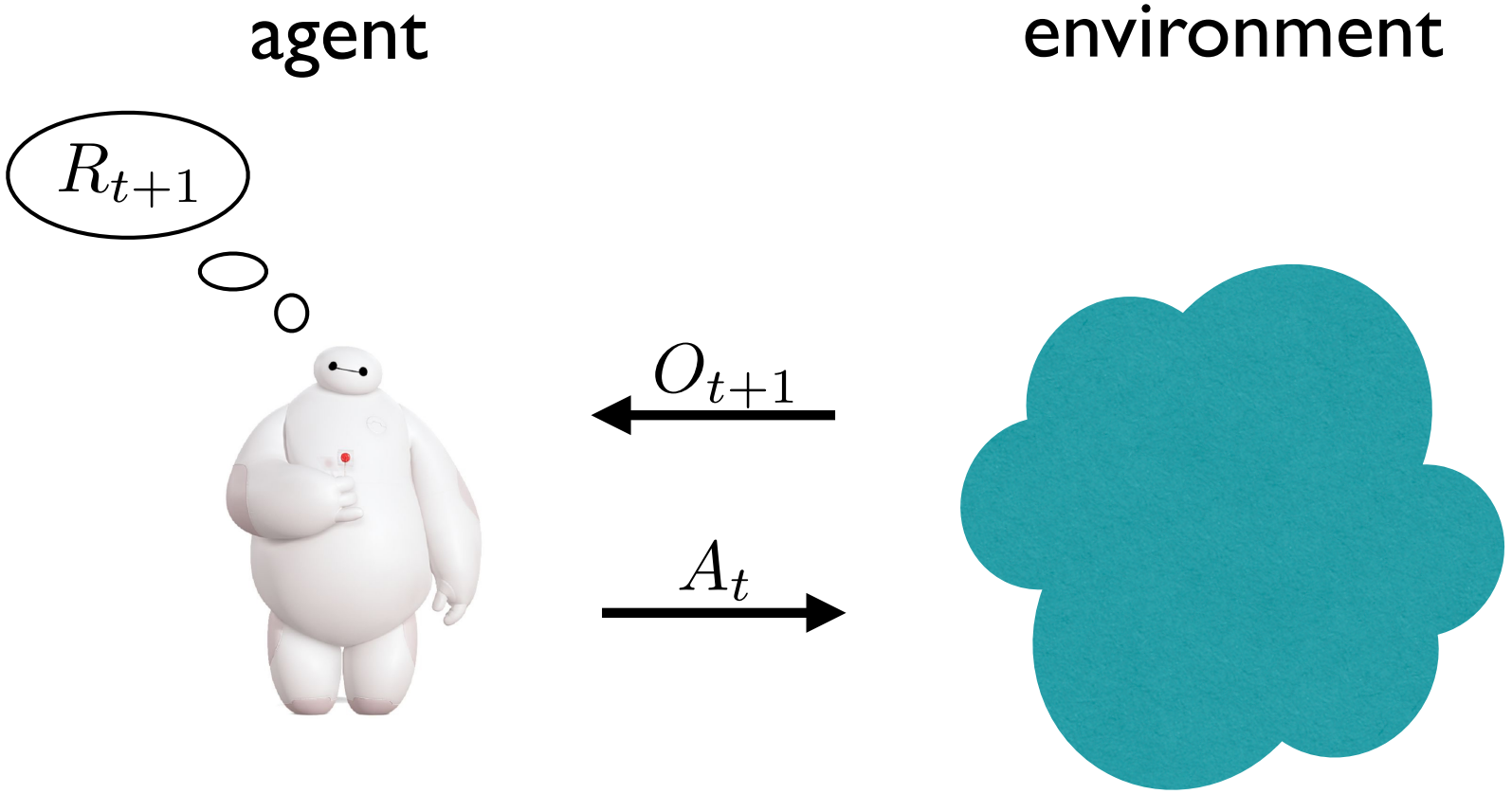
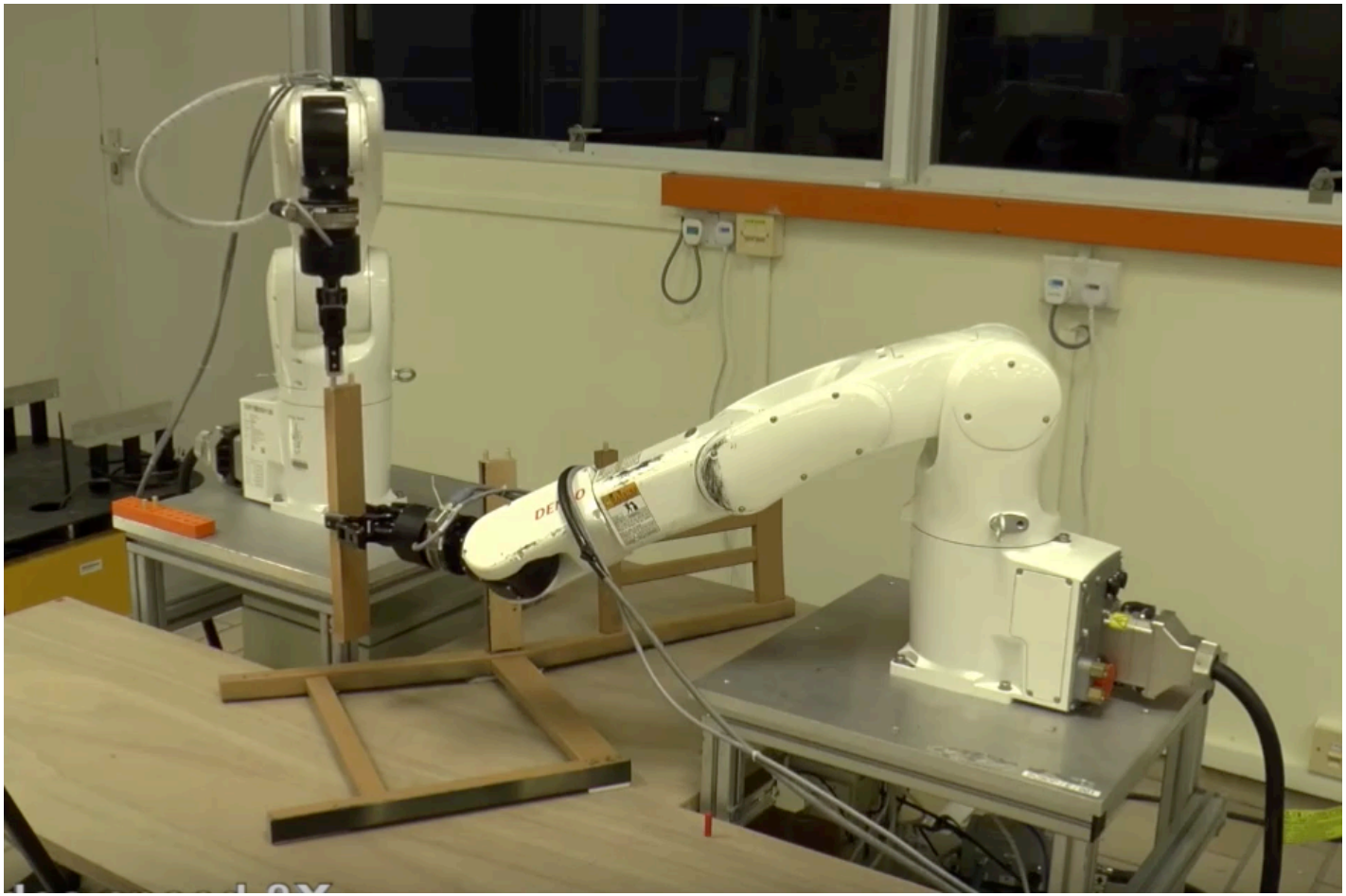
exploration in reinforcement learning

Benjamin Van Roy

Reinforcement Learning

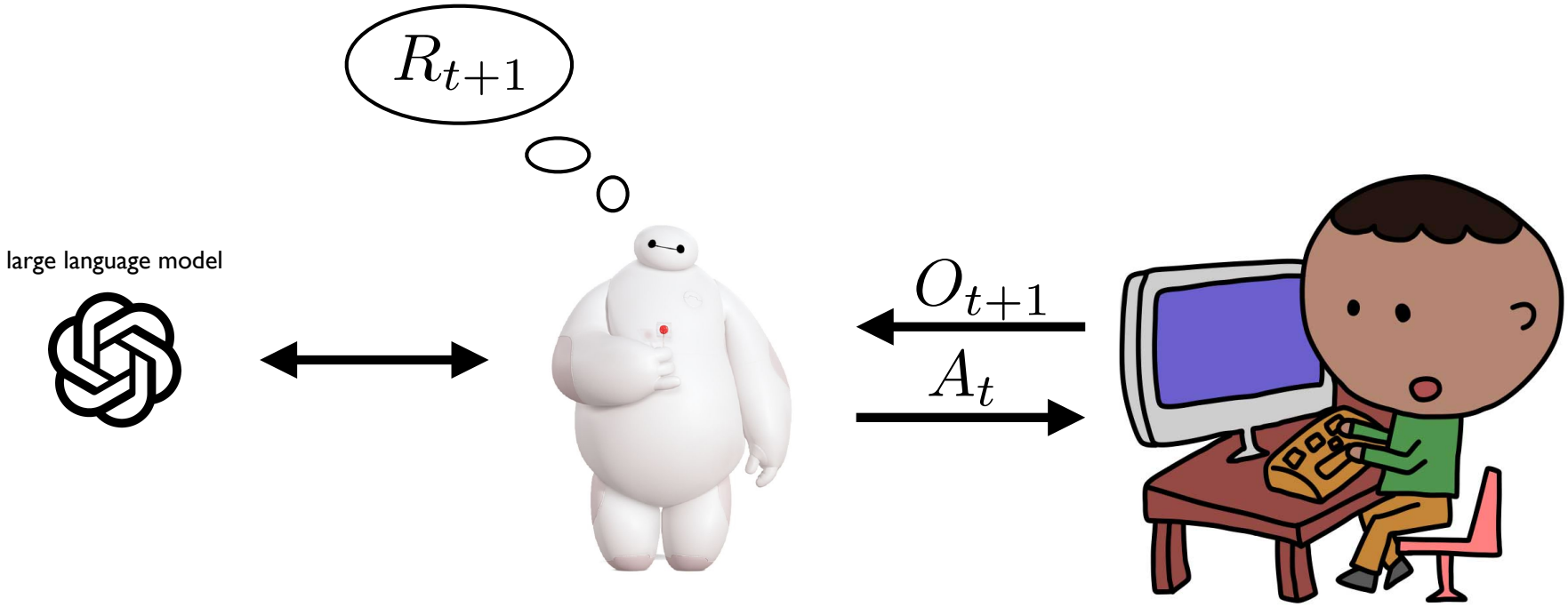


robotics



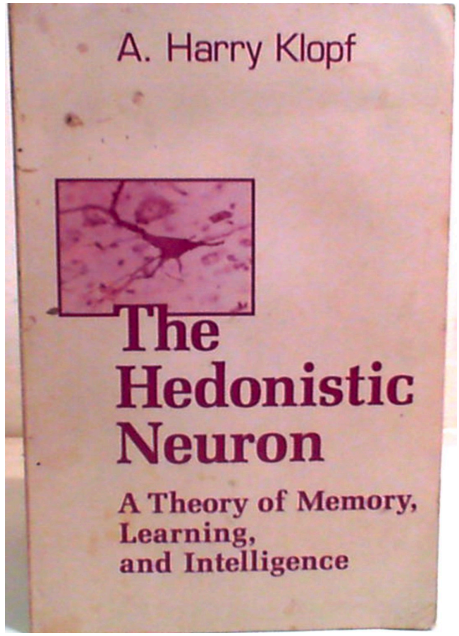
online education

leveraging llms



exploration

other learning paradigms are about **minimization**,
reinforcement learning is about **maximization**

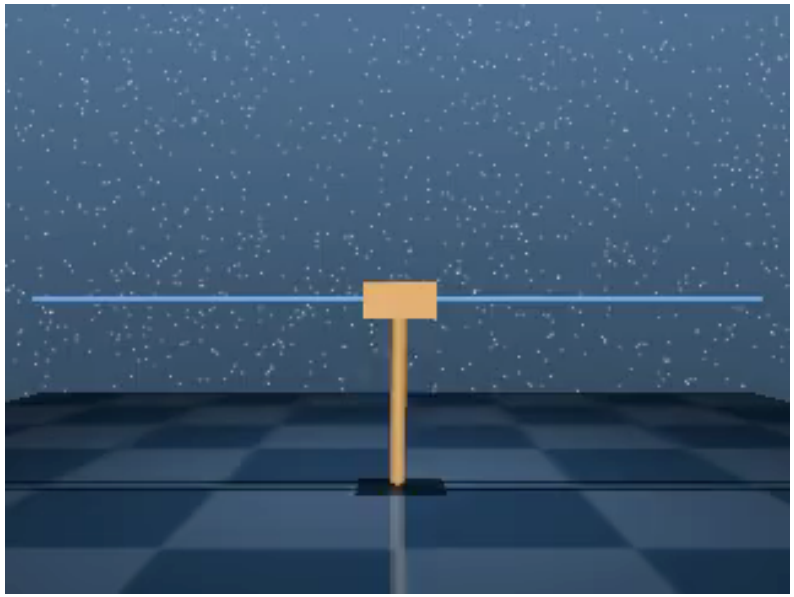


dithering

the prevalent approach to exploration in deep rl

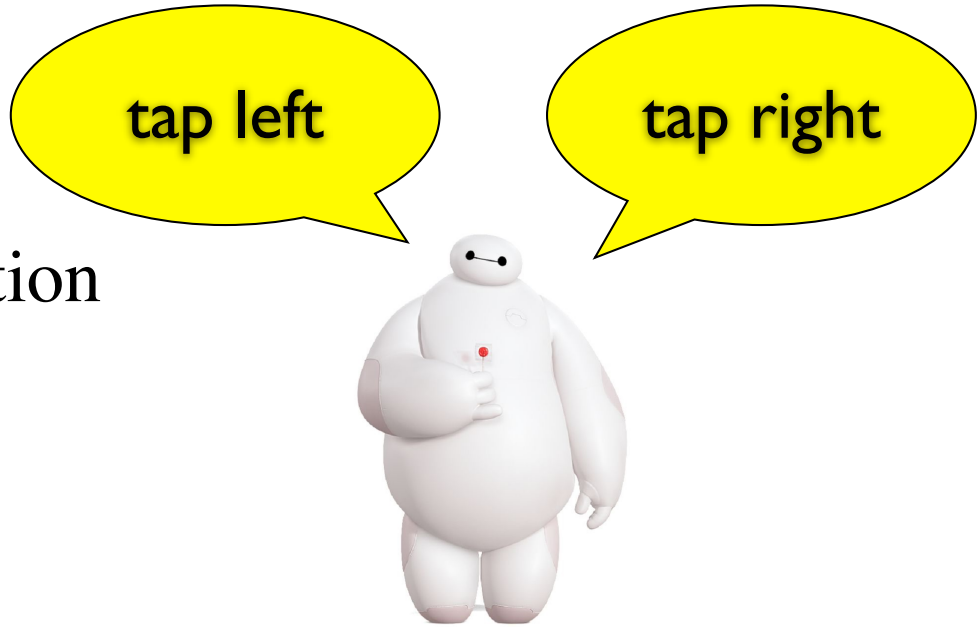
- consider learning a value function $Q_\theta(s, a)$
- greedy $A_t = \arg\max_{a \in \mathcal{A}} Q_\theta(S_t, a)$
- ϵ-greedy $A_t = \begin{cases} \arg\max_{a \in \mathcal{A}} Q_\theta(S_t, a) & \text{w.p. } 1 - \epsilon \\ \text{unif}(\mathcal{A}) & \text{w.p. } \epsilon \end{cases}$
- ϵ-Boltzmann $A_t \sim \frac{e^{Q_\theta(S_t, \cdot)/\epsilon}}{\sum_{a \in \mathcal{A}} e^{Q_\theta(S_t, a)}}$

Cart-Pole Problem



- Easy case
  - Continual feedback
  - Distance from goal state
  - Solved by basic RL algorithms

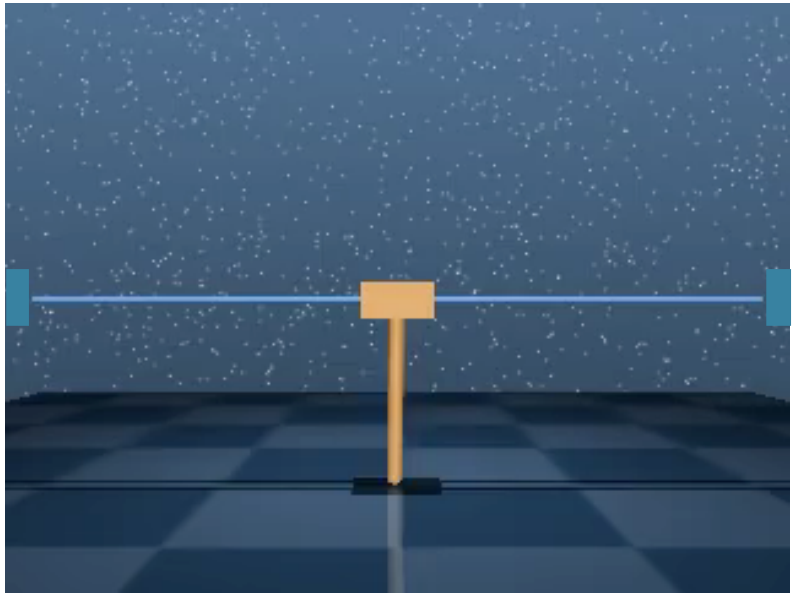
- Hard case
  - Sparse feedback
  - Reward for completion
  - Requires sophisticated exploration



Cart-Pole with Sparse Reward

dithering exploration

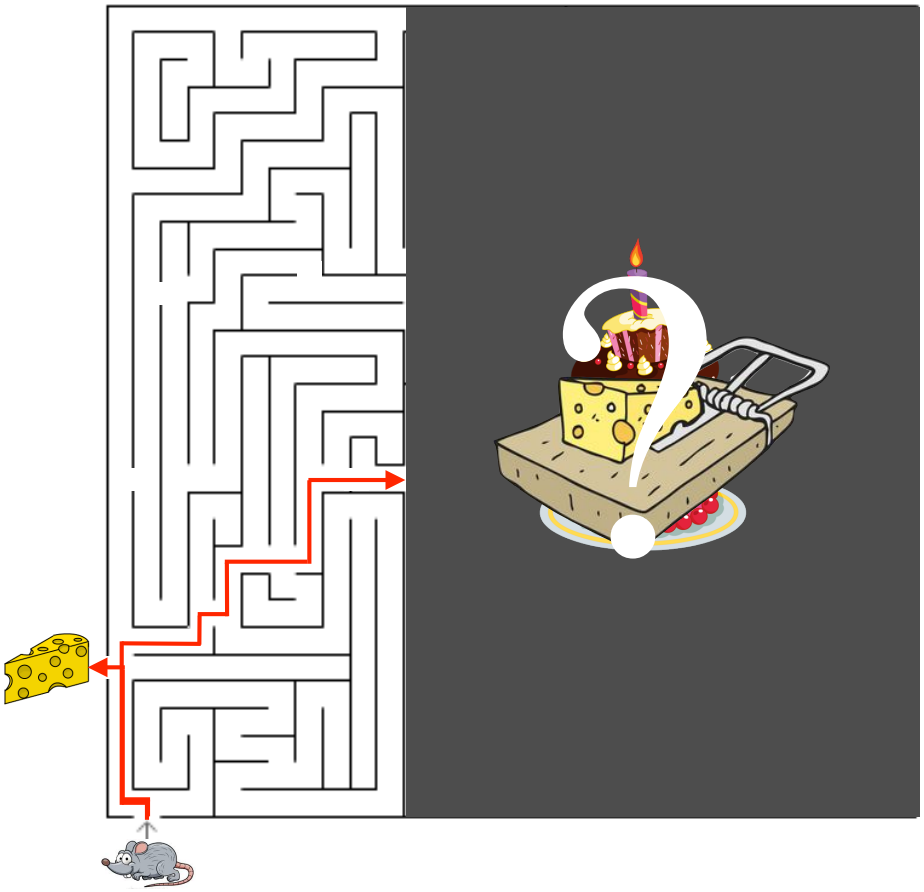
deep exploration



after 1000 episodes

after 1000 episodes

What is Deep Exploration?

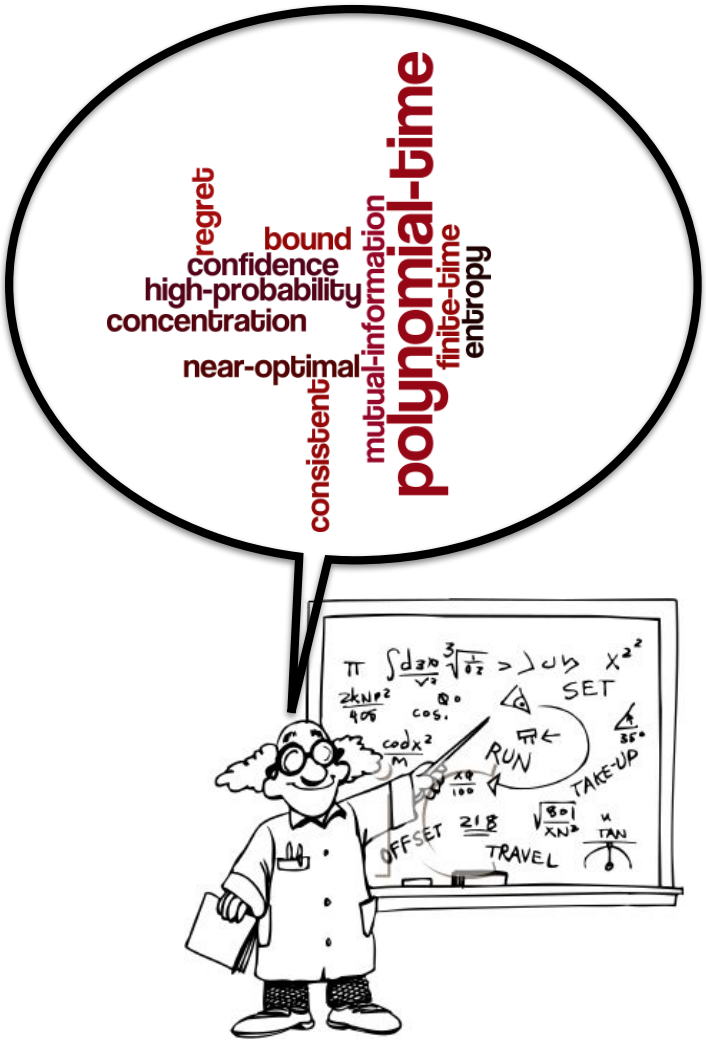


motivations

	reward	learning
immediate	myopic optimization	myopic exploration
later in episode	multi-period optimization	**deep exploration**

how to make this scalable for deep rl?

Two Cultures of Reinforcement Learning?



efficient exploration may rely on insight from theory

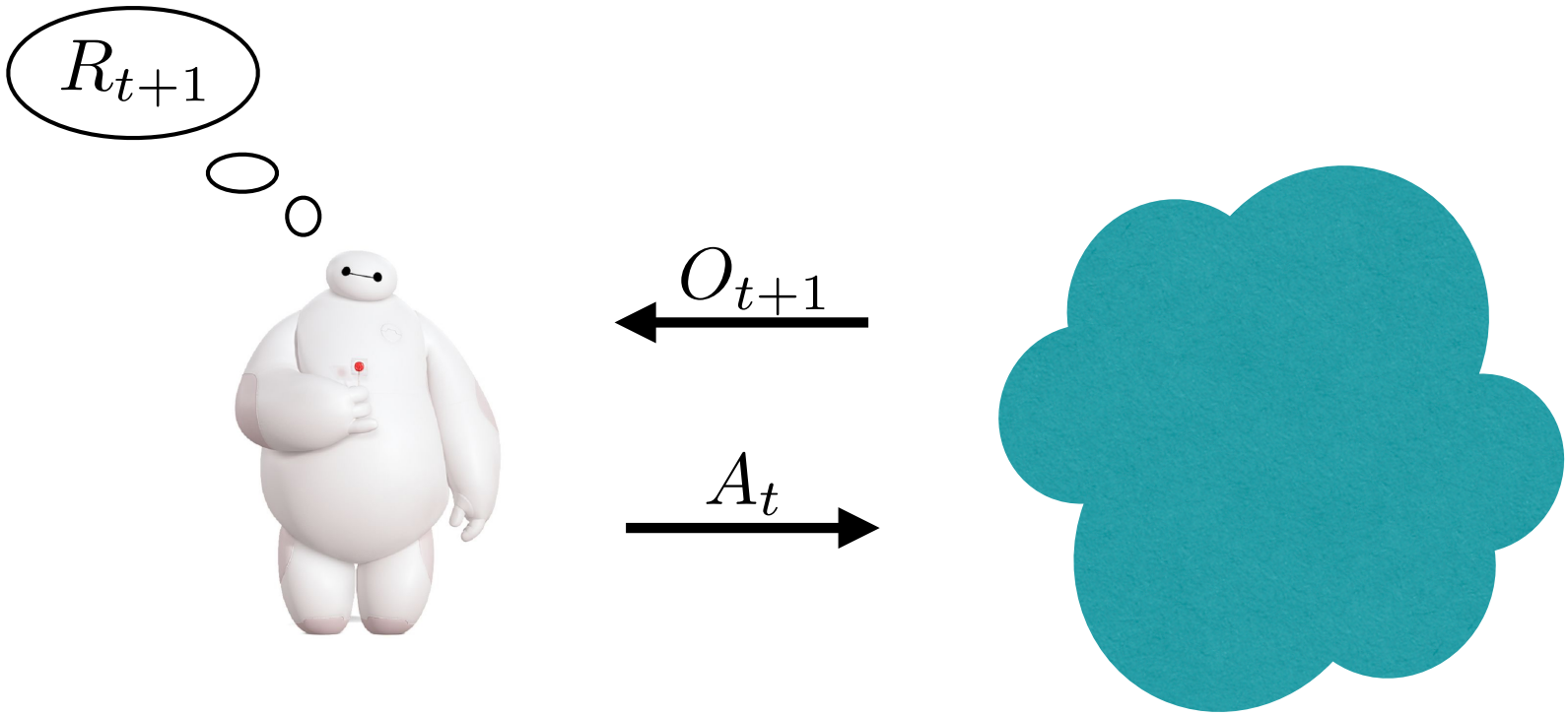
motivation via simple example

- Thompson sampling for the linear bandit
- ensemble sampling for the linear bandit
- ensemble sampling for deep rl

bandits

agent

environment



bandit: action influences only the next observation

linear bandit

$$\mathcal{A} \subseteq \text{unit ball}$$

$$R_{t+1} = O_{t+1}$$

$$R_{t+1} = \theta^\top A_t + W_{t+1}$$

$$\theta \sim \mathcal{N}(\mu_0, \Sigma_0)$$

$$W_{t+1} \sim \mathcal{N}(0, \sigma^2)$$

$$\text{Regret}(T) = \sum_{t=0}^{T-1} (\max_{a \in \mathcal{A}} \theta^\top a - \theta^\top A_t)$$

Thompson sampling for the linear bandit

for $t = 0, 1, 2, \ldots$

$\hat{\theta}_t \sim \mathcal{N}(\mu_t, \Sigma_t)$

$A_t = \arg\max_{a \in \mathcal{A}} \hat{\theta}_t^\top a$

observe R_{t+1}

compute posterior μ_{t+1}, Σ_{t+1}

$$\begin{matrix}\mu_0 = 0 \\ \Sigma_0 = I\end{matrix} \longrightarrow \mathbb{E}[\text{Regret}(T)] = d\sqrt{T \log\left(3 + \frac{3\sqrt{2T}}{d}\right)}$$

Thompson sampling via data perturbation

for $t = 0, 1, 2, \ldots$

$$\tilde{\theta}_t \sim \mathcal{N}(\mu_0, \Sigma_0)$$

$$\tilde{W}_{t+1} \sim \mathcal{N}(0, \sigma^2)$$

$$\hat{\theta}_t = \arg\min_{\theta} \frac{1}{\sigma^2} \sum_{k=0}^{t-1} (R_{t+1} - \theta^\top A_t + \tilde{W}_{t+1})^2 + (\theta - \tilde{\theta}_t)^\top \Sigma_0^{-1} (\theta - \tilde{\theta}_t)$$

$$A_t = \arg\max_{a \in \mathcal{A}} \hat{\theta}_t^\top a$$

observe R_{t+1}

ensemble sampling

for $t = 0, 1, 2, \ldots$

$n \sim \text{unif}(1, \ldots, N)$

$A_t = \arg\max_{a \in \mathcal{A}} \hat{\theta}_n^\top a$

observe R_{t+1}

$$\mathcal{L}_n(\theta) = \frac{1}{\sigma^2} \sum_{k=0}^{t-1} (R_{t+1} - \theta^\top A_t + \tilde{W}_{t,n})^2 + (\theta - \tilde{\theta}_n)^\top \Sigma_0 (\theta - \tilde{\theta}_n)$$

$$\hat{\theta}_n = \arg\min_\theta \mathcal{L}_n(\theta)$$

as *N* grows, ensemble sampling approximates TS

ensemble sampling with incremental updating

for $t = 0, 1, 2, \ldots$

$n \sim \text{unif}(1, \ldots, N)$

$A_t = \arg\max_{a \in \mathcal{A}} \hat{\theta}_n^\top a$

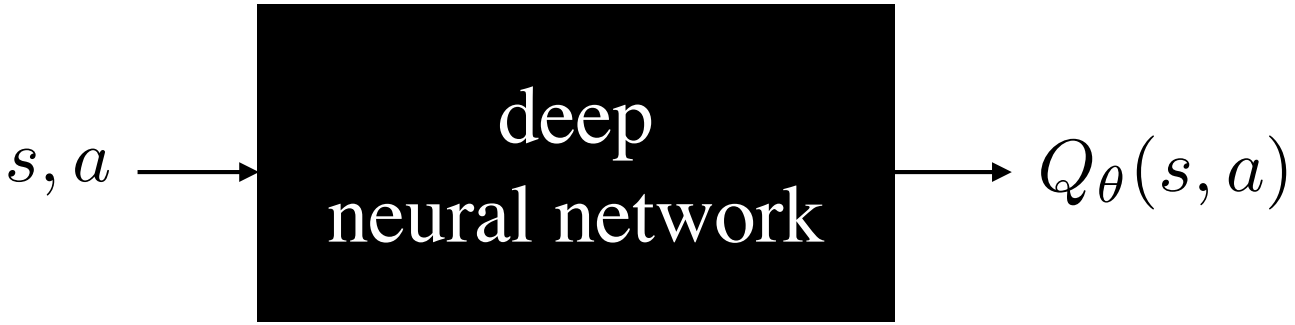
observe R_{t+1}

$$\mathcal{L}_n(\theta) = \frac{1}{\sigma^2} \sum_{k=0}^{t-1} (R_{t+1} - \theta^\top A_t + \tilde{W}_{t,n})^2 + (\theta - \tilde{\theta}_n)^\top \Sigma_0 (\theta - \tilde{\theta}_n)$$

$$\hat{\theta}_n \leftarrow \hat{\theta}_n - \alpha \nabla \mathcal{L}_n(\hat{\theta}_n)$$

as *N* grows, ensemble sampling approximates TS

from the linear bandit to deep rl



- consider learning value functions $Q_{\theta_1}, \ldots, Q_{\theta_N}$
- minimize perturbed loss functions $\hat{\theta}_n \leftarrow \hat{\theta}_n - \alpha \nabla \mathcal{L}_n(\hat{\theta}_n)$

occasionally ← why occasionally?

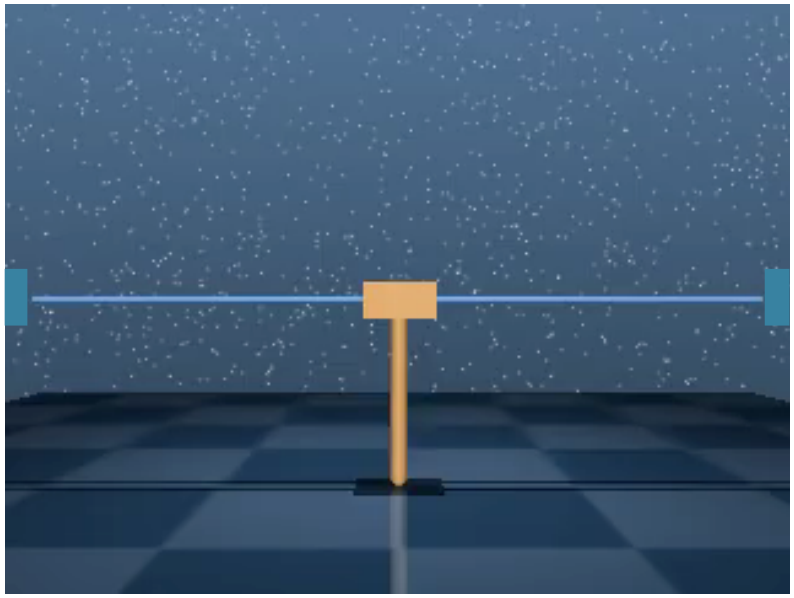
$$n \sim \text{unif}(1, \ldots, N)$$

$$A_t = \arg\max_{a \in \mathcal{A}} Q_{\theta_n}(S_t, a)$$

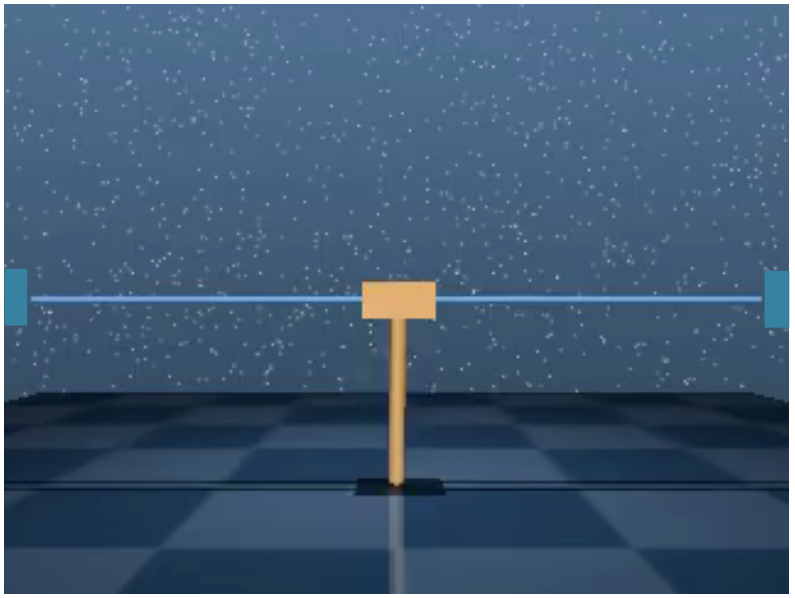
observe O_{t+1}

Cart-Pole with Sparse Reward

dithering

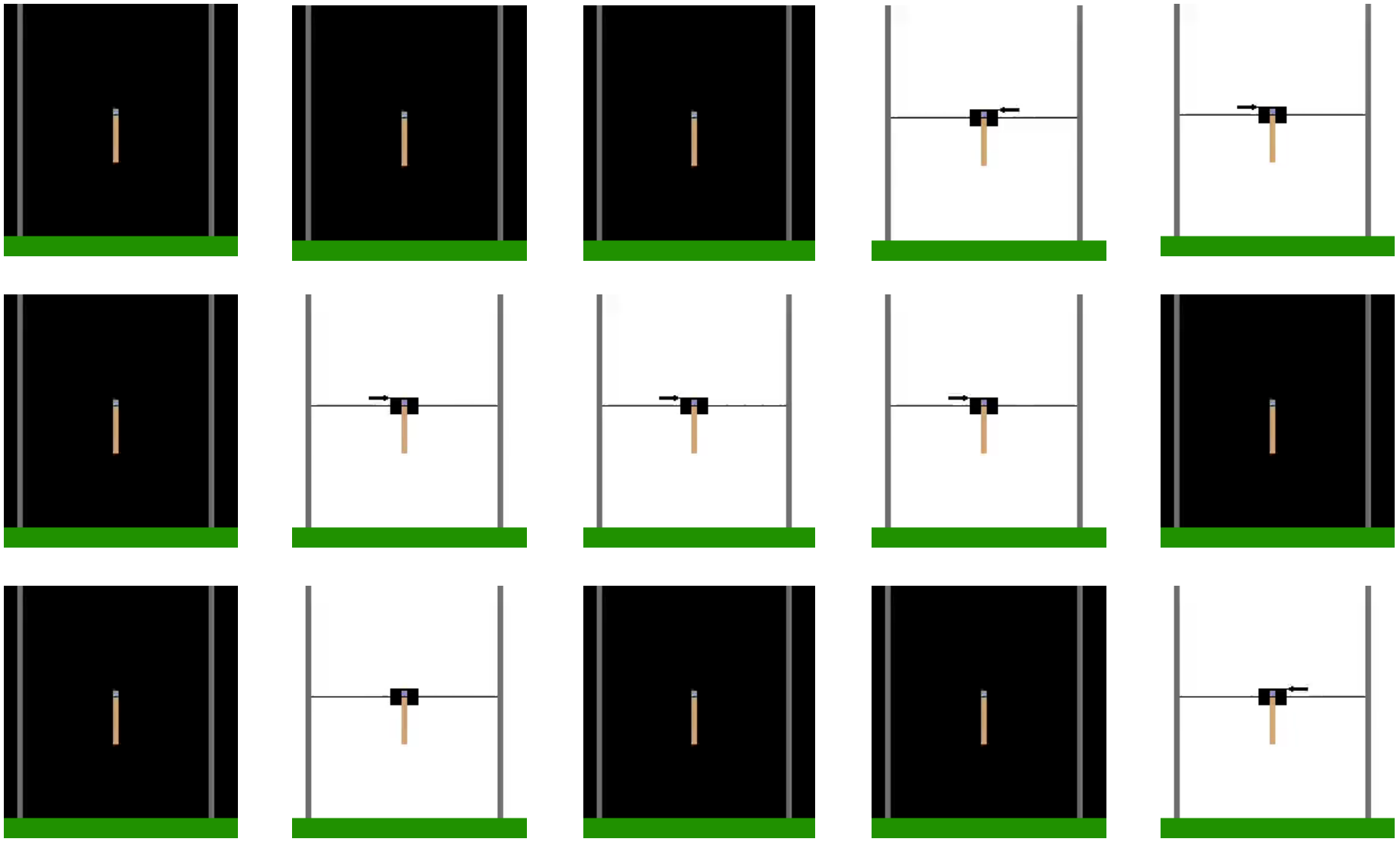


ensemble



opportunity for faster learning through
data sharing and coordinated exploration

Coordinated Exploration via Bootstrapped DQN

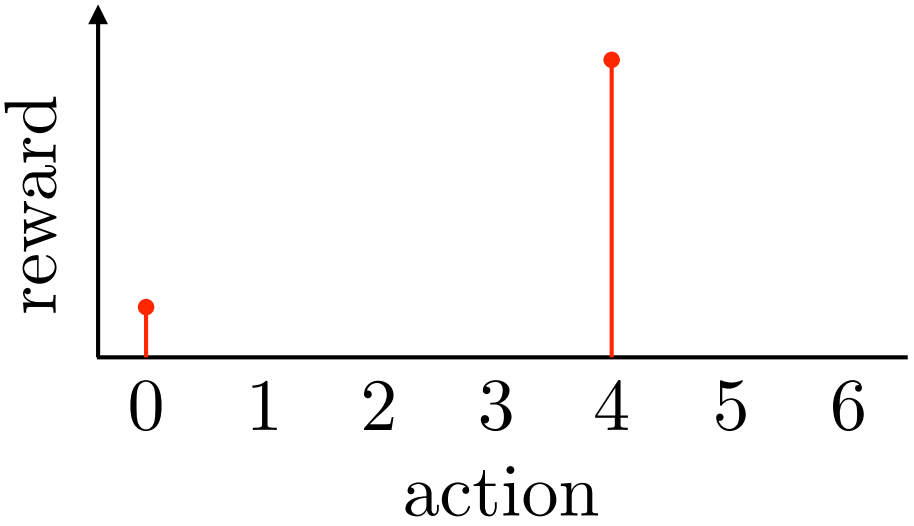


100 workers total

Can we do better than Thompson Sampling?

- TS suitable tries actions that might be optimal
  - works well for linear bandit
  - bad idea when there are complex dependencies

- example: a revealing action



$$\mathcal{A} = \{0, 1, \ldots, 6\}$$

$$R_{t+1} = \begin{cases} 1 & \text{if } A_t = \theta \\ 1/2\theta & \text{if } A_t = 0 \end{cases}$$

$$\theta = \{1, \ldots, 6\}$$

Example: Sparse Linear Bandit

- a 1-sparse case

$$R_{t+1} = \theta^\top A_t$$

$$\theta \in \{0,1\}^N \quad \|\theta\|_0 = 1$$
uniform prior

each $a \in \mathcal{A}$ averages over a subset of components

- UCB/TS require $\Omega(d)$ samples to identify
  - rule out one action per period

representation learning

- easy to design algorithms for which $\log_2(d)$ suffice
  - bisection search

information-directed sampling

- can we capture the benefits of binary search?
  - across a broad class of problems
  - with a tractable algorithm

- IDS: minimize the information ratio

$$\frac{\mathbb{E}_t[R_* - R_{t+1}]^2}{\mathbb{I}_t(A_*; (A_t, R_{t+1}))}$$

  - designing practical scalable algorithms remains a challenge

benefits may be significant when
the agent aims to learn features, like in deep RL

reading

Journal of Machine Learning Research 20 (2019) 1-62 Submitted 5/18; Revised 8/19; Published 8/19

Deep Exploration via Randomized Value Functions

Ian Osband IOSBAND@GOOGLE.COM
DeepMind
Benjamin Van Roy BVR@STANFORD.EDU
Stanford University
Daniel J. Russo DJR2174@GSB.COLUMBIA.EDU
Columbia University
Zheng Wen ZWEN@ADOBE.COM
Adobe Research

Editor: Peter Auer

Abstract

We study the use of randomized value functions to guide deep exploration in reinforcement learning. This offers an elegant means for synthesizing statistically and computationally efficient exploration with common practical approaches to value function learning. We present several reinforcement learning algorithms that leverage randomized value functions and demonstrate their efficacy through computational studies. We also prove a regret bound that establishes statistical efficiency with a tabular representation.

Keywords: Reinforcement learning, exploration, value function, neural network