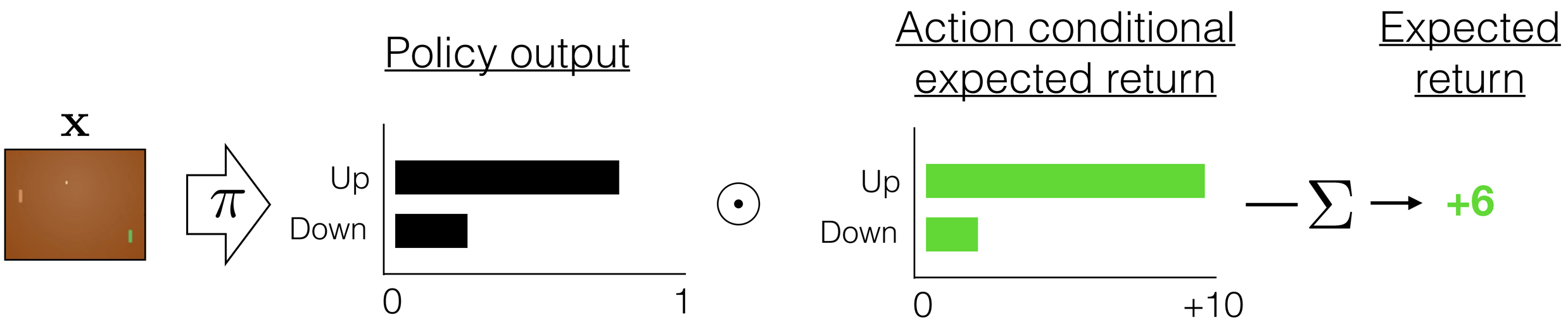
$$r(\tau) = \sum_{t=0}^{T} R(s_t)$$

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

Policy gradient

1. Start with an arbitrary initial policy.
2. **Roll out** this *stochastic* policy many 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

Policy gradient

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

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

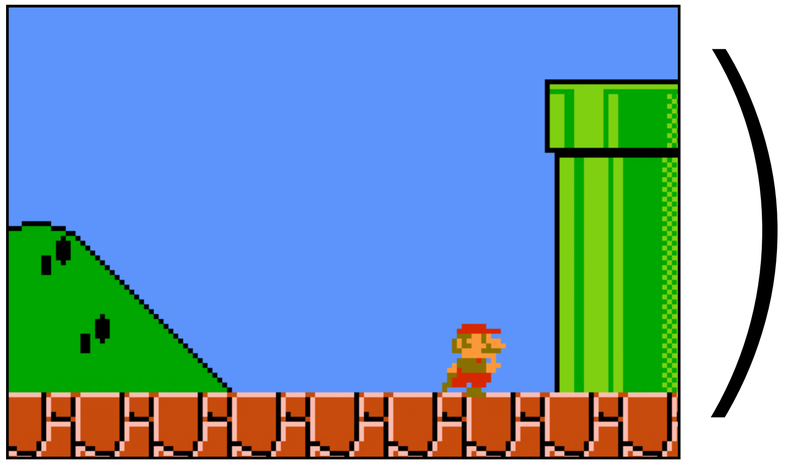
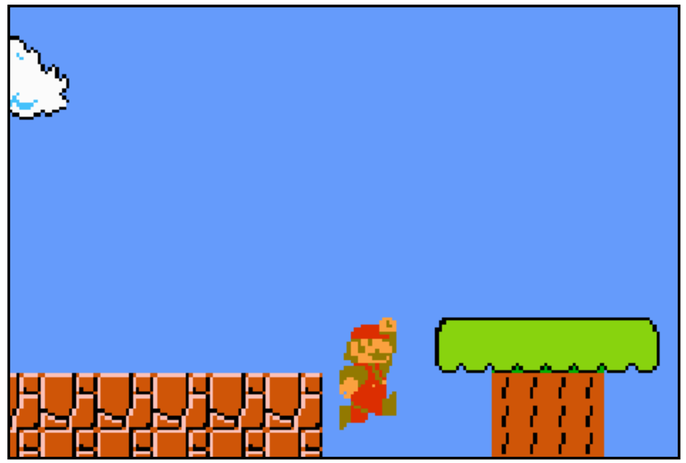
- *All* actions become more likely if the reward is high.
- Doesn't do *credit assignment*.

Slide adapted from R. Grosse and J. Ba

How good is a state?

Value function: expected future reward from starting in *s*.

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t \geq 0} \gamma^t r_t \mid s_0 = s, \pi\right]$$

V^{π}  vs. V^{π} 

- One advantage is credit assignment: we know which state/action was useful.
- Sometimes more sample efficient, and updates have less variance.

How good is a state-action pair?

- Could we *learn* the value function and use it to choose actions?
  - We need more than that. You'd also need to know the dynamics, i.e. what state you'd end up with if you took each action.

- Instead, learn **action-value function** (or **Q function**).

$$Q(s, a) = \mathbb{E}\left[\sum_{t \geq 0} R_t \mid s_0 = s, a_t = a\right]$$

- Optimal action for a state: $\operatorname*{argmax}_a Q(s, a)$

Adapted from R. Grosse and J. Ba

Finding a good Q function

- Good Q function should satisfy a recurrence relation called the Optimal Bellman Equation:

Quality of state/action pair

Where will I end up?

What if I take the *very best* next action?

$$Q^*(s,a) = r(s,a) + \gamma \mathbb{E}_{p(s'|s,a)} \left[\max_{a'} Q^*(s_{t+1}, a') \mid s_t = s, a_t = a \right]$$

Finding a good Q function

- Measuring the Bellman error for Q:

$$r(s_t, a_t) + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)$$

- Approximate Q with a neural net $Q(s, a; \theta)$. For each episode *i*:

1. Do the policy induced by Q and get a trajectory:

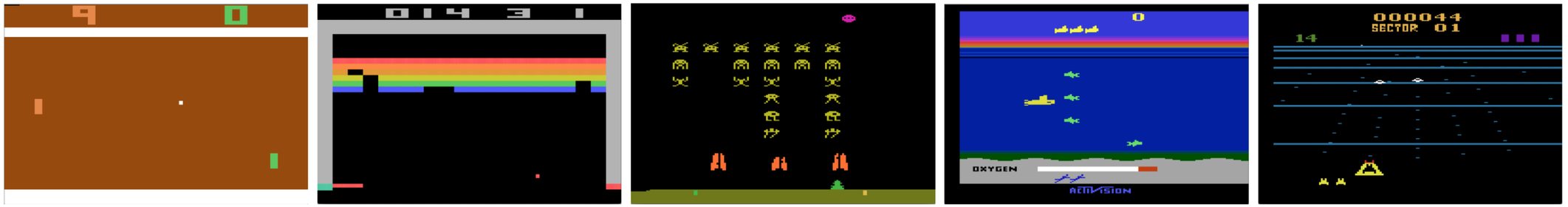
$$\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots)$$

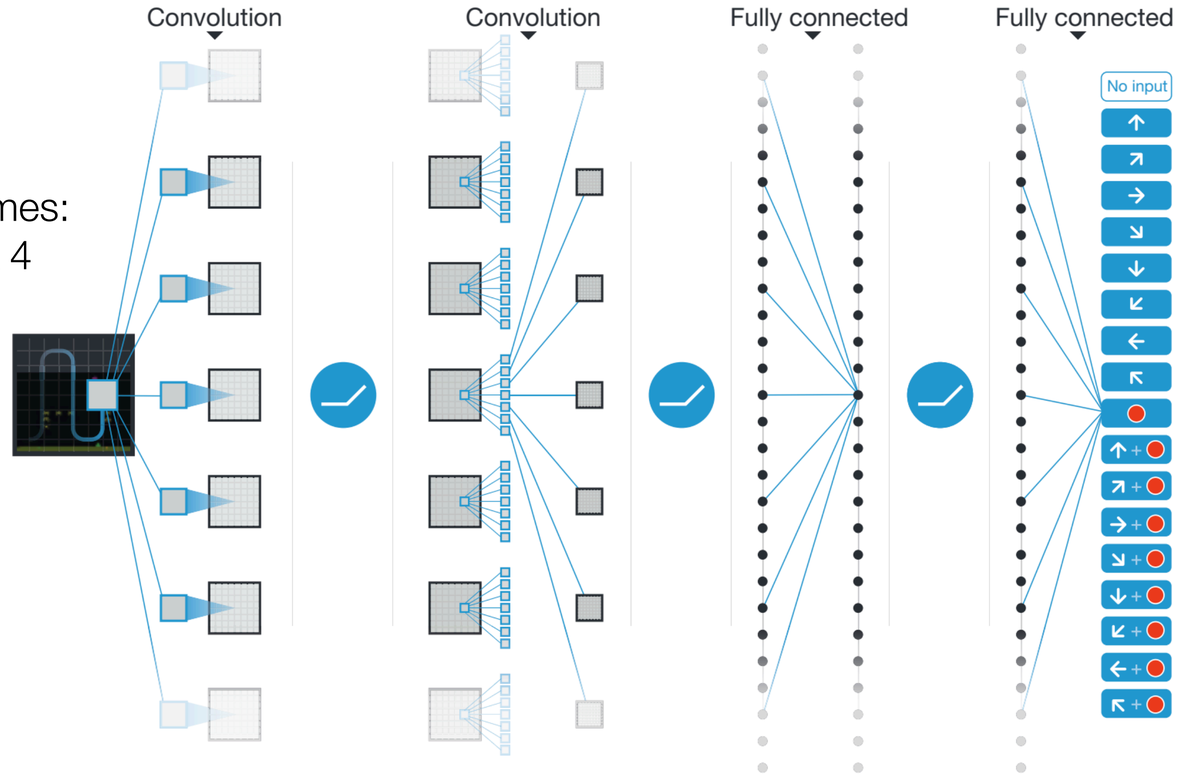
2. Update the parameters using backprop, minimizing approximation error:

$$t_i = r(s_t, a_t) + \gamma \max_a Q(s_{t+1}, a; \theta_{i-1})$$

$$L(\theta_i) = (t_i - Q(s_t, a_t; \theta_i))^2$$

Playing Atari games with deep Q-learning





Input:
4 last frames:
84 x 84 x 4

Output:
Q(s, a) for each action *a*

5-layer CNN

[Mnih et al., “Playing Atari with Deep Reinforcement Learning”, 2013]

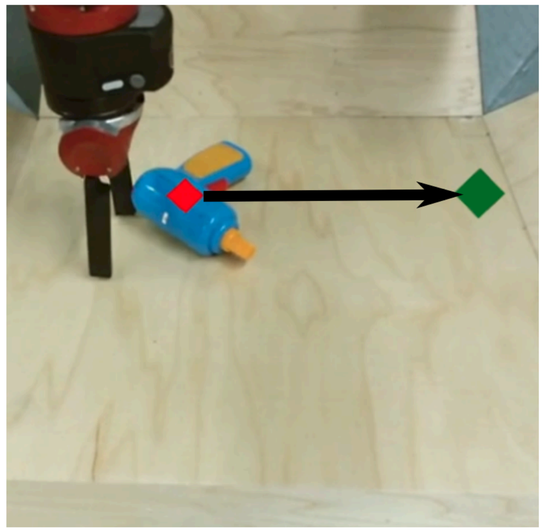
Playing Atari games

Model-based control

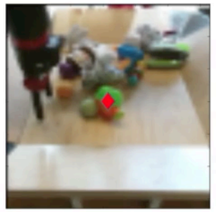
- Learn the **dynamics** of the environment: $p(s_{t+1} \mid s_t, a_t)$
- Where do I end up in the future if I perform this action?
- If states are images, we want to **predict the future** after you do an action

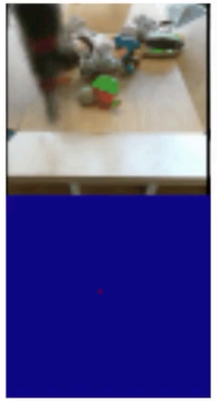
Model-based control

- Learn the **dynamics** of the environment: $p(s_{t+1} \mid s_t, a_t)$
- Where do I end up in the future if I perform this action?
- If states are images, we want to **predict the future** after you do an action



Pushing task

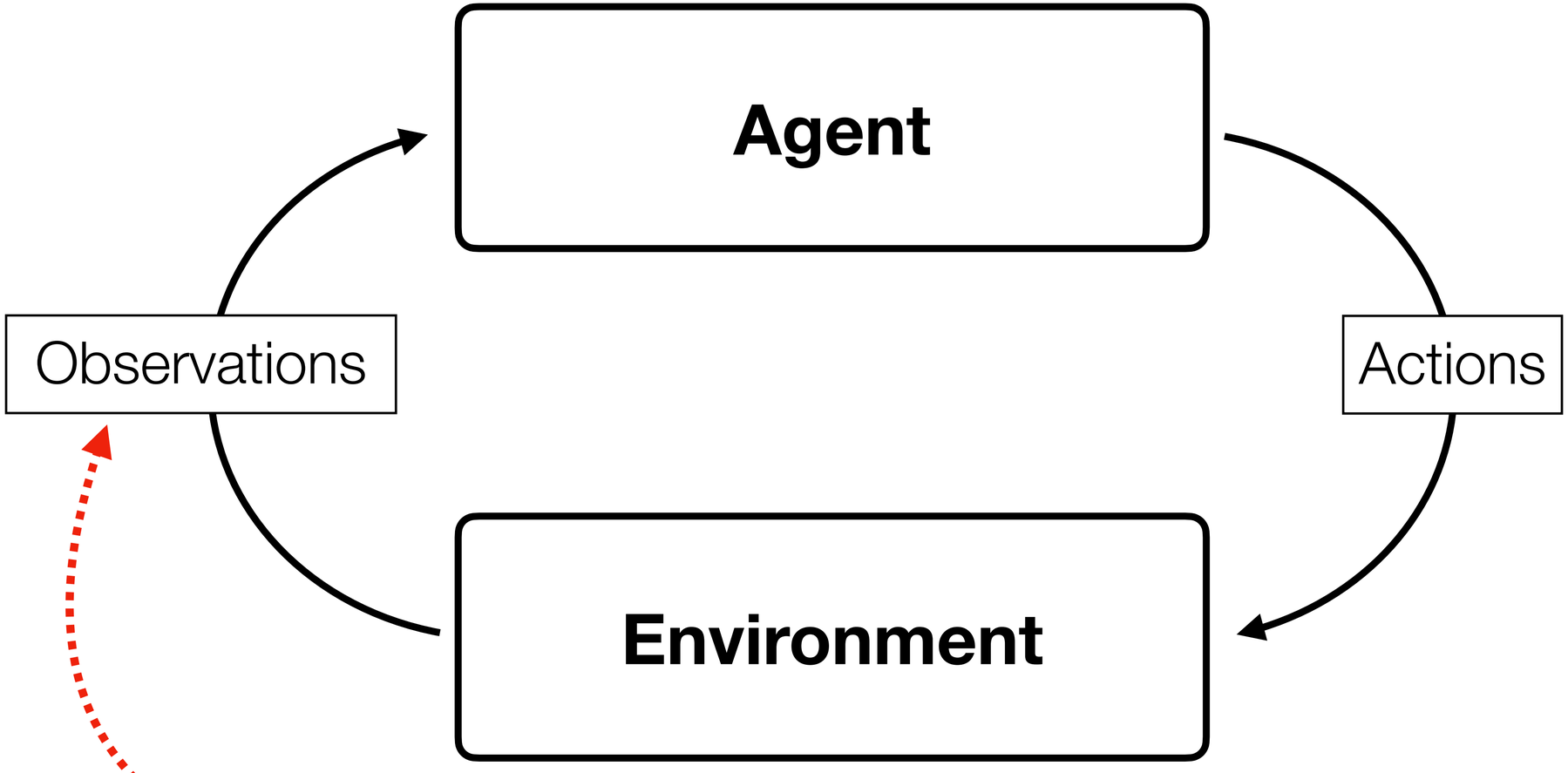




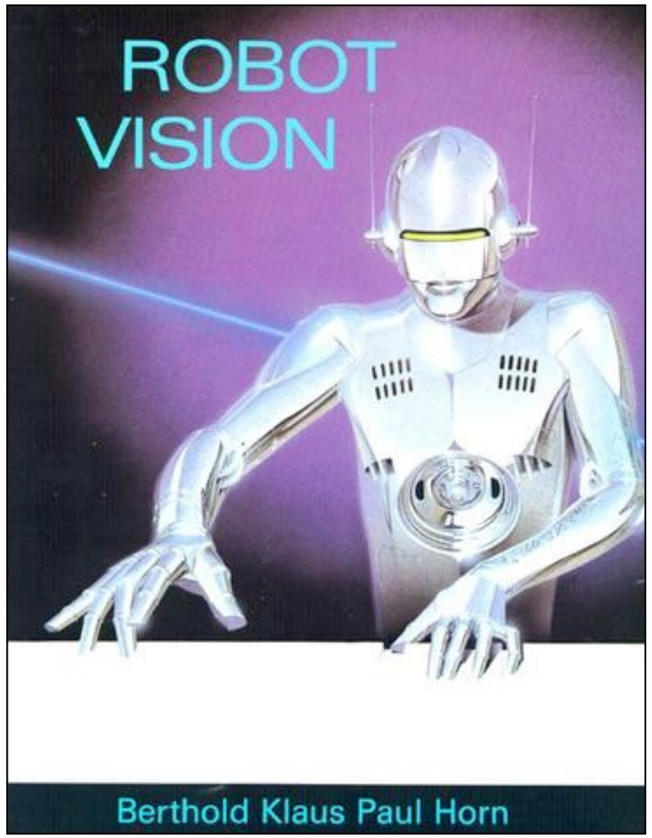
Video prediction

From [Ebert et al., "Self-Supervised Visual Planning with Temporal Skip Connections" 2017]

Intelligent agents



Why vision?



Source: Isola, Torralba, Freeman

Why vision?

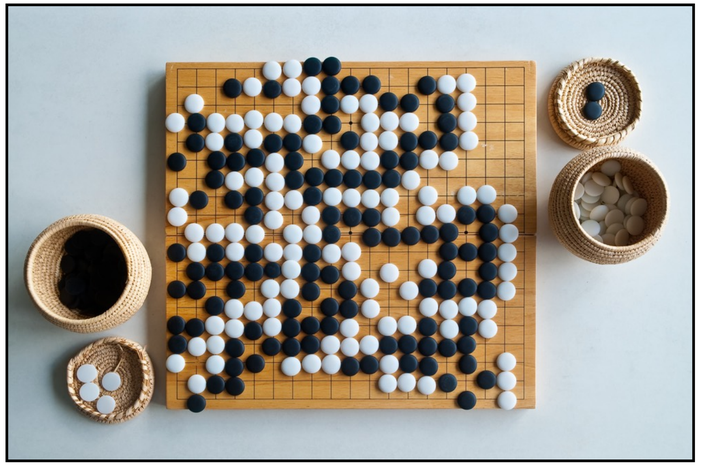
1. Human-like intelligence (and animal-like), often relies heavily on vision

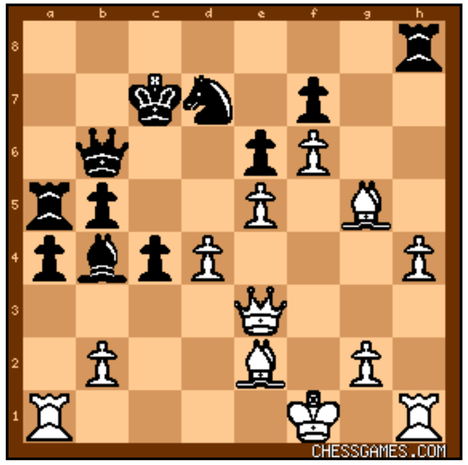
We already know it works well!

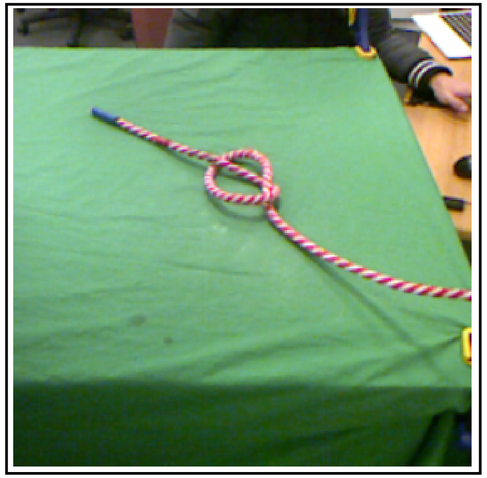
Source: Isola, Torralba, Freeman

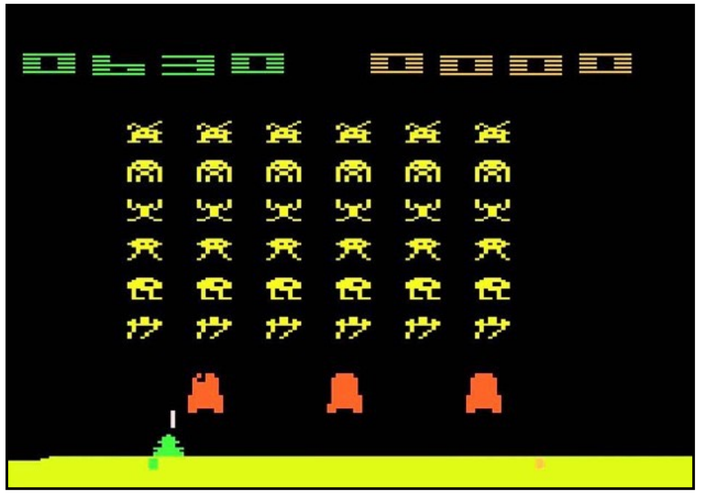
Why vision?

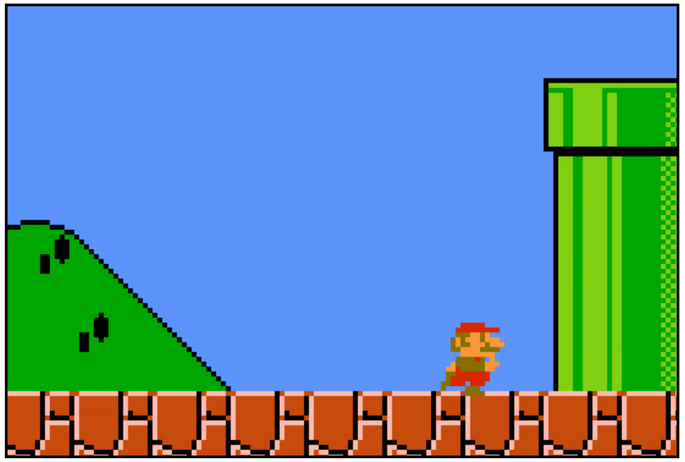
2. Universal interface







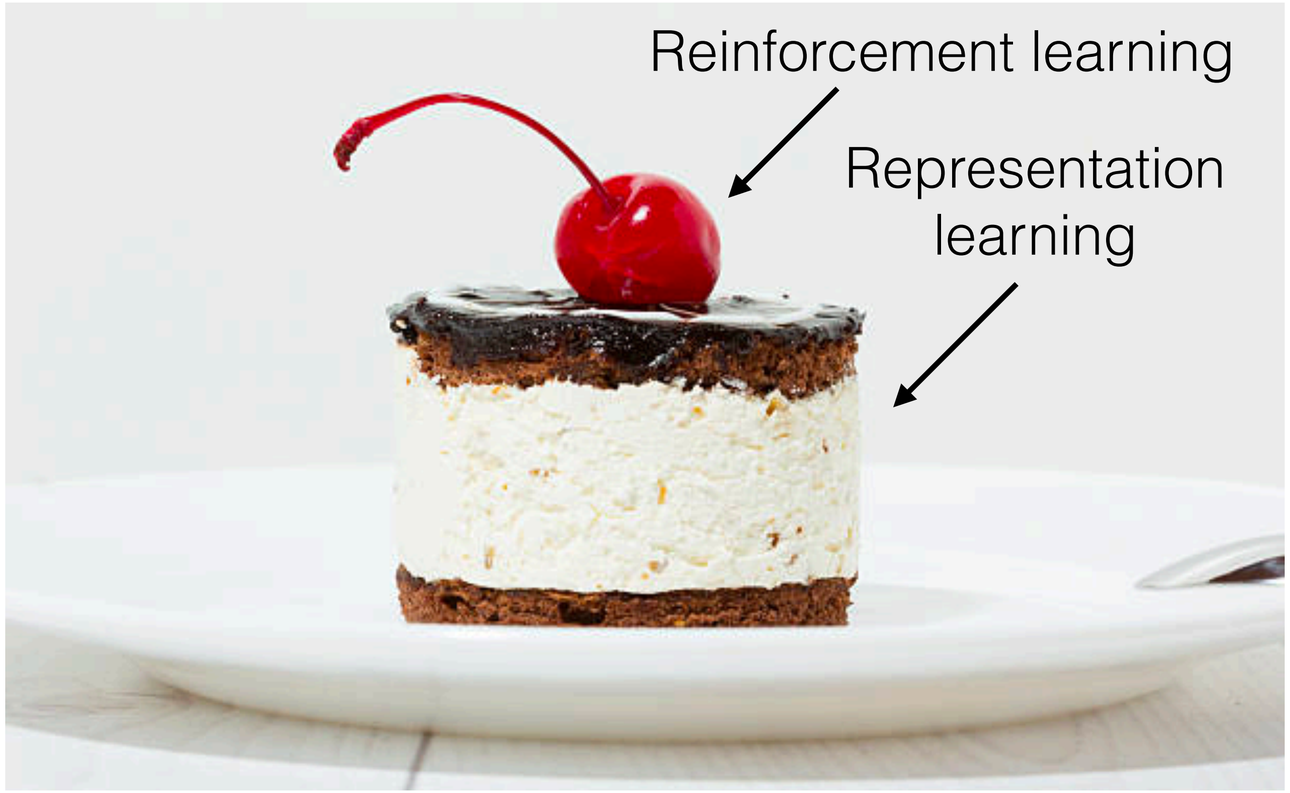




Source: Isola, Torralba, Freeman

Importance of perception

A hypothesis: 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

Next class: recent advances in 3D