Microsoft

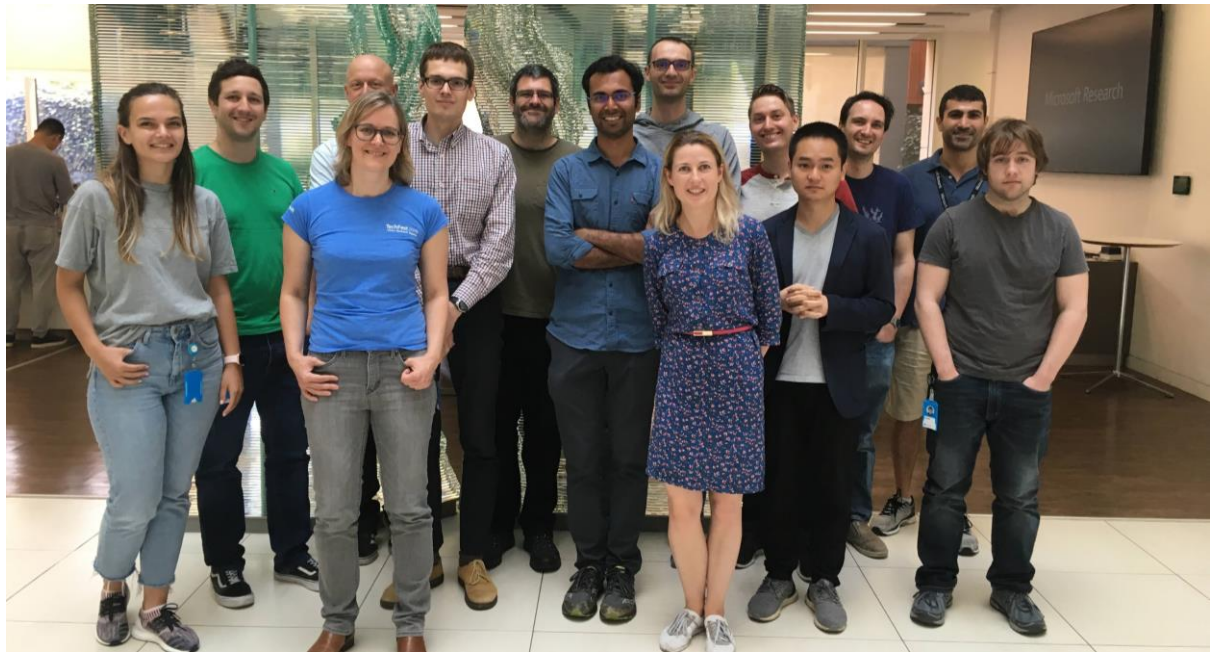
Reinforcement Learning

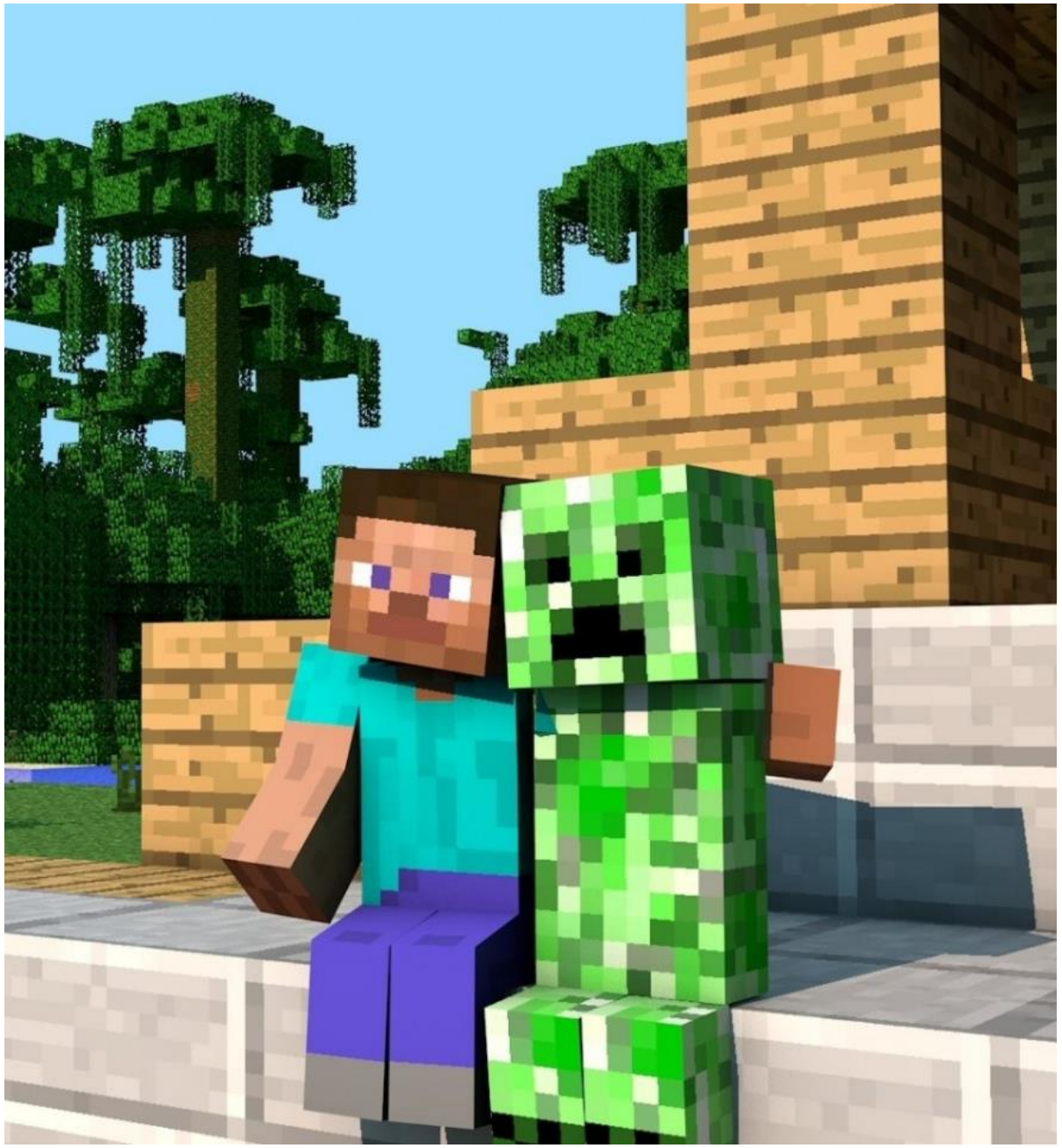
Lecture at MLSS 2019
Part I

Katja Hofmann

Game Intelligence
Microsoft Research, Cambridge, UK
aka.ms/gameintelligence

@katjahofmann


Microsoft Research

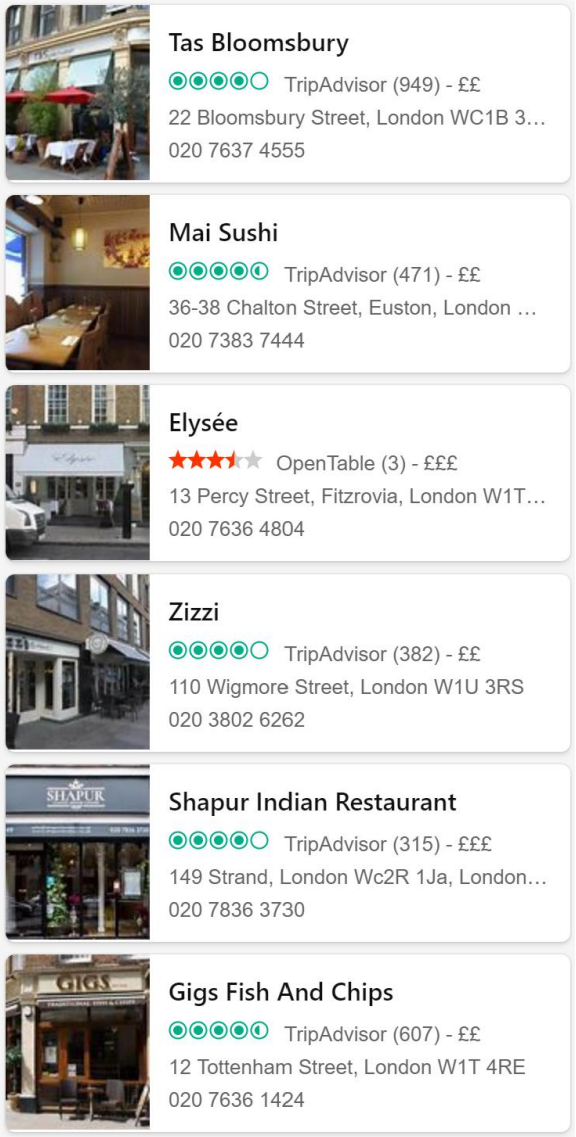


Reinforcement Learning (RL)

= the science and engineering of decision making and learning under uncertainty

= a type of machine learning that models learning from experience in a wide range of applications

Decision making under uncertainty is a common problem



Tas Bloomsbury
TripAdvisor (949) - ££
22 Bloomsbury Street, London WC1B 3…
020 7637 4555

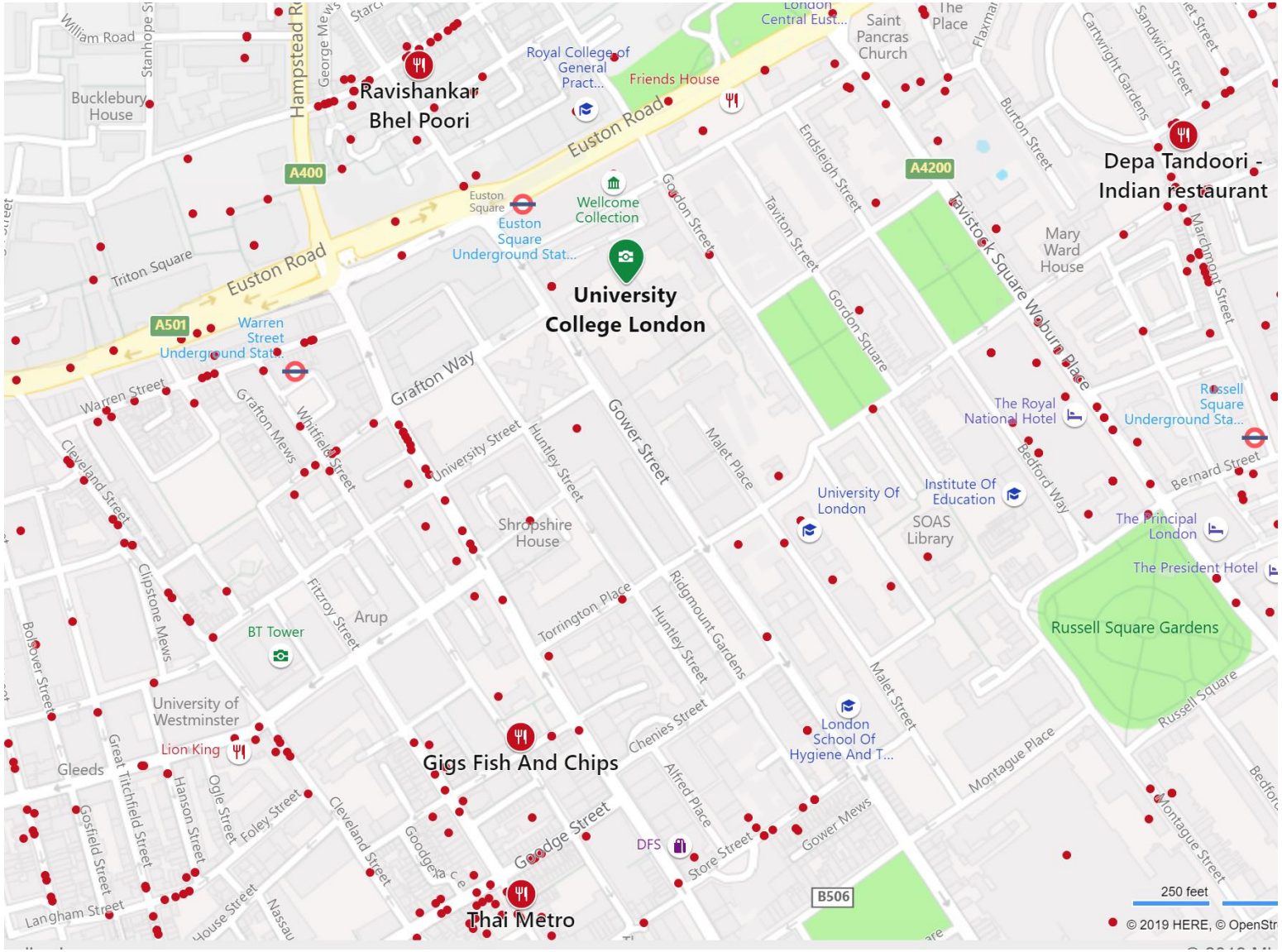
Mai Sushi
TripAdvisor (471) - ££
36-38 Chalton Street, Euston, London …
020 7383 7444

Elysée
OpenTable (3) - £££
13 Percy Street, Fitzrovia, London W1T…
020 7636 4804

Zizzi
TripAdvisor (382) - ££
110 Wigmore Street, London W1U 3RS
020 3802 6262

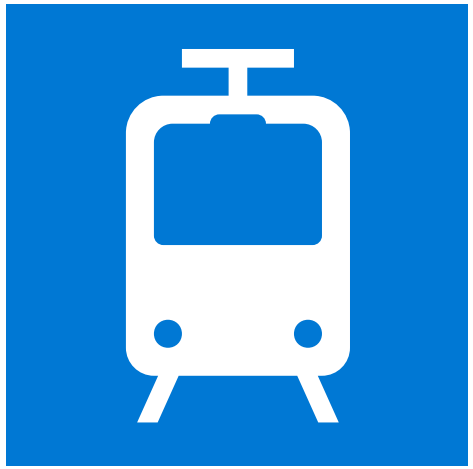
Shapur Indian Restaurant
TripAdvisor (315) - £££
149 Strand, London Wc2R 1Ja, London…
020 7836 3730

Gigs Fish And Chips
TripAdvisor (607) - ££
12 Tottenham Street, London W1T 4RE
020 7636 1424



Decision making under uncertainty is a common problem






Agenda

Lecture 1

Case Studies (and a bit of history)

Formalizing RL

Exploration and Exploitation

RL Approaches 1:
Policy Gradient Methods

Lecture 2

RL Approaches 2:
Temporal Differences, Q-Learning

Advanced Topics: Multi-task RL,
Learning from Demonstration

Tutorial

Implementing a Deep Q Agent in Minecraft

Case Studies (and a bit of history)

RL can model a vast range of problems

Example problems that
motivated RL research

Animal Learning

Games

Optimal Control

Lindquist, J. 1962, "*Operations of a hydrothermal electric system: A multistage decision process.*" Transactions of the American Institute of Electrical Engineers.

Mario Pereira, Nora Campodónico, & Rafael Kelman, 1998, "*Long-term hydro scheduling based on stochastic models.*" *EPSOM* 98.

Photo by Magda Ehlers from Pexels

Long-term consequences in optimal control

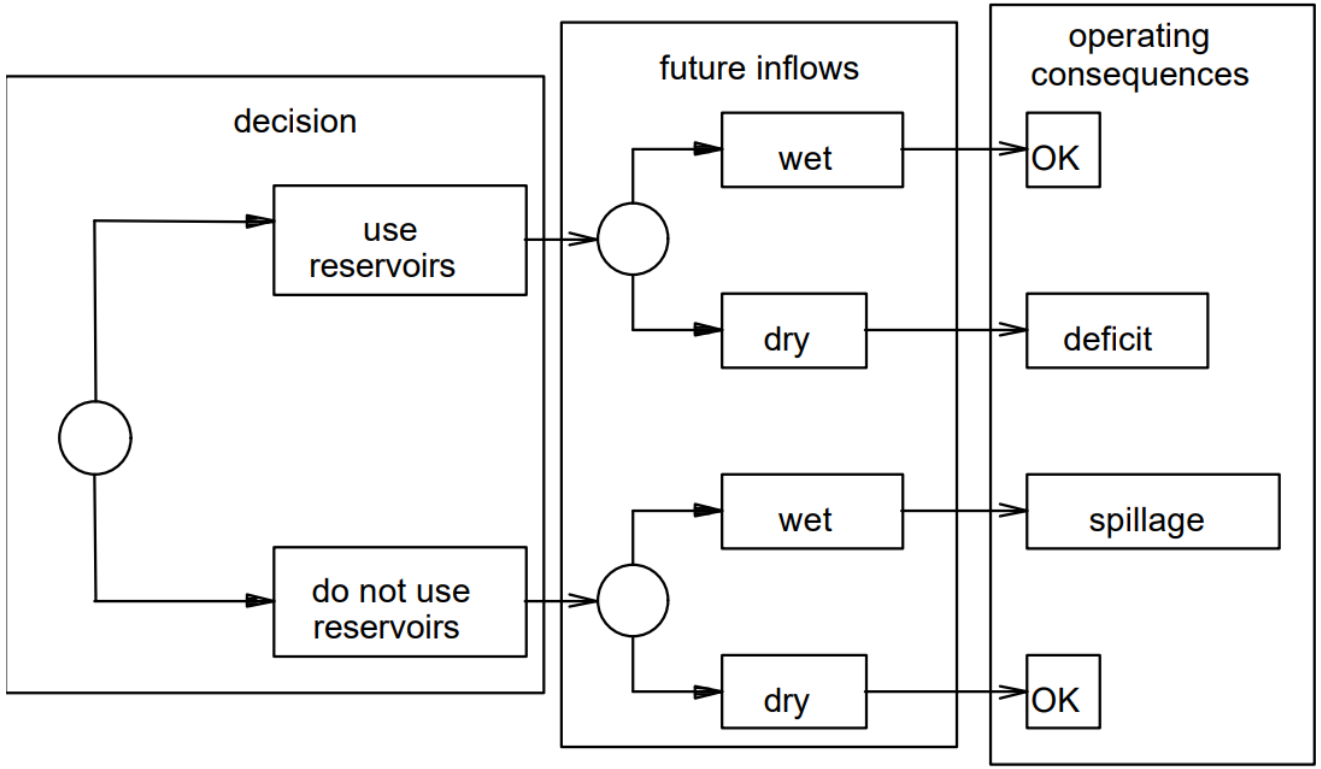
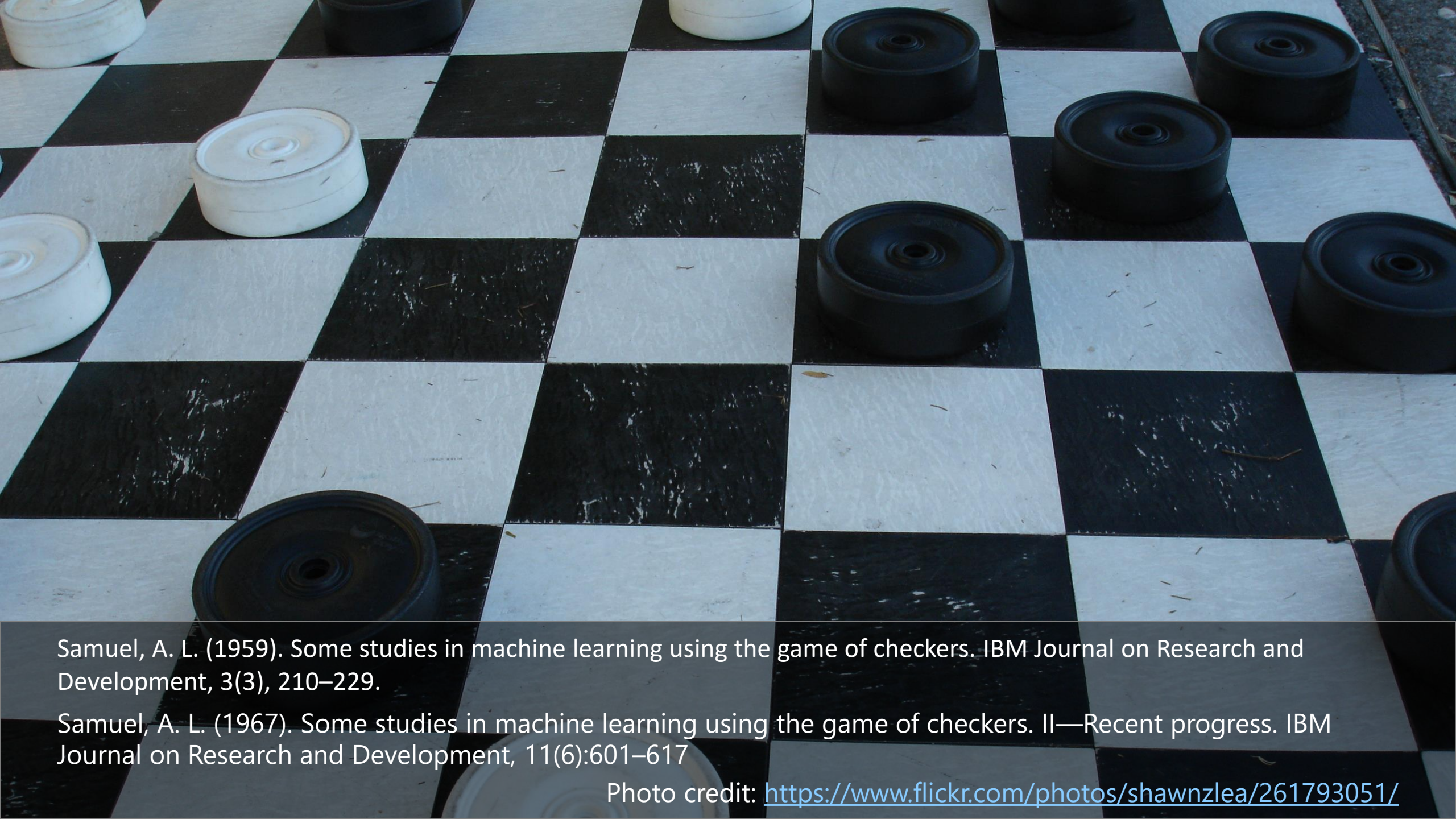


Figure 2.1 - Decision Process for Hydrothermal Systems

Figure from:
Mario Pereira, Nora Campodónico, & Rafael Kelman. "Long-term hydro scheduling based on stochastic models." *EPSOM* 98.



Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. IBM Journal on Research and Development, 3(3), 210–229.

Samuel, A. L. (1967). Some studies in machine learning using the game of checkers. II—Recent progress. IBM Journal on Research and Development, 11(6):601–617

Photo credit: https://www.flickr.com/photos/shawnzlea/261793051/

Samuel's Checkers Player

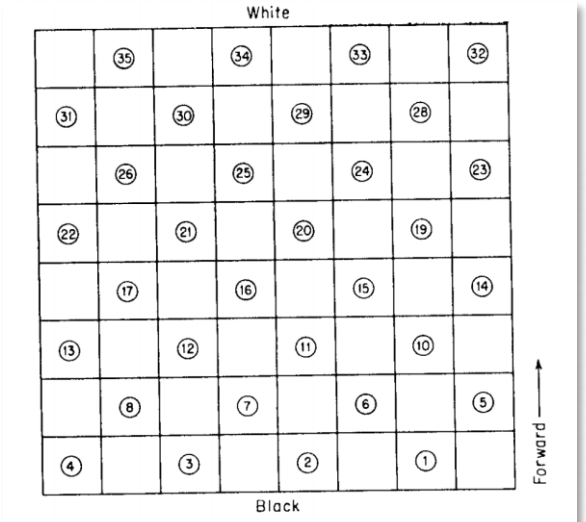


Figure 5. Checkerboard notation for internal computations.

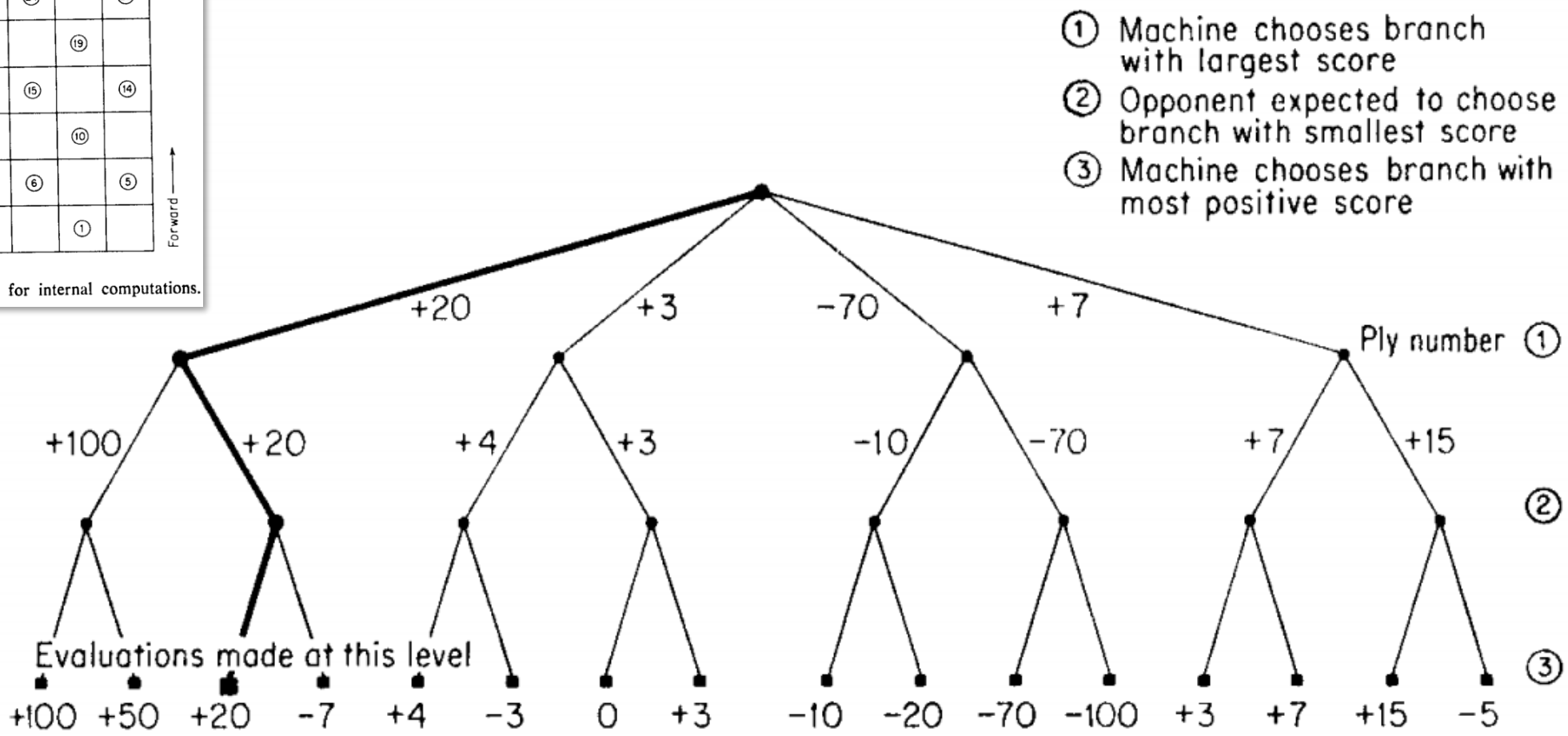


Figure 2. Simplified diagram showing how the evaluations are backed up through the "tree" of possible moves to arrive at the best next move. The evaluation process starts at (3).

Schultz, Wolfram, Peter Dayan, and P. Read Montague. "A neural substrate of prediction and reward." *Science* 275.5306 (1997): 1593-1599.

Photo credit: https://www.flickr.com/photos/scorius/750037290

RL as a valuable tool for modelling neurological phenomena

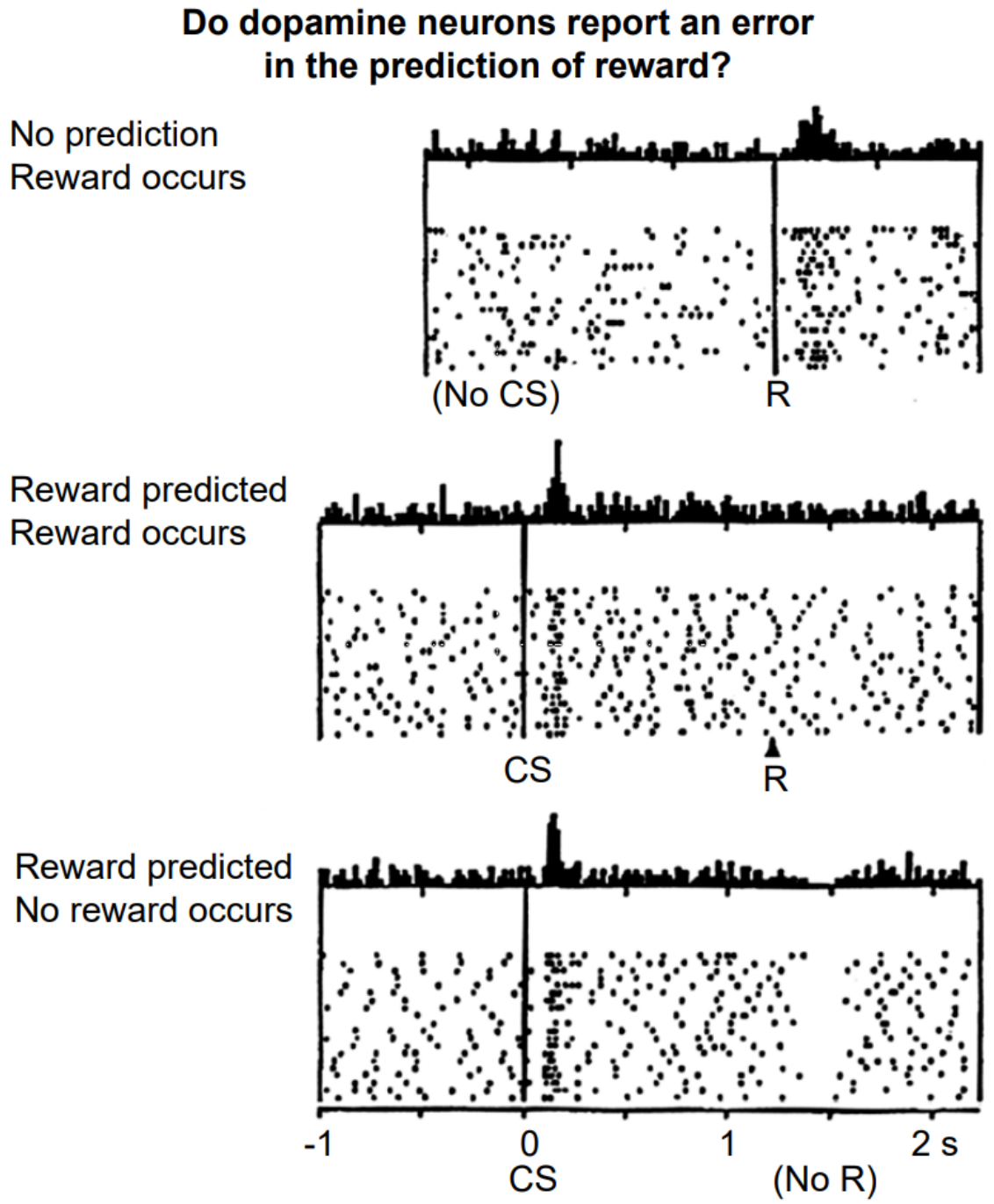


Figure from: Schultz, Wolfram, Peter Dayan, and P. Read Montague. "A neural substrate of prediction and reward." *Science* 1997.

Further Reading

White, D. J. (1985). Real applications of Markov decision processes. Interfaces, 15(6).

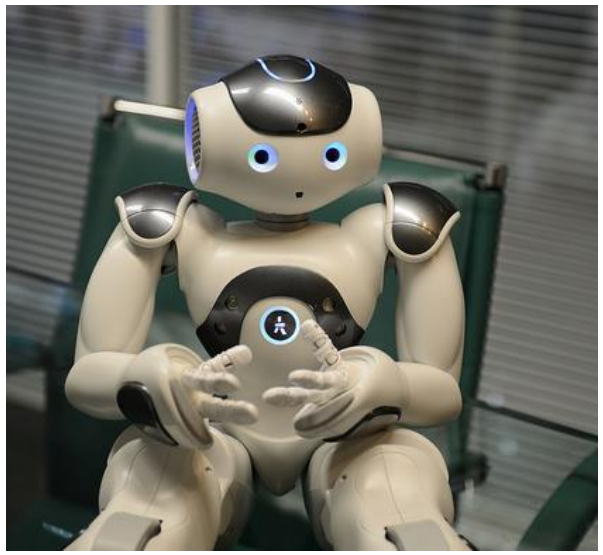
White, D. J. (1988). Further real applications of Markov decision processes. Interfaces, 18(5).

Maia, Tiago V., and Michael J. Frank. "From reinforcement learning models to psychiatric and neurological disorders." *Nature neuroscience* 14.2 (2011).

Sutton, R. S., & Barto, A. G. (2017). *Reinforcement learning: An introduction*. MIT press, 2nd Edition. http://incompleteideas.net/book/the-book-2nd.html Chapter 1, 14-16

Formalizing RL

In RL – agent interacts with an environment

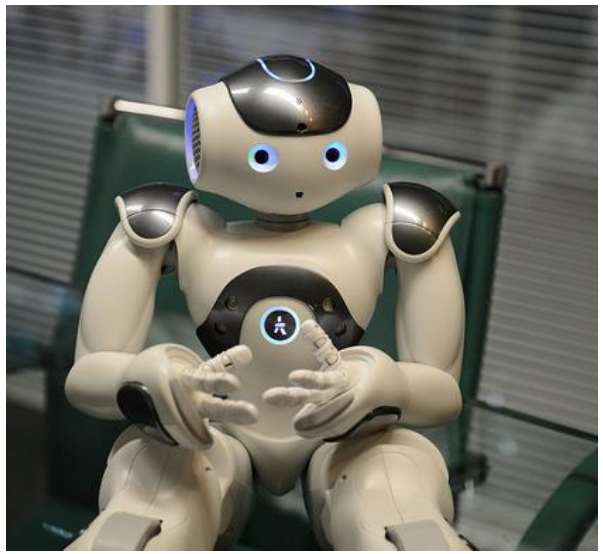


agent

environment

Photo credit: https://www.flickr.com/photos/steveonjava/8170183457

In RL – agent interacts with an environment



agent

environment

state $s_0 \in S$

Photo credit: https://www.flickr.com/photos/steveonjava/8170183457

In RL – agent interacts with an environment

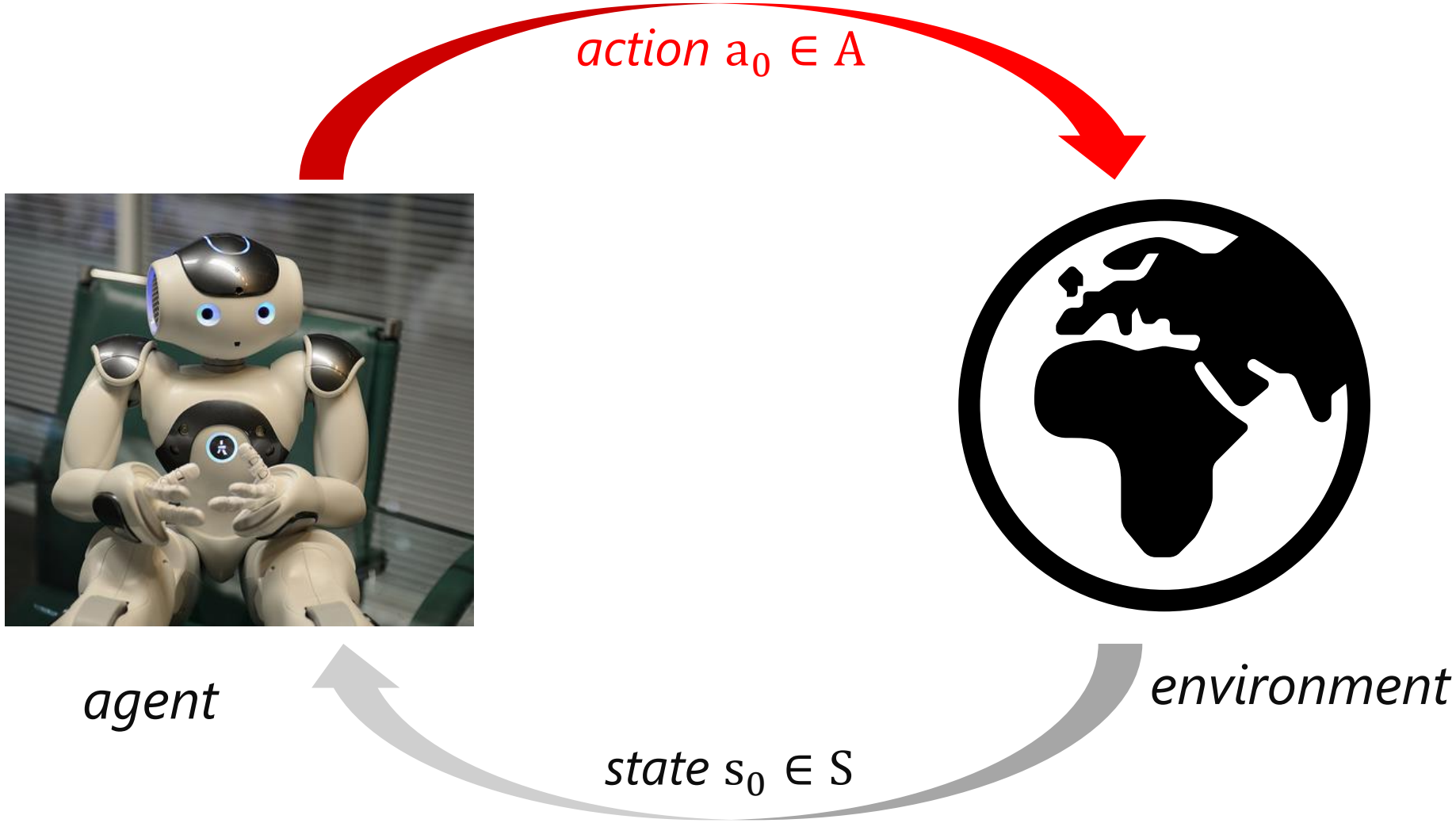


Photo credit: https://www.flickr.com/photos/steveonjava/8170183457

In RL – agent interacts with an environment

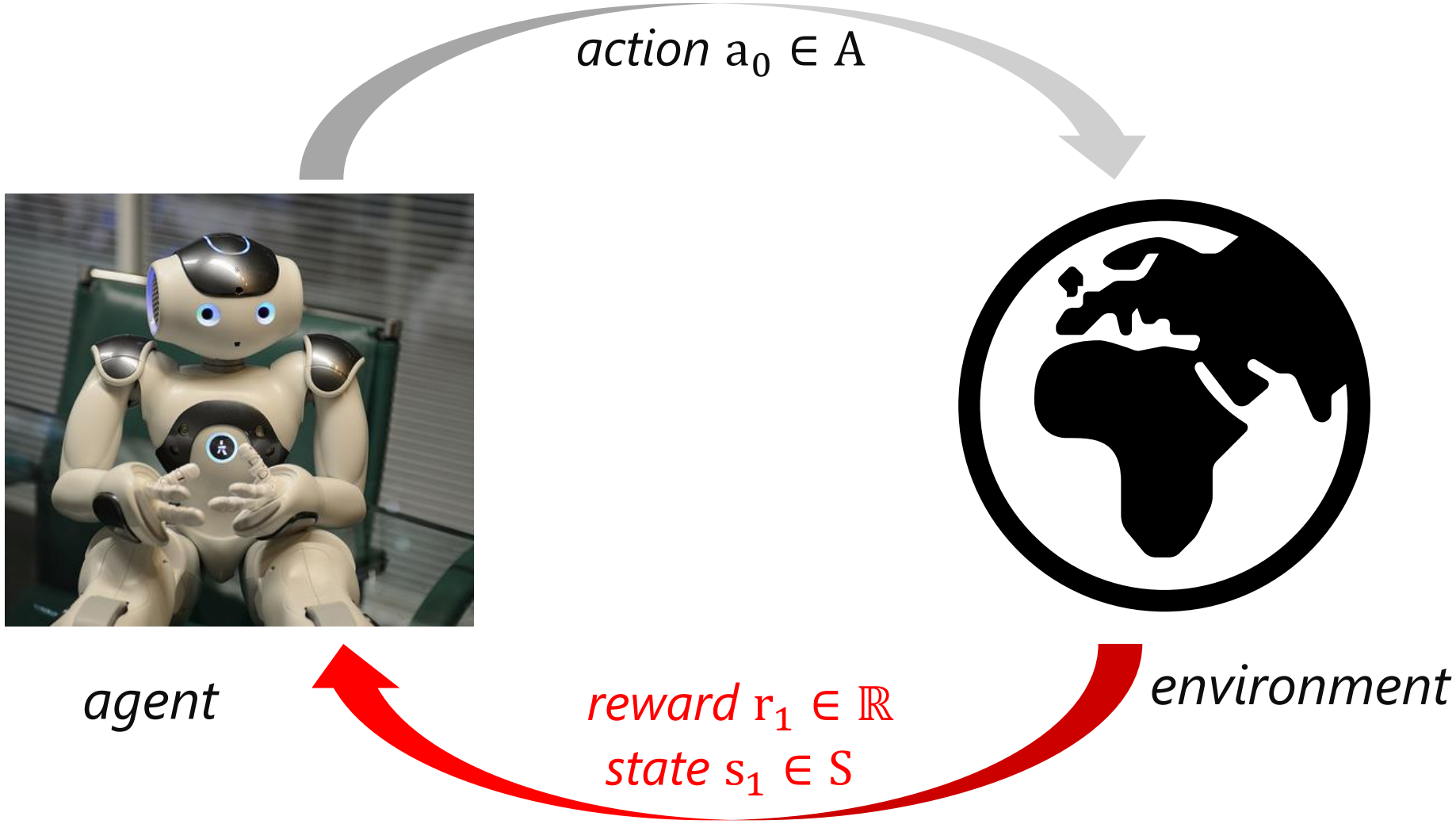


Photo credit: https://www.flickr.com/photos/steveonjava/8170183457

In RL – agent interacts with an environment

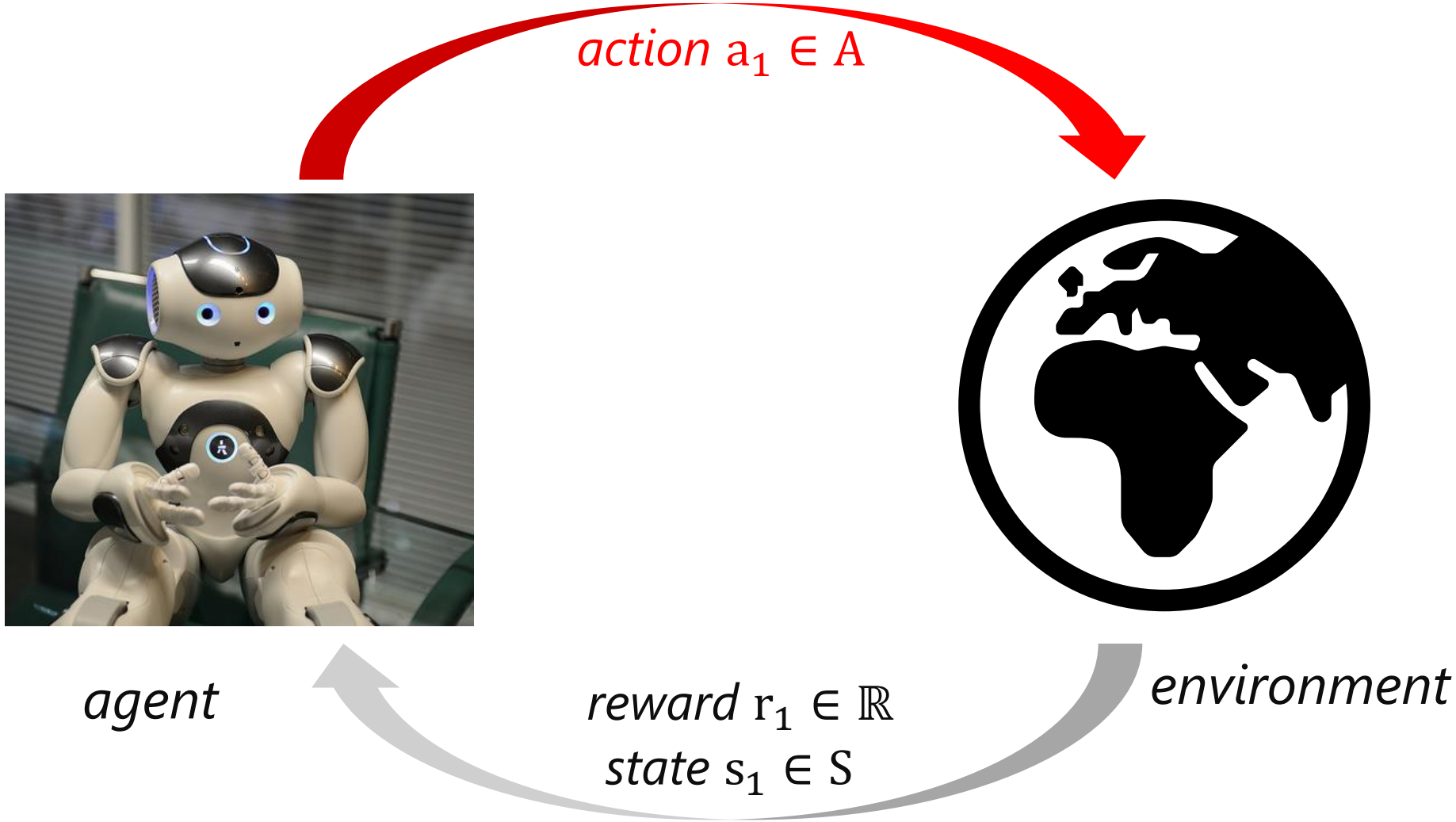


Photo credit: https://www.flickr.com/photos/steveonjava/8170183457

In RL – agent interacts with an environment

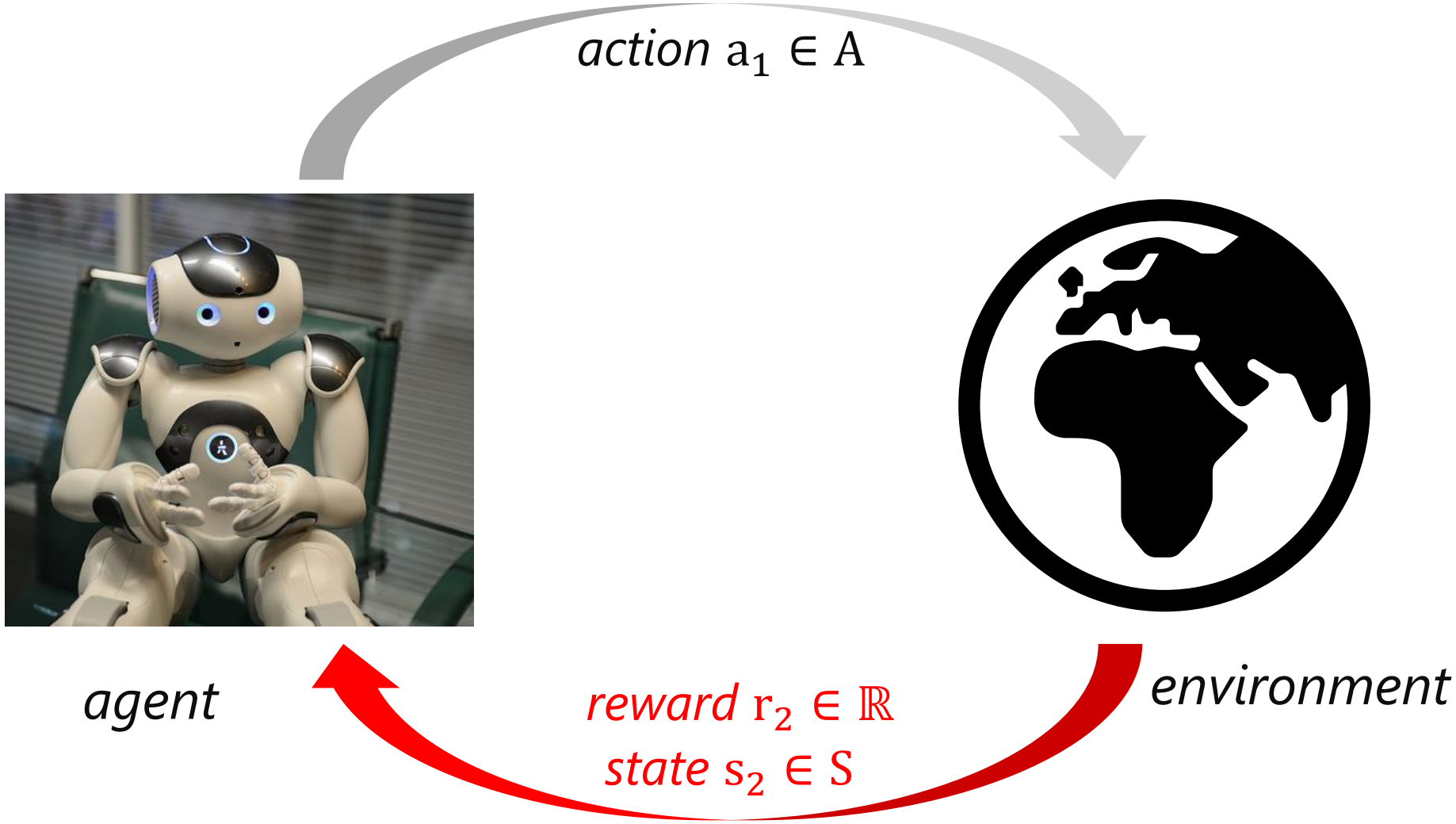


Photo credit: https://www.flickr.com/photos/steveonjava/8170183457

In RL – agent interacts with an environment

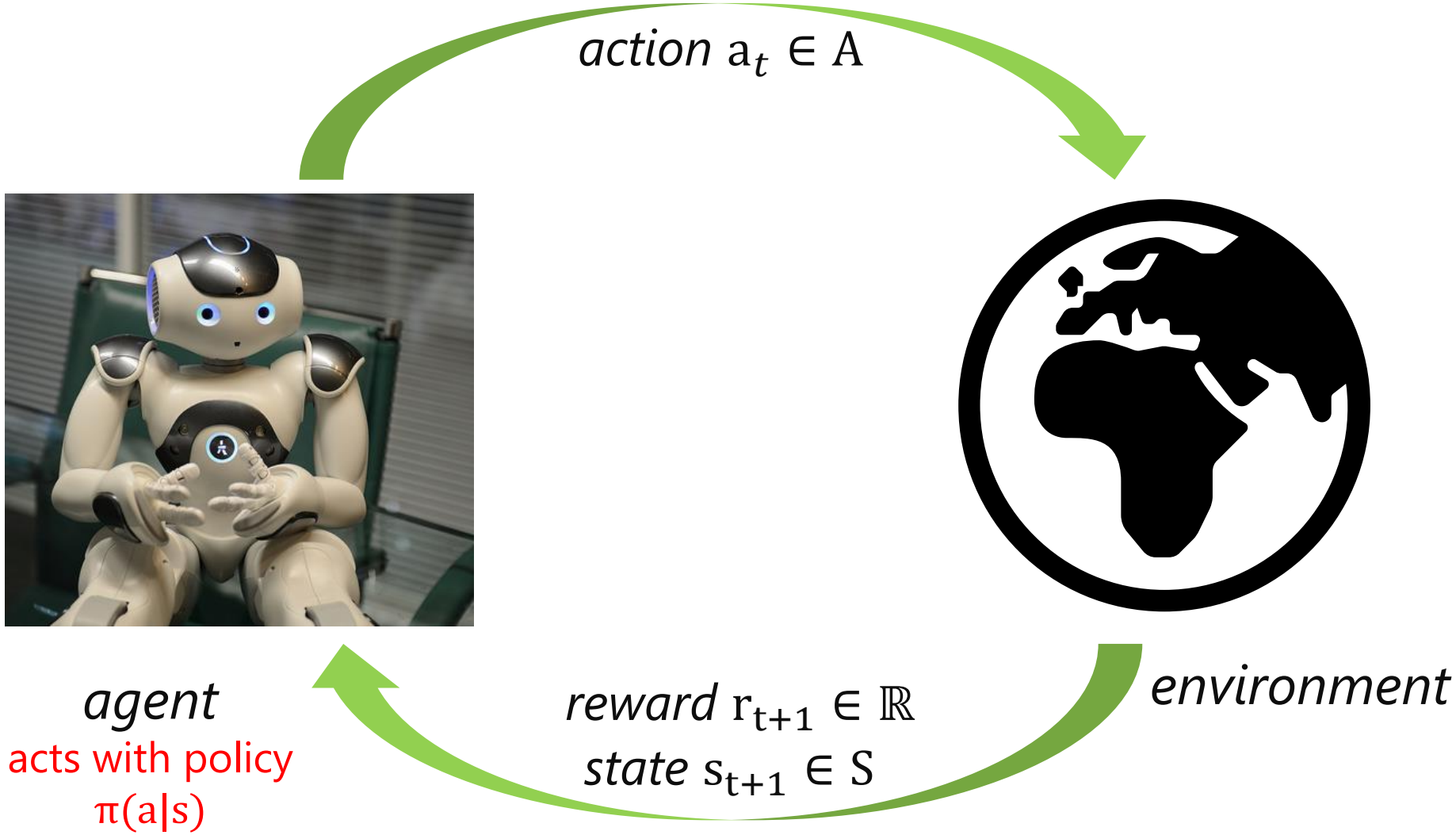
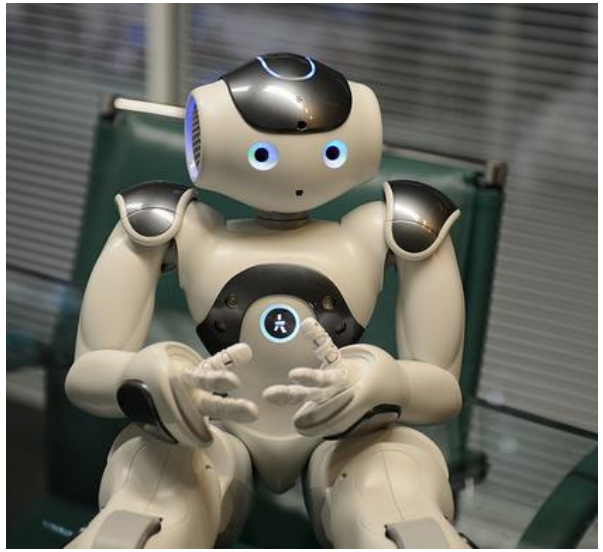


Photo credit: https://www.flickr.com/photos/steveonjava/8170183457

In RL – agent interacts with an environment



action $a_t \in A$

reward $r_{t+1} \in \mathbb{R}$

state $s_{t+1} \in S$

agent

acts with policy $\pi(a|s)$

environment

transition dynamics $p(s_{t+1}|s_t, a_t)$ and reward function $r(r_{t+1}|s_t, a_t)$

Photo credit: https://www.flickr.com/photos/steveonjava/8170183457

Markov Decision Process (MDP)

Defined by $M = (S, A, P, R, \gamma)$

With state space S, action space A, transition dynamics P, reward function R, and discount factor: $\gamma \in (0,1)$

Key assumption: Markov property (dynamics only depend on most recent state and action)

Define goal:

Take actions that maximize (discounted) cumulative return

$$\mathrm{G_t} = \sum_{t=0}^{\infty} \gamma^t r_{t+1}$$

Examples – States, Actions, Rewards

State space

Important modelling choice: how to represent the problem?

Example: hydroelectric power control problem

How could state be represented?

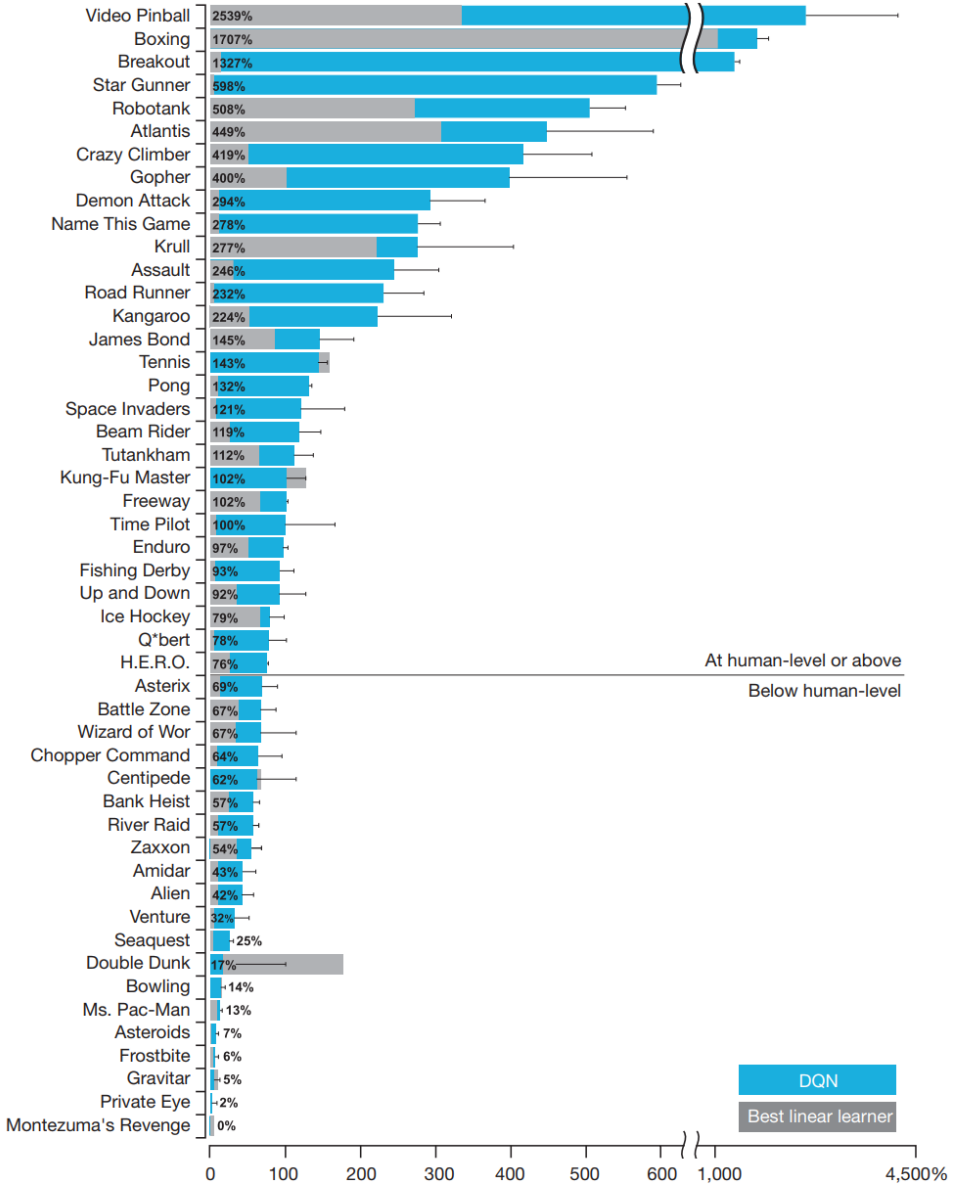
State space

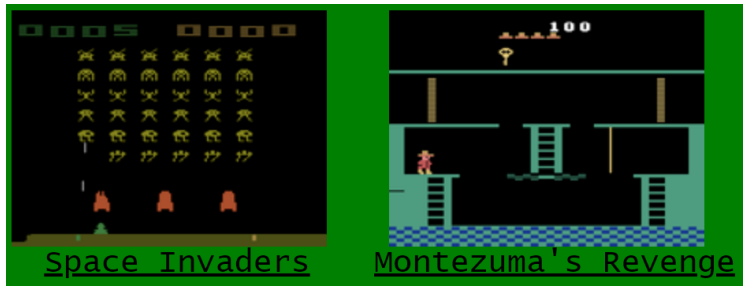
Important modelling choice: how to represent the problem?

Considerations:

- Is the Markov property satisfied?
- (How) can prior (expert) knowledge be encoded?
- Effects on optimal solution?
- Effects on data efficiency?

Mnih et al. results in Atari – a lesson in generality





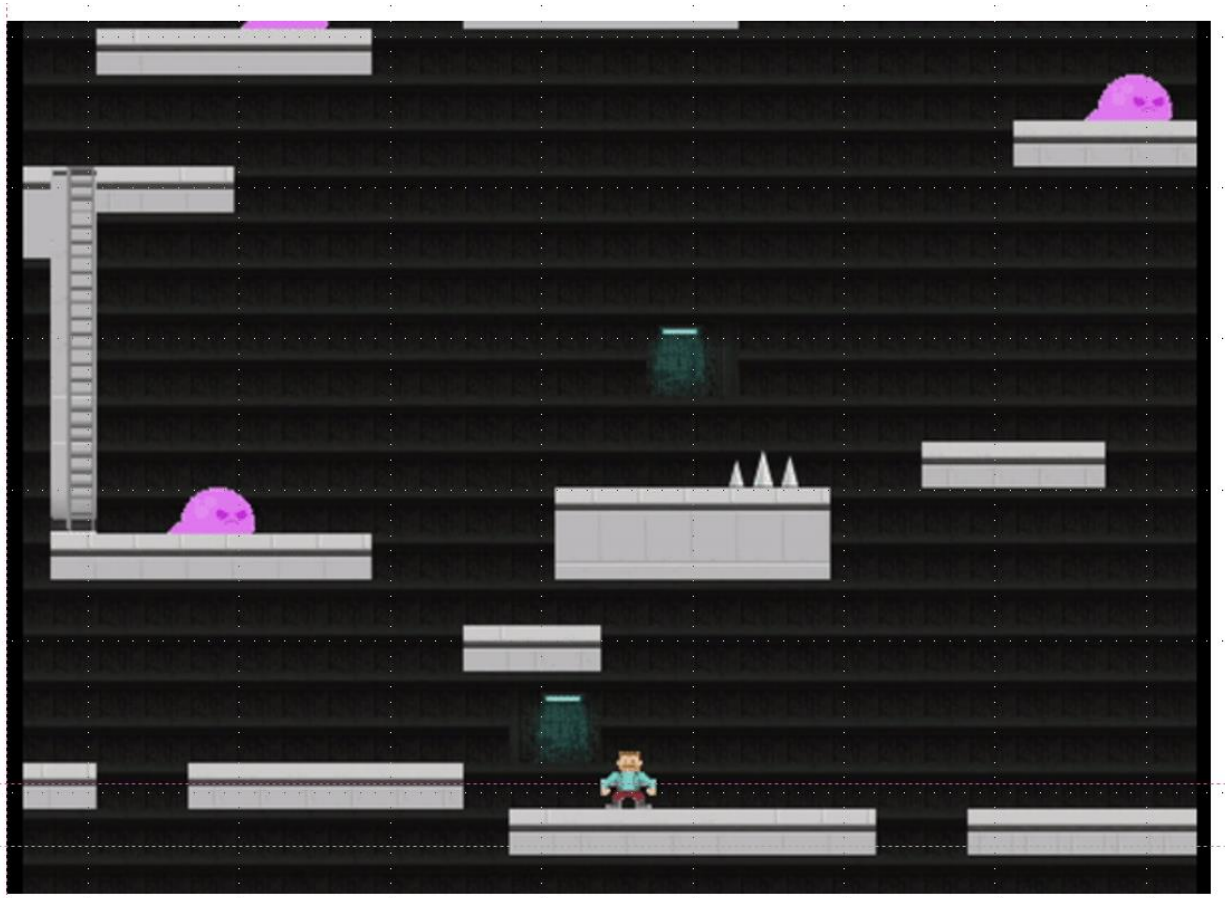
Screenshots from: Kurin, V., Nowozin, S., Hofmann, K., Beyer, L., & Leibe, B. (2017). The Atari Grand Challenge Dataset. http://atarigrandchallenge.com/

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540).

Case Study:

Investigating Human Priors for Playing Video Games

Dubey, R., Agrawal, P., Pathak, D., Griffiths, T. L., & Efros, A. A, ICML 2018.



Original Game

https://rach0012.github.io/humanRL_website/

Case Study:

Investigating Human Priors for Playing Video Games

Dubey, R., Agrawal, P., Pathak, D., Griffiths, T. L., & Efros, A. A, ICML 2018.

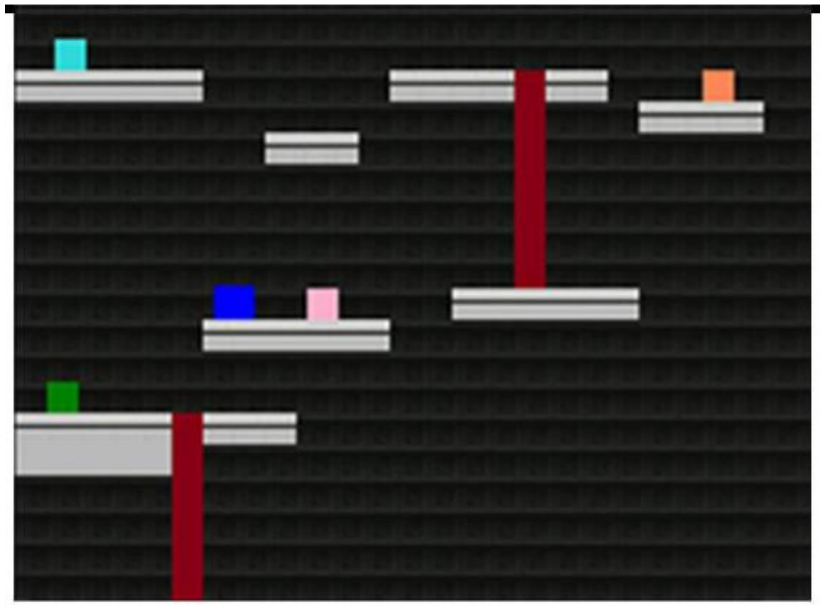
Original Game

https://rach0012.github.io/humanRL_website/

Case Study:
Investigating Human Priors for Playing Video Games

Dubey, R., Agrawal, P., Pathak, D., Griffiths, T. L., & Efros, A. A, ICML 2018.

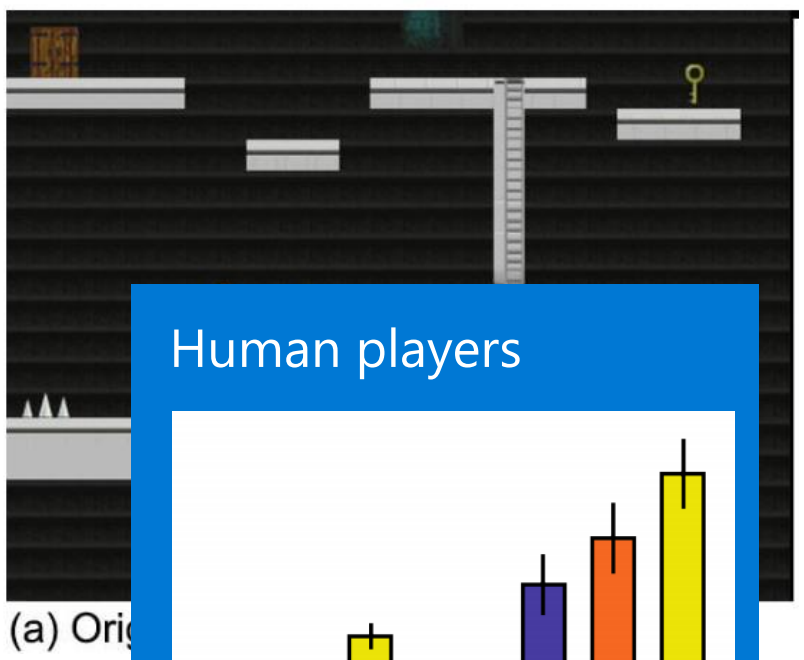
(a) Original Game



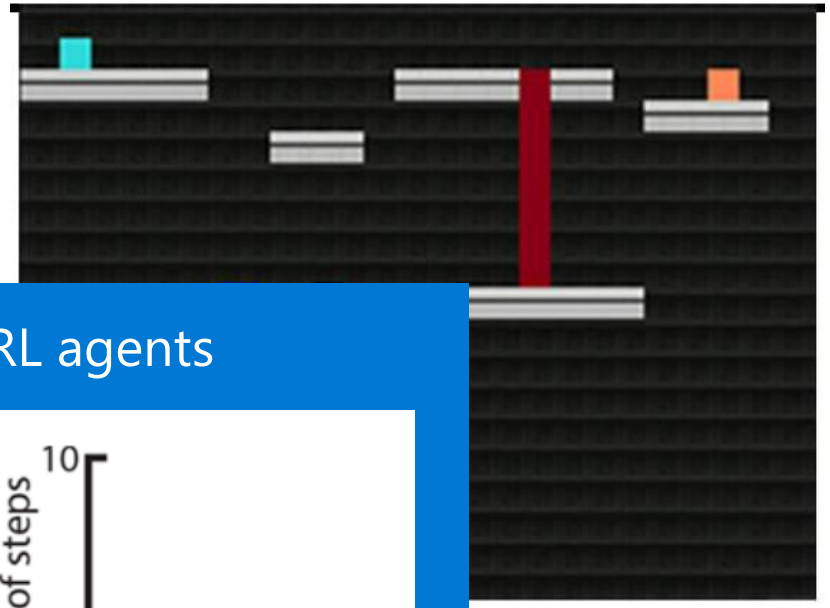
(b) Masked Semantics

Case Study:
Investigating Human Priors for Playing Video Games

Dubey, R., Agrawal, P., Pathak, D., Griffiths, T. L., & Efros, A. A, ICML 2018.

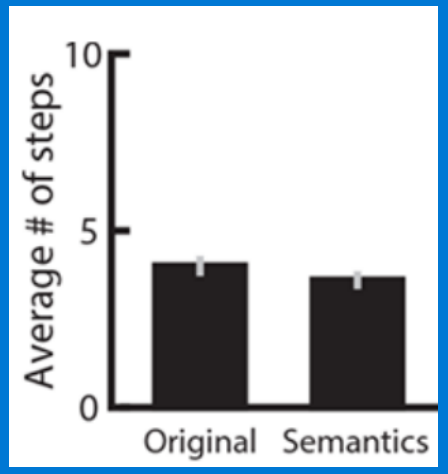


(a) Ori



Human players

RL agents



Case Study:
Investigating Human Priors for Playing Video Games

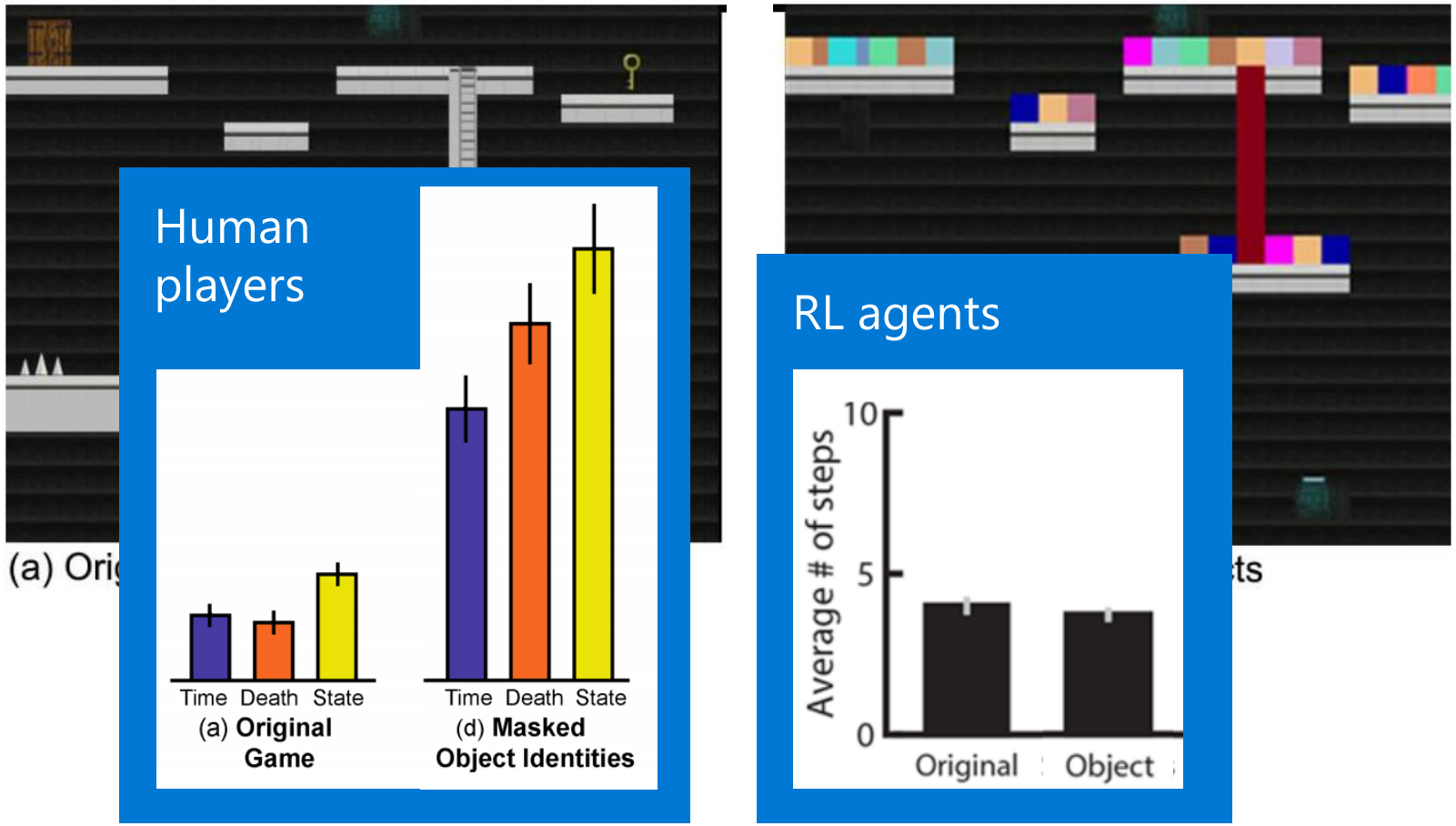
Dubey, R., Agrawal, P., Pathak, D., Griffiths, T. L., & Efros, A. A, ICML 2018.

(a) Original Game

(d) Masked identity of objects

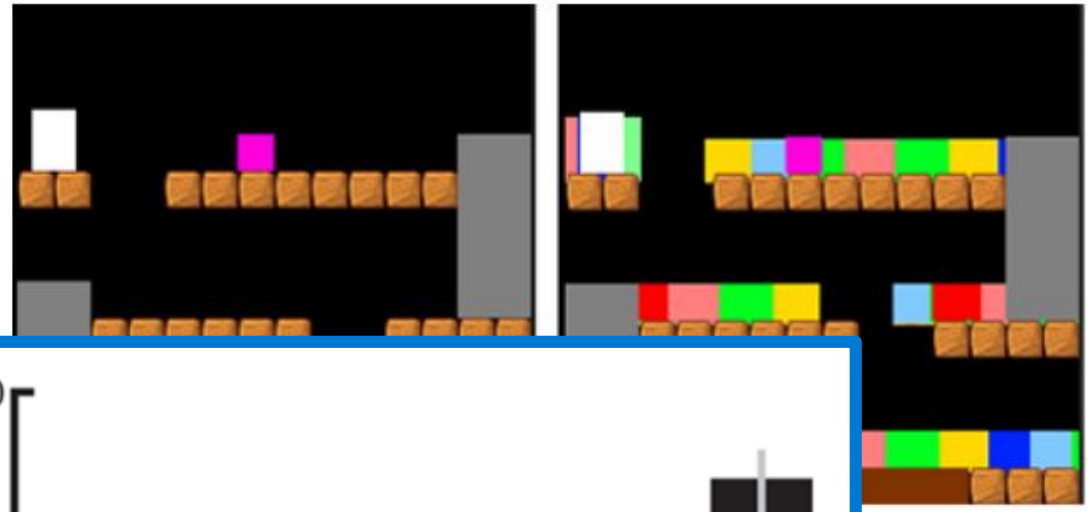
Case Study:
Investigating Human Priors for Playing Video Games

Dubey, R., Agrawal, P., Pathak, D., Griffiths, T. L., & Efros, A. A, ICML 2018.

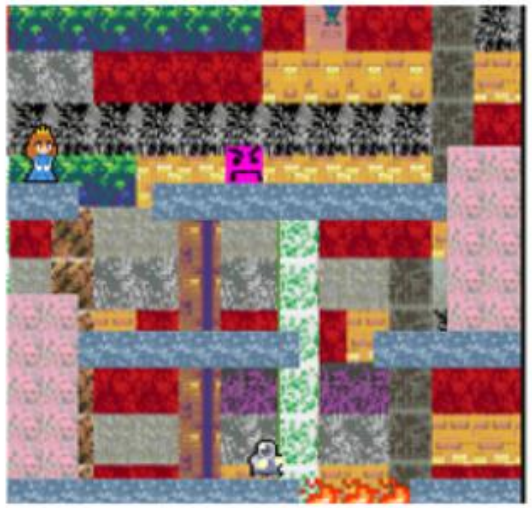


Case Study:
Investigating Human Priors for Playing Video Games

Dubey, R., Agrawal, P., Pathak, D., Griffiths, T. L., & Efros, A. A, ICML 2018.

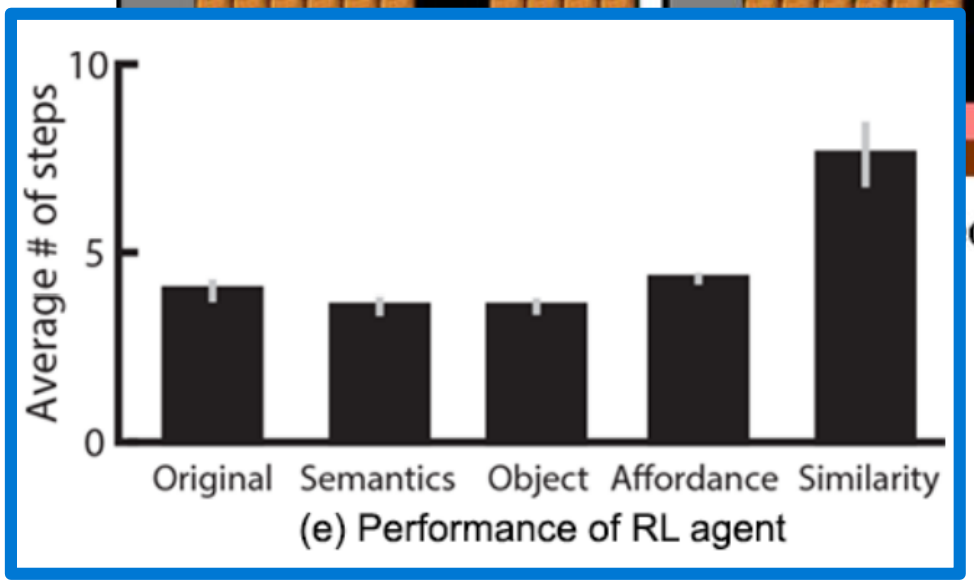


ct identities



(c) Masked affordances

(d) Masked similarity



(e) Performance of RL agent

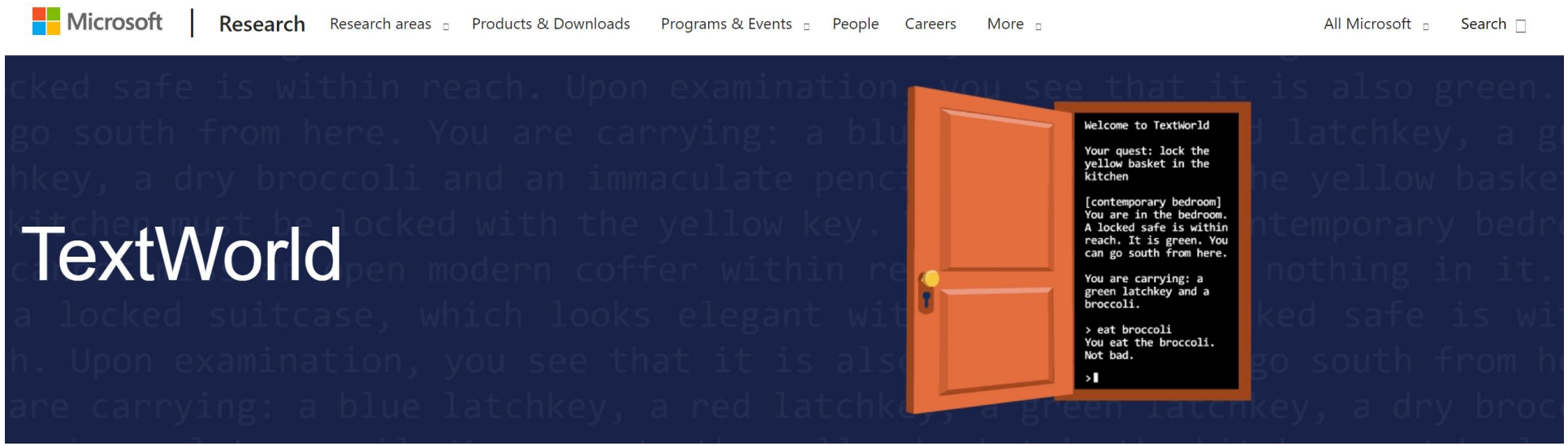
Action space

Important modelling choice, common variants:

a) Discrete, e.g., on/off, which button to press (Atari)
b) Continuous, e.g., how much force to apply, how quickly to accelerate
c) Active research area: large, complex action spaces (e.g., combinatorial, mixed discrete/continuous, natural language)

Trade-offs include: data efficiency, generalization

A Platform for Research: TextWorld



Overview People Publications Contribute

```
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.
```

https://www.microsoft.com/en-us/research/project/textworld/

Rewards

Key Question: where do RL agents' goals come from?

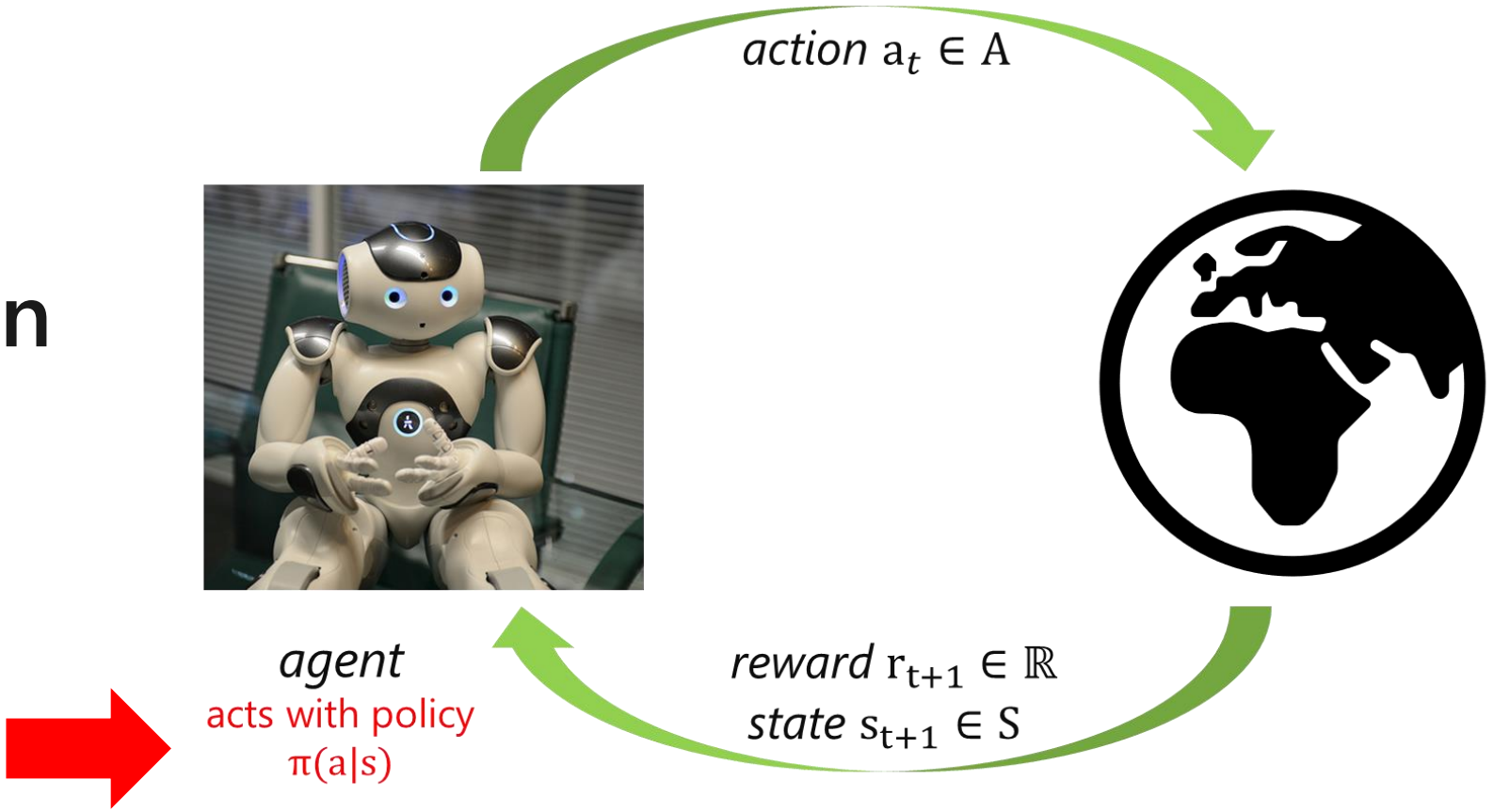
In some settings – natural reward signal may be available (e.g., game score in Atari)

More typically – important modelling choice with strong effects on learned solutions

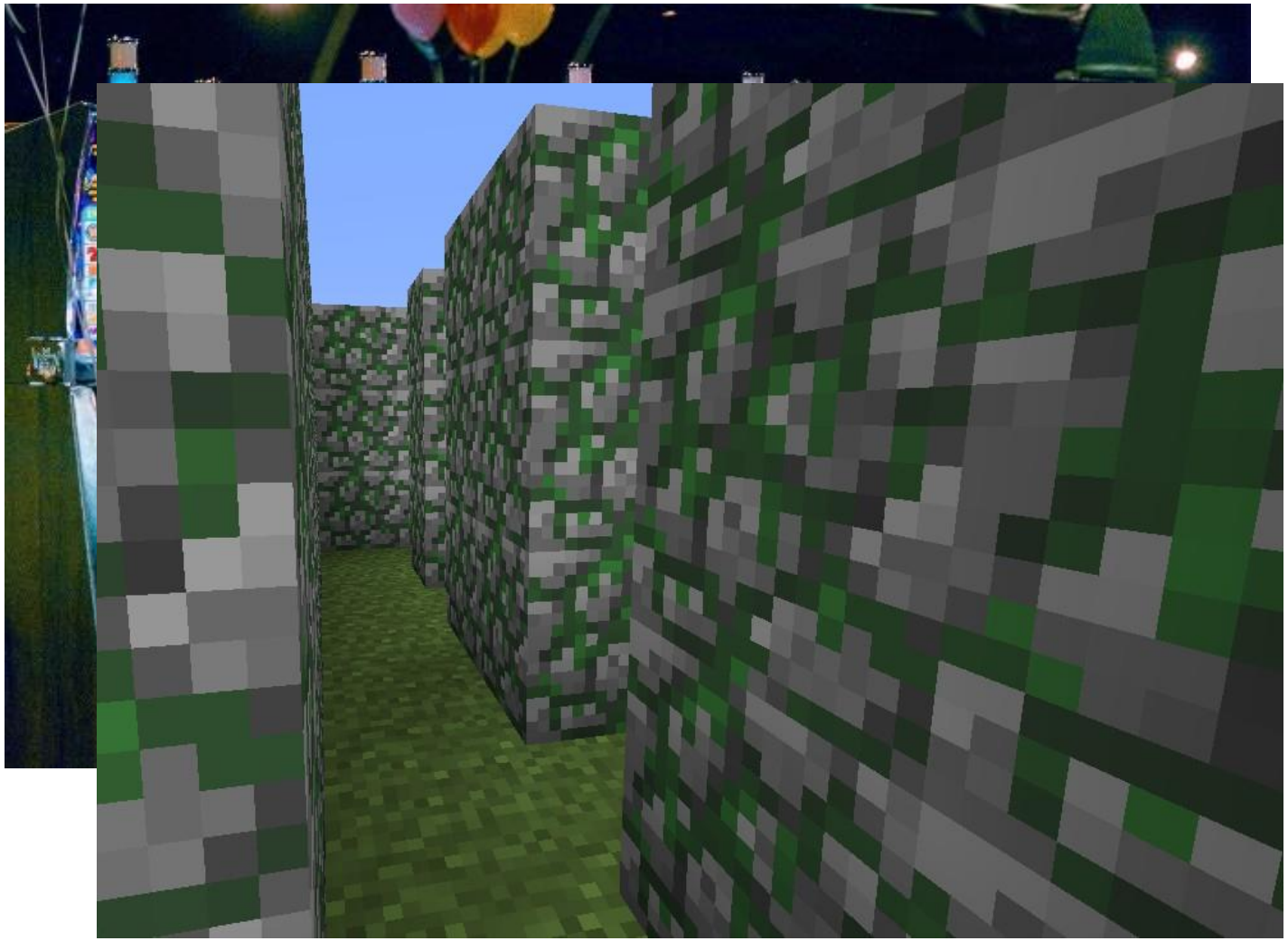
Rewards

For details and full video: https://blog.openai.com/faulty-reward-functions/

RL Key Challenge: Balance Exploration and Exploitation



Exploration vs Exploitation - Example



You have a choice of three slot machines to play, and suspect they have different payoff – how to decide which to choose?

Would this change if you were facing several hallways to choose from?

Performance is problem dependent – here: overview of common choices

Exploration vs Exploitation – ε-greedy

ε-greedy:

$$\pi_t = \begin{cases} \underset{a \in A}{\operatorname{argmax}}\, \hat{r}_t(a) & w.prob.\, 1-\varepsilon \\ rand(a) & w.prob.\, \varepsilon \end{cases}$$

- ✓ Simple to implement
- ✓ Theoretical guarantees
- ✓ Often competitive in practice

Exploration vs Exploitation – Softmax action selection

Softmax policy:

$$\pi(a|s) = \frac{e^{h(s,a)}}{\sum_{a' \in A} e^{h(s,a')}}$$

✓ Strong connection with policy gradient learners (more in a bit)

Exploration vs Exploitation – Optimism

Common principle – optimism!

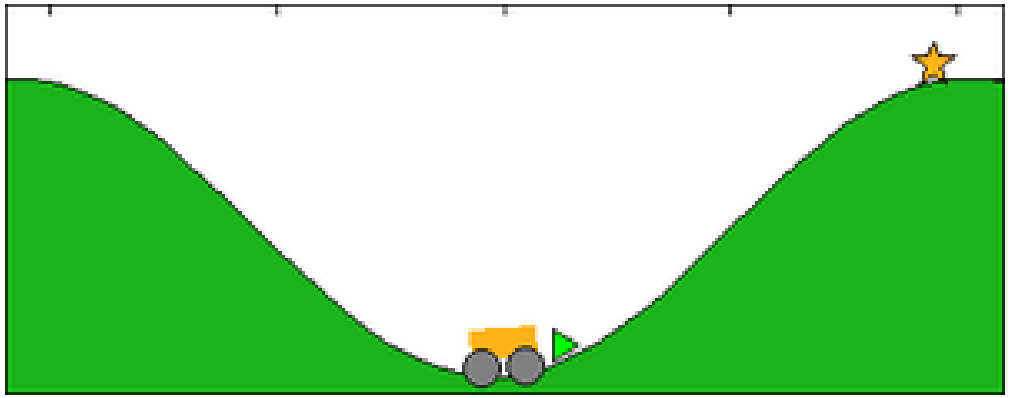
Simplest approach: optimistic initialization, combine with greedy or ε-greedy policy

- ✓ Simple to implement
- ❑ Requires knowledge of reward structure

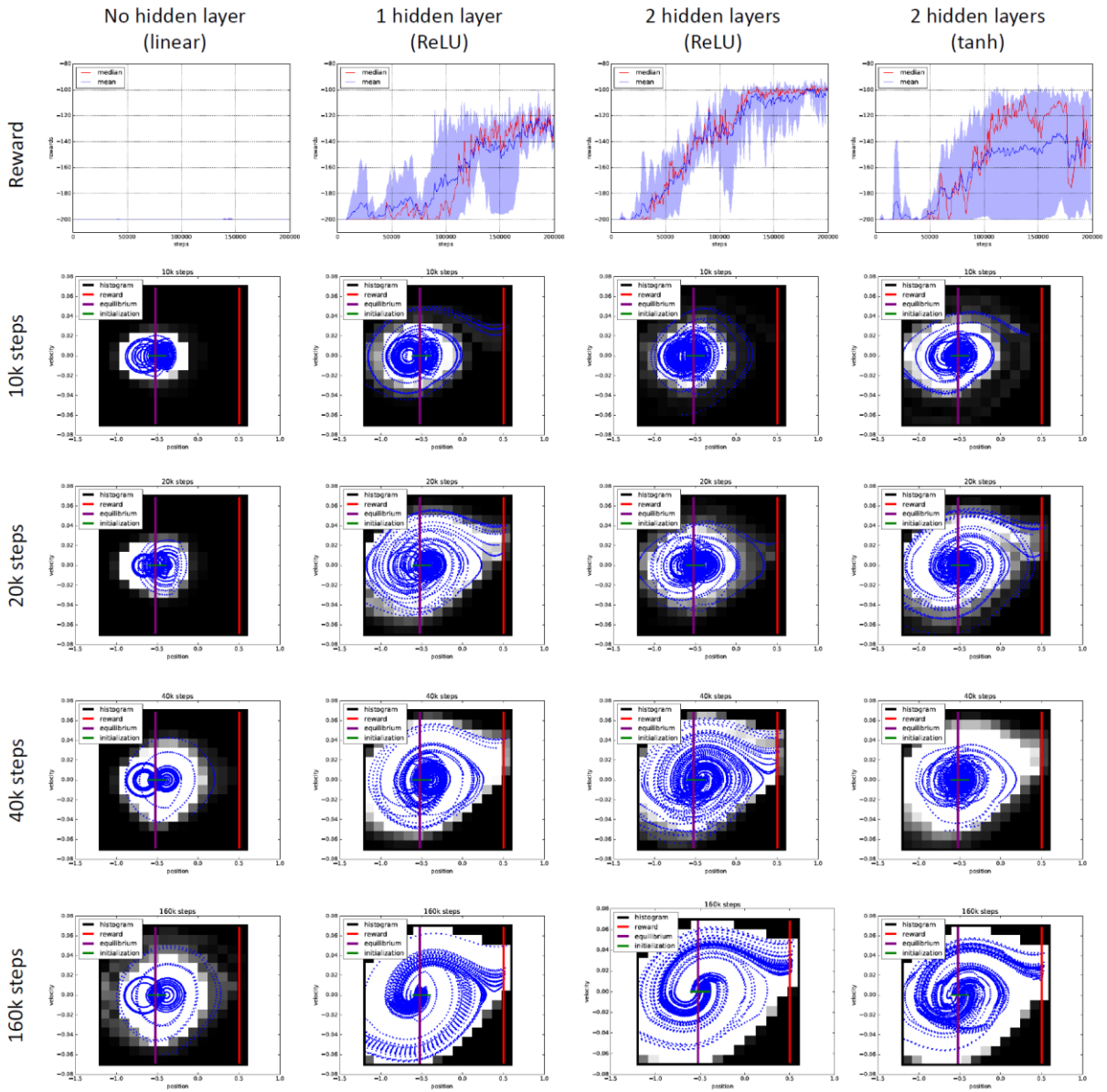
Case Study: **Depth and nonlinearity induce implicit exploration for RL**

J. Dauparas, R. Tomioka, K. Hofmann,
ERL workshop at ICML 2018

Interesting interaction with deep learning / optimization



https://en.wikipedia.org/wiki/Mountain_car_problem



Exploration vs Exploitation – Optimism in the Face of Uncertainty (OFU)

Upper Confidence Bound (UCB):

$$\pi_t = \underset{a \in A}{\operatorname{argmax}}\, \hat{r}_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}}$$

[Auer et al. '02]

- ✓ Strong guarantees (log)
- ❑ Key challenge: flexible yet meaningful uncertainty estimates

Posterior Sampling

Maintain distribution $P(r|a)$. At time t sample from this distribution, and take the optimal action according to the sample; update P.

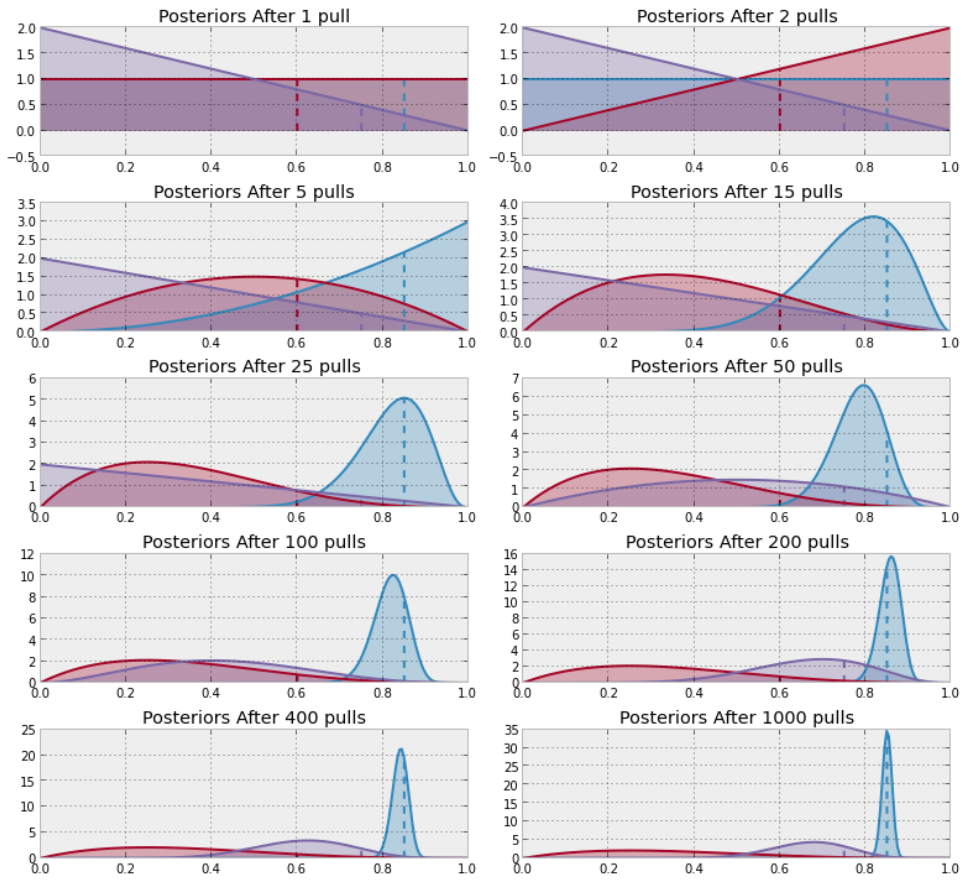
[Thompson '33, Chapelle & Li '11, Russo & Van Roy '14]

Example: Thompson Sampling

Balances exploration-exploitation using a simple principle:

1. maintain reward distribution (e.g., per action)
2. when taking an action, sample from that distribution and act optimally according to the sample
3. use observed reward to update reward distribution

Example: 3-armed bandit



https://dataorigami.net/blogs/napkin-folding/79031811-multi-armed-bandits – Credit: Cameron Davidson-Pilon

Case Study:
Deep Exploration via Bootstrapped DQN

Ian Osband , Charles Blundell, Alexander Pritzel, Benjamin Van Roy. ICML 2018.

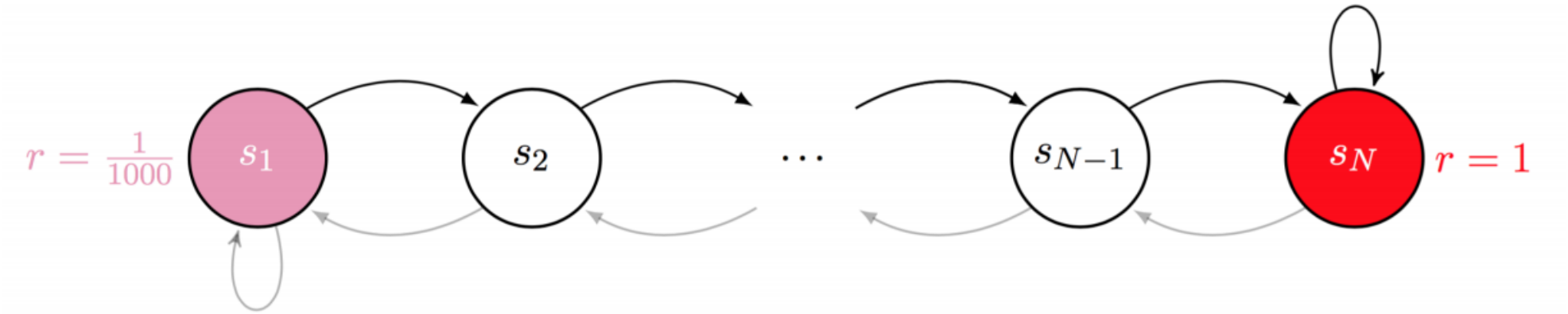
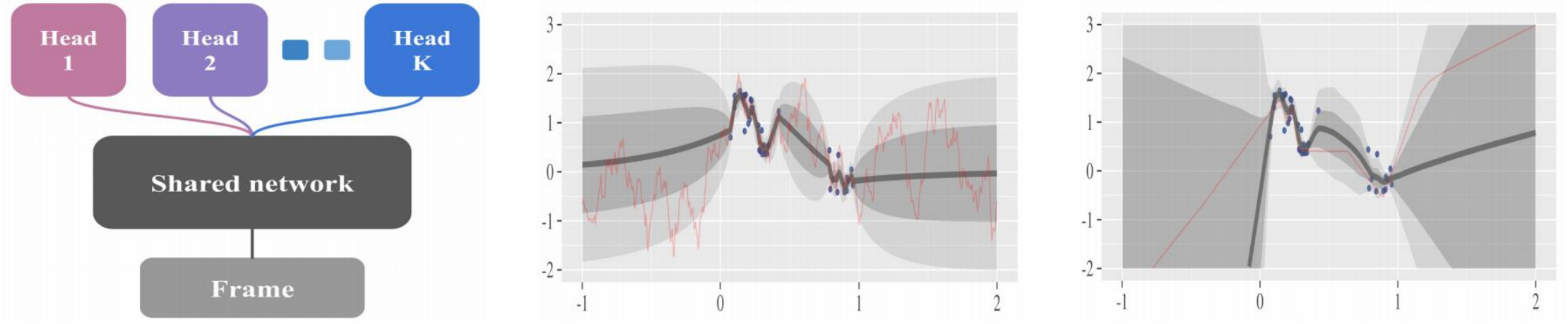


Figure 3: Scalable environments that requires deep exploration.

Case Study:
Deep Exploration via Bootstrapped DQN

Ian Osband , Charles Blundell, Alexander Pritzel, Benjamin Van Roy. ICML 2018.



(a) Shared network architecture (b) Gaussian process posterior (c) Bootstrapped neural nets

Figure 1: Bootstrapped neural nets can produce reasonable posterior estimates for regression.

Case Study:

Deep Exploration via Bootstrapped DQN

Ian Osband , Charles Blundell, Alexander Pritzel, Benjamin Van Roy. ICML 2018.

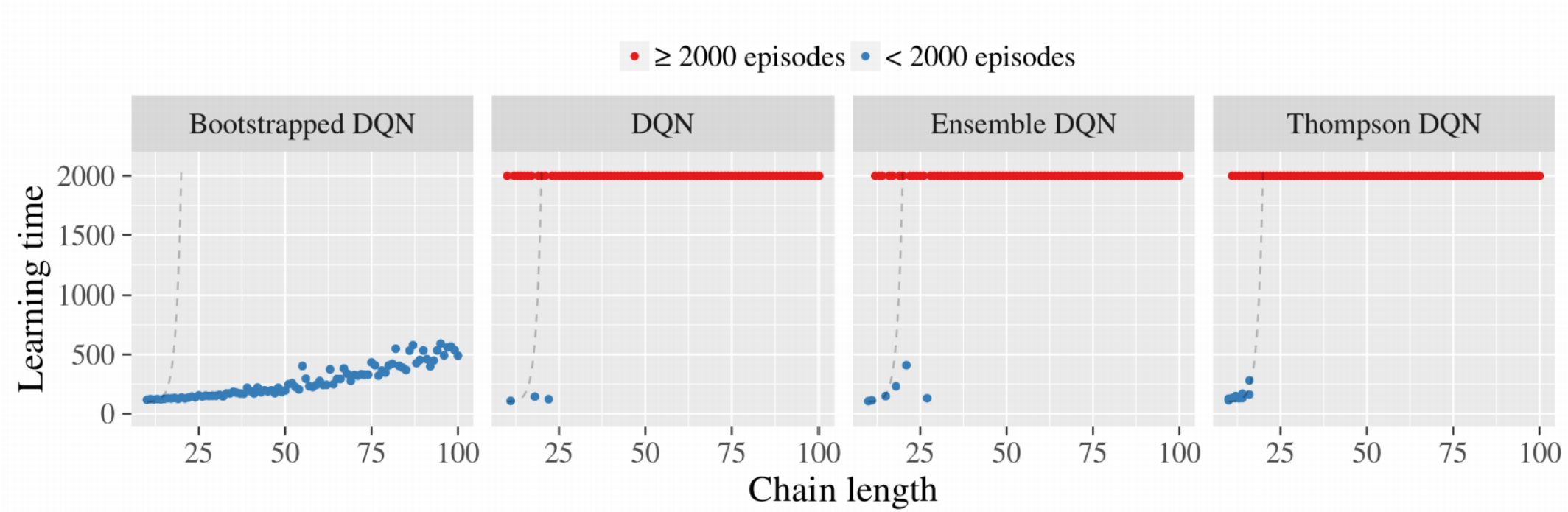


Figure 4: Only Bootstrapped DQN demonstrates deep exploration.

Further Reading

[Auer et al. ’02] P. Auer, N. Cesa-Bianchi, P. Fischer: *Finite-time analysis of the Multiarmed Bandit Problem*. Machine Learning 47, 2002a.

[Chapelle & Li ‘11] O. Chapelle & L. Li: *An Empirical Evaluation of Thompson Sampling*. NIPS, 2011.

[Mnih et al. ‘15] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540).

[Russo & Van Roy ‘14] D. Russo & B. Van Roy: *An Information-Theoretic Analysis of Thompson Sampling*. JMLR pre-print (to appear).

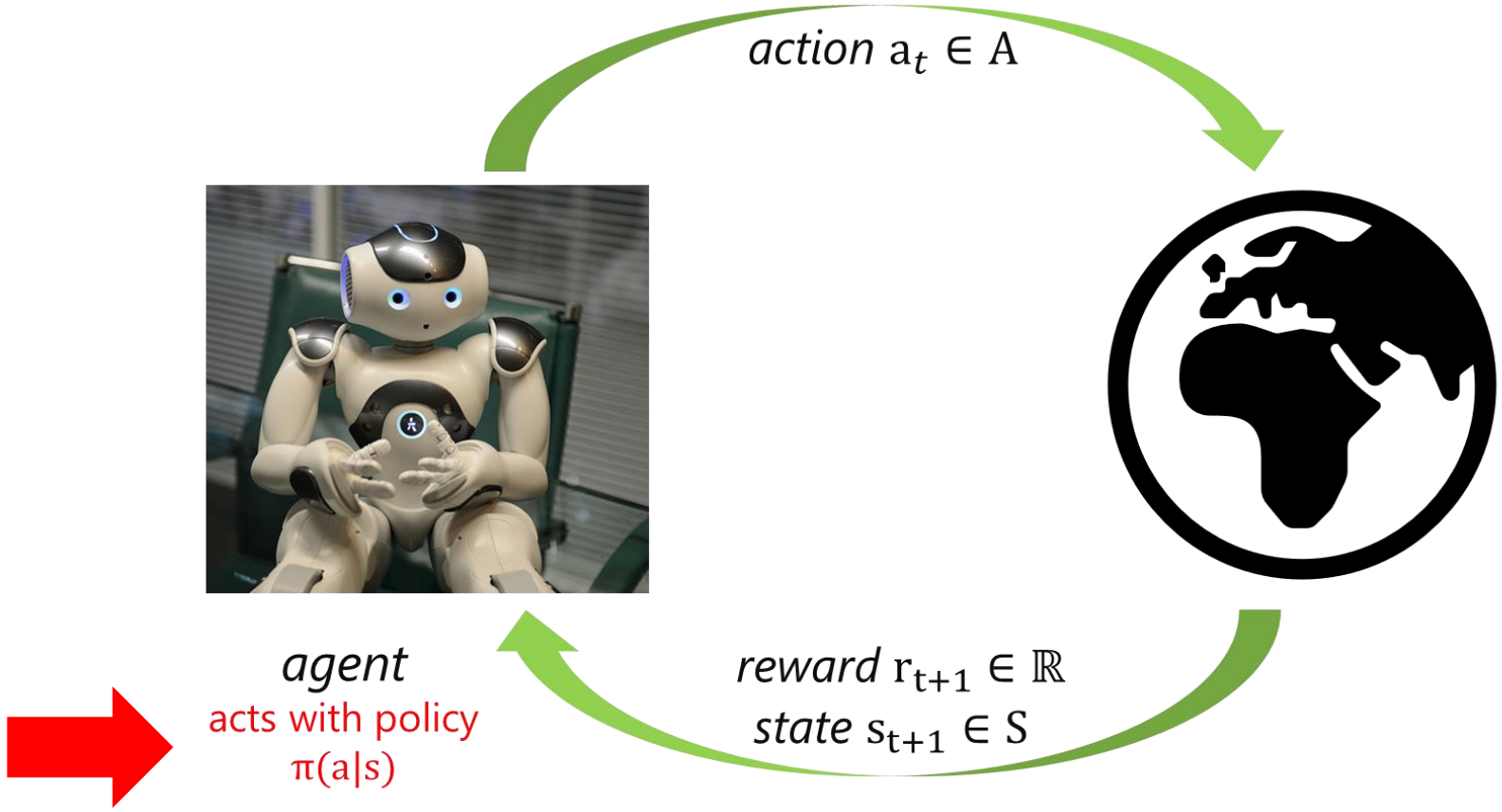
[Sutton & Barto ‘98] R.S. Sutton & A.G. Barto: *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[Thompson ‘33] W. R. Thompson: *On the likelihood that one unknown probability exceeds another in view of the evidence of two samples*. Biometrika, 25(3–4):285–294, 1933.

RL Approaches 1:
Policy Gradient Methods

Policy Gradient: Intuition

Focus on learning a good behaviour policy



Policy Gradient: Intuition

Example: Learning in Multi-armed bandit problems

Photo credit: https://www.flickr.com/photos/knothing/11264853546/

Policy Gradient: Intuition

Example: Learning in Multi-armed bandit problems

π	.5	.5

Photo credit: https://www.flickr.com/photos/knothing/11264853546/

Policy Gradient: Intuition

Example: Learning in Multi-armed bandit problems

π	.5	.5

Photo credit: https://www.flickr.com/photos/knothing/11264853546/

Policy Gradient: Intuition

Example: Learning in Multi-armed bandit problems

π	.5	.5

Photo credit: https://www.flickr.com/photos/knothing/11264853546/

Policy Gradient: Intuition

Example: Learning in Multi-armed bandit problems

π	.45	.55

Photo credit: https://www.flickr.com/photos/knothing/11264853546/

Policy Gradient: Intuition

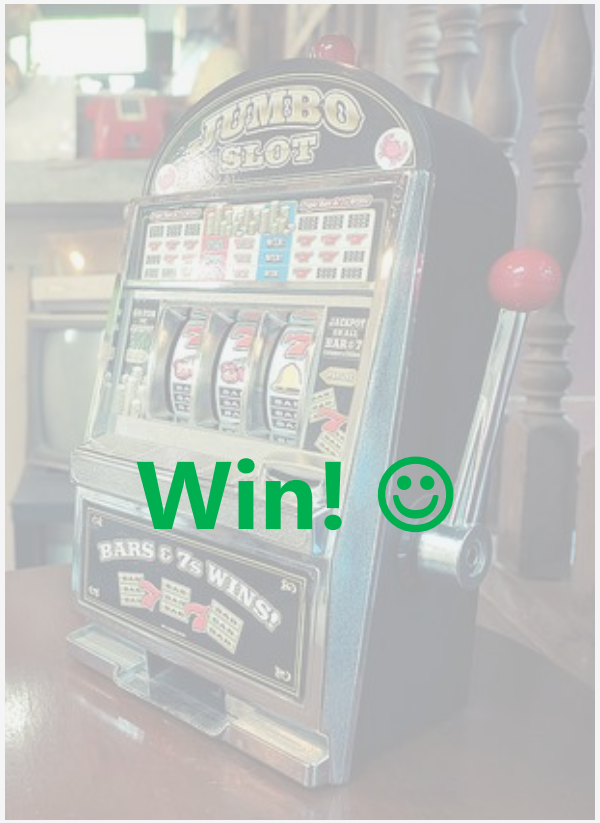
Example: Learning in Multi-armed bandit problems

π	.45	.55

Photo credit: https://www.flickr.com/photos/knothing/11264853546/

Policy Gradient: Intuition

Example: Learning in Multi-armed bandit problems



π	.45	.55

Photo credit: https://www.flickr.com/photos/knothing/11264853546/

Policy Gradient: Intuition

Example: Learning in Multi-armed bandit problems

π	.4	.6

Photo credit: https://www.flickr.com/photos/knothing/11264853546/

Policy Gradient: Intuition

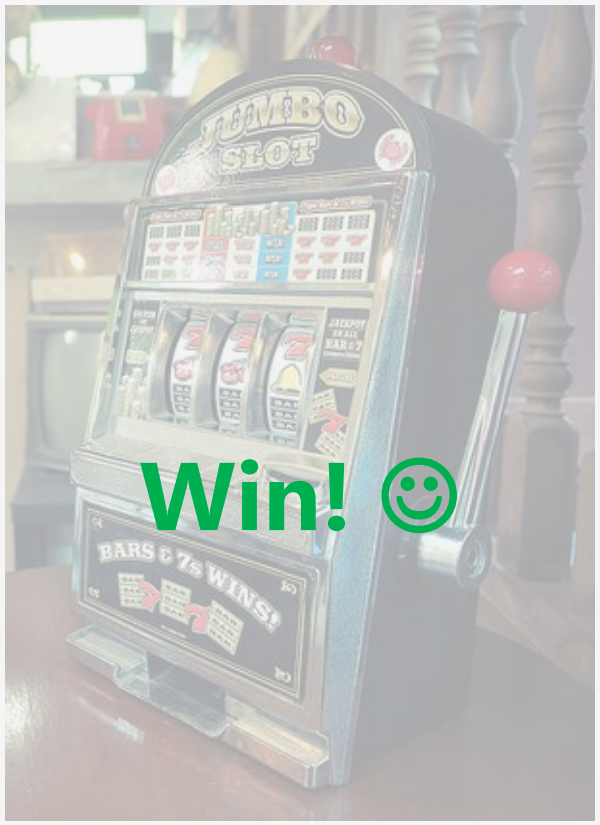
Example: Learning in Multi-armed bandit problems

π	.4	.6

Photo credit: https://www.flickr.com/photos/knothing/11264853546/

Policy Gradient: Intuition

Example: Learning in Multi-armed bandit problems



π	.4	.6

Photo credit: https://www.flickr.com/photos/knothing/11264853546/

Policy Gradient: Intuition

Example: Learning in Multi-armed bandit problems

π	.45	.55

Photo credit: https://www.flickr.com/photos/knothing/11264853546/

Policy Gradient: Intuition

Focus on learning a good behaviour policy

Repeat:

1. Collect experience using the current policy
2. Update the policy towards better outcomes

Focus on the Policy: Parametric Form

Most common parameterization:

$$\pi(a|s;\theta) = \frac{e^{h(s,a;\theta)}}{\sum_{a' \in A} e^{h(s,a';\theta)}}$$

Focus on the Policy: Parametric Form

Most common parameterization:

$$\pi(a|s;\theta) = \frac{e^{h(s,a;\theta)}}{\sum_{a' \in A} e^{h(s,a';\theta)}}$$

Policy: **probability distribution** over actions, given the current state

Focus on the Policy: Parametric Form

Most common parameterization:

$$\pi(a|s;\theta) = \frac{e^{h(s,a;\theta)}}{\sum_{a' \in A} e^{h(s,a';\theta)}}$$

The policy has **learnable parameters**

Focus on the Policy: Parametric Form

Most common parameterization:

$$\pi(a|s;\theta) = \frac{e^{h(s,a;\theta)}}{\sum_{a' \in A} e^{h(s,a';\theta)}}$$

The policy parameters encode **action preferences** – more preferred actions are more likely

Focus on the Policy: Parametric Form

Most common parameterization:

$$\pi(a|s;\theta) = \frac{e^{h(s,a;\theta)}}{\sum_{a' \in A} e^{h(s,a';\theta)}}$$

The denominator ensures **normalized probabilities**

Policy Gradient Objective

Goal: find parameters θ that maximize expected reward

$$J(\theta) = \sum_{s \in S} p^{\pi}(s) \sum_{a \in A} \pi(a|s;\theta) R_s^a$$

Policy Gradient Objective

Goal: find parameters θ that maximize expected reward

$$J(\theta) = \sum_{s \in S} p^{\pi}(s) \sum_{a \in A} \pi(a|s;\theta) R_s^a$$

Stationary state distribution under π_θ

Policy Gradient Objective

Goal: find parameters θ that maximize expected reward

$$J(\theta) = \sum_{s \in S} p^{\pi}(s) \sum_{a \in A} \pi(a|s;\theta) R_s^a$$

Stationary state distribution under π_θ

The parameterized policy

Policy Gradient Objective

Goal: find parameters θ that maximize expected reward

$$J(\theta) = \sum_{s \in S} p^{\pi}(s) \sum_{a \in A} \pi(a|s;\theta) R_s^a$$

Stationary state distribution under π_θ

The parameterized policy

Expected return when starting from state s and taking action a

Policy Gradient Objective

Goal: find parameters θ that maximize expected reward

$$J(\theta) = \sum_{s \in S} p^{\pi}(s) \sum_{a \in A} \pi(a|s;\theta) R_s^a$$

$$\nabla J(\theta) \propto \sum_{s \in S} p^{\pi}(s) \sum_{a \in A} \nabla \log \pi(a|s;\theta)\, R_s^a$$

Challenge: compute update to parameterized policy – which depends on the unknown environment dynamics

The Policy Gradient Theorem

Key insight: gradient of J does not require derivatives of $p^{\pi}(s)$

$$\nabla J(\theta) \propto \sum_{s \in S} p^{\pi}(s) \sum_{a \in A} \nabla \log \pi(a|s;\theta)\, R_s^a$$

$$= \mathbb{E}_{\pi}[\nabla \log \pi(a|s;\theta)\, R_s^a]$$

Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. *NIPS 2000*.
See Sutton & Barto 2018, chapter 13

The Policy Gradient Theorem

Terms can be estimated from data!

$$\mathbb{E}_{\pi}[\nabla \log \pi(a|s;\theta)\, R_s^a]$$

Rollout the current policy π

Implement forward pass in your favourite deep learning framework + auto-diff to compute gradients

Episode returns under π

Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. *NIPS 2000*.
See Sutton & Barto 2018, chapter 13

Policy Gradient Algorithm: REINFORCE

REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ (G_t)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, *8*(3-4), 229-256.
Algorithm from: Sutton & Barto 2018, chapter 13, page 328

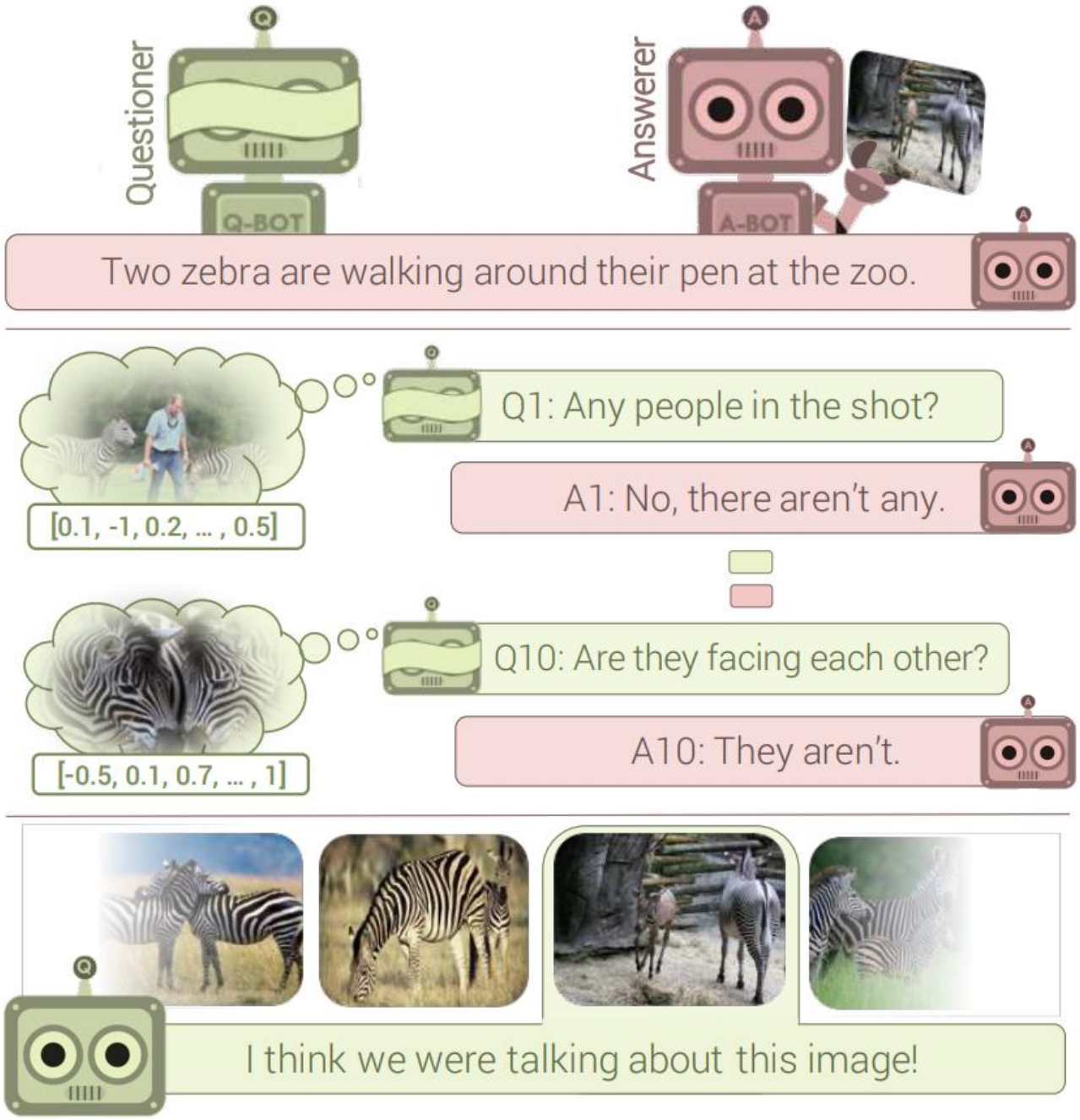
Case Study: Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning

Das, A., Kottur, S., Moura, J. M., Lee, S., & Batra, D. (2017). *ICCV, 2017*.

https://visualdialog.org/

Key insight: REINFORCE can improve dialog policy on top of supervised (imitation) learning

$$r_t\Big(\underbrace{s_t^Q}_{\text{state}}, \underbrace{(q_t, a_t, y_t)}_{\text{action}}\Big) = \underbrace{\ell\left(\hat{y}_{t-1}, y^{gt}\right)}_{\text{distance at } t\text{-}1} - \underbrace{\ell\left(\hat{y}_t, y^{gt}\right)}_{\text{distance at } t}$$



Case Study: Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning

Das, A., Kottur, S., Moura, J. M., Lee, S., & Batra, D. (2017). *ICCV, 2017.*

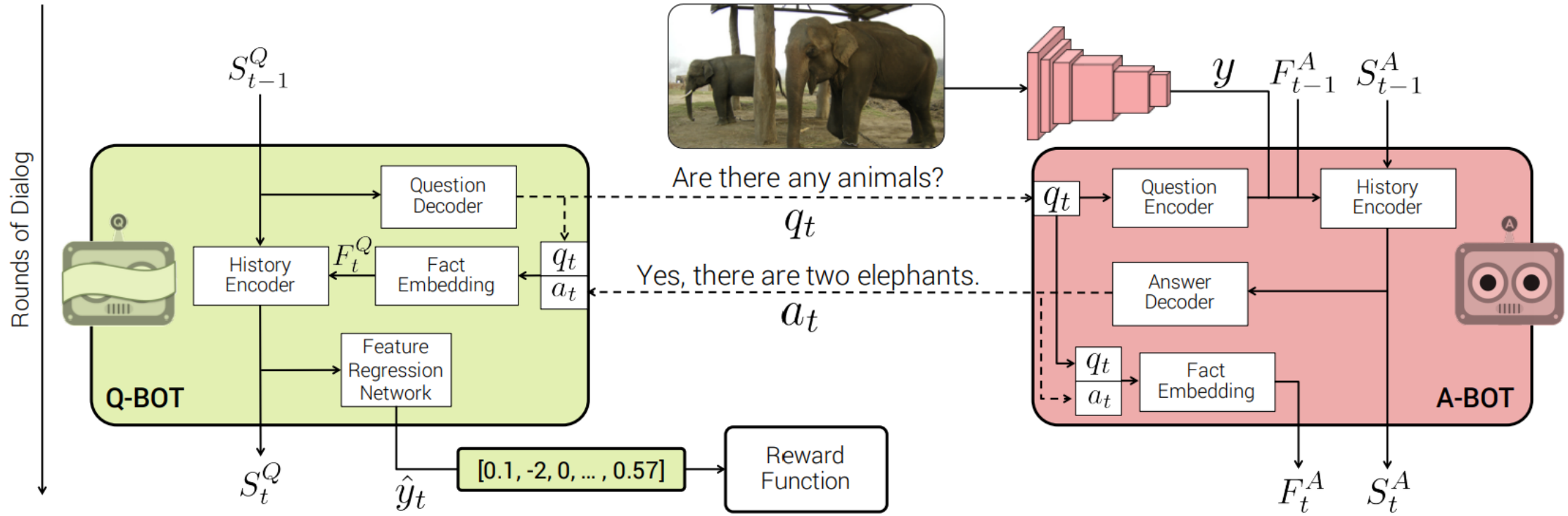


Figure 2: Policy networks for Q-BOT and A-BOT. At each round t of dialog, (1) Q-BOT generates a question q_t from its question decoder conditioned on its state encoding S^Q_{t-1}, (2) A-BOT encodes q_t, updates its state encoding S^A_t, and generates an answer a_t, (3) both encode the completed exchange as F^Q_t and F^A_t, and (4) Q-BOT updates its state to S^Q_t, predicts an image representation $\hat{y}_t$, and receives a reward.

Further Reading

Research focus: reduce variance, local vs global optima, exploration.

Examples:

Lilian Weng's Blog: https://lilianweng.github.io/lil-log/2018/04/08/policy-gradient-algorithms.html

[Haarnoja et al. '18] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine: *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor*. ICML 2018.

[Schulman et al. '17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov. *Proximal Policy Optimization Algorithms*. Arxiv: 1707.06347

[Sutton & Barto '98] R.S. Sutton & A.G. Barto: *Reinforcement Learning: An Introduction*. MIT Press, 1998.