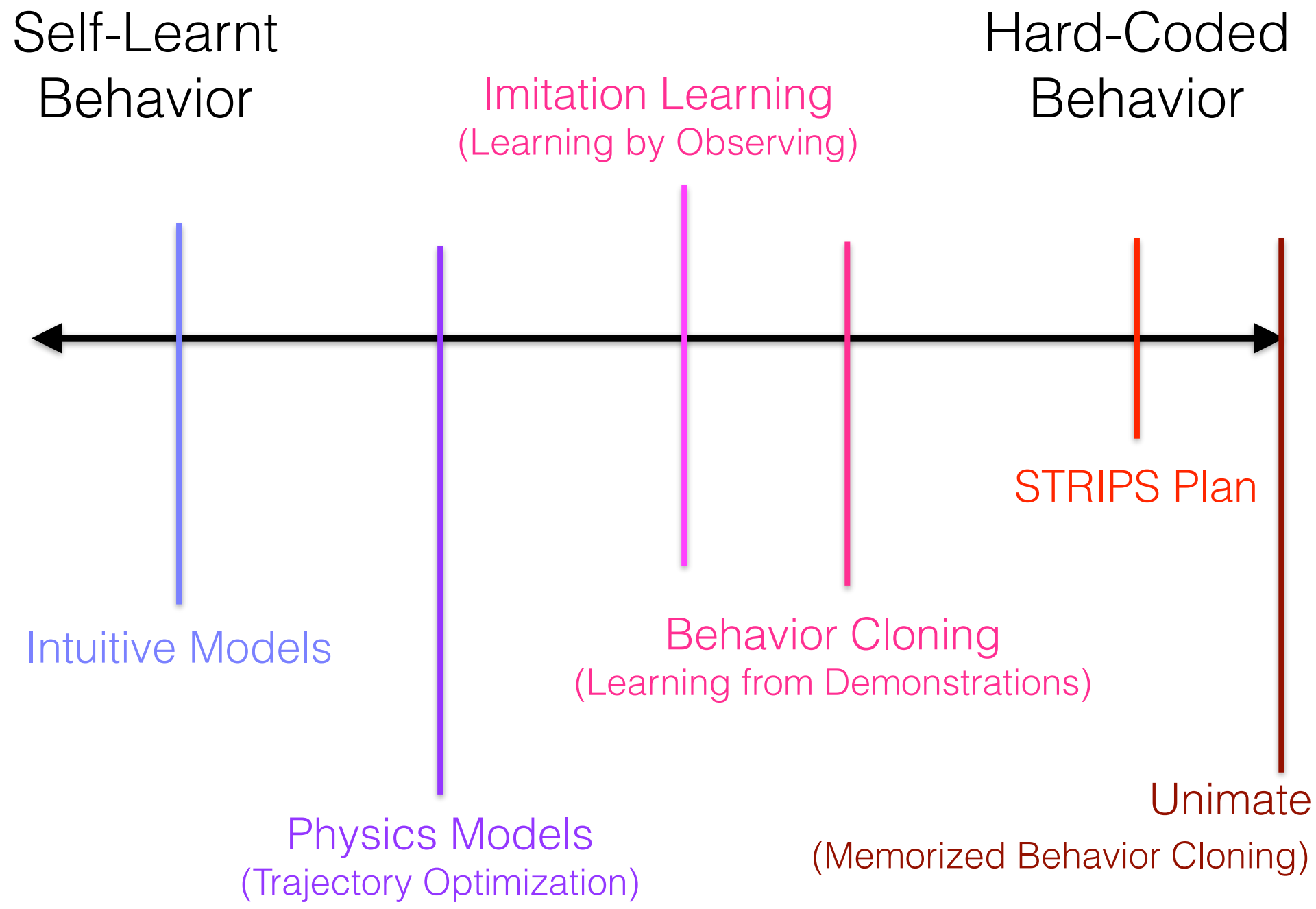
Computational Sensorimotor Learning (Spring'21)

Pulkit Agrawal

Lecture 2

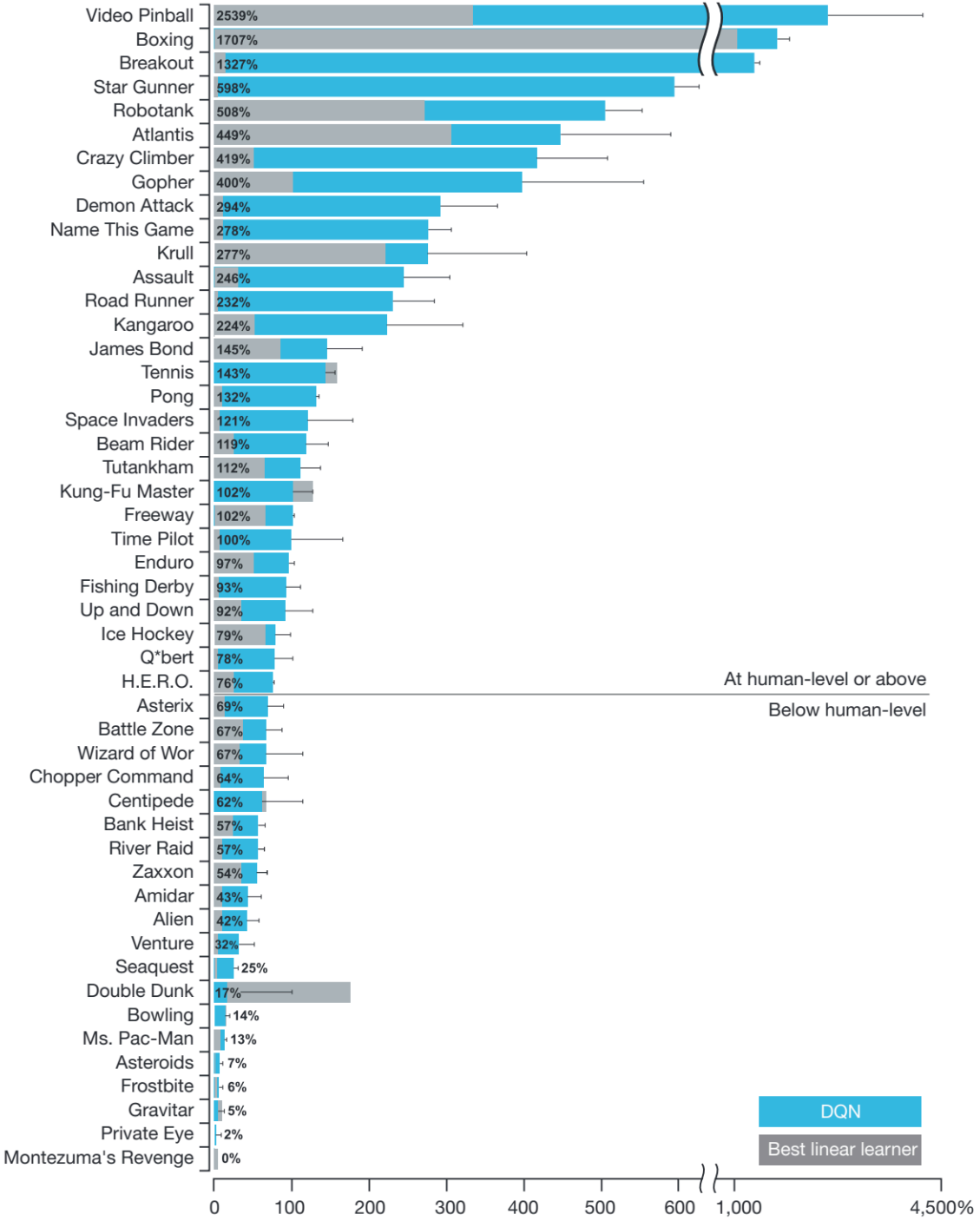
Feb 18 2021


Self-Learnt Behavior
Hard-Coded Behavior
Imitation Learning
(Learning by Observing)
STRIPS Plan
Intuitive Models
Behavior Cloning
(Learning from Demonstrations)
Unimate
Physics Models
(Trajectory Optimization)
(Memorized Behavior Cloning)

Hard to achieve

Super-Human Performance

with behavior cloning



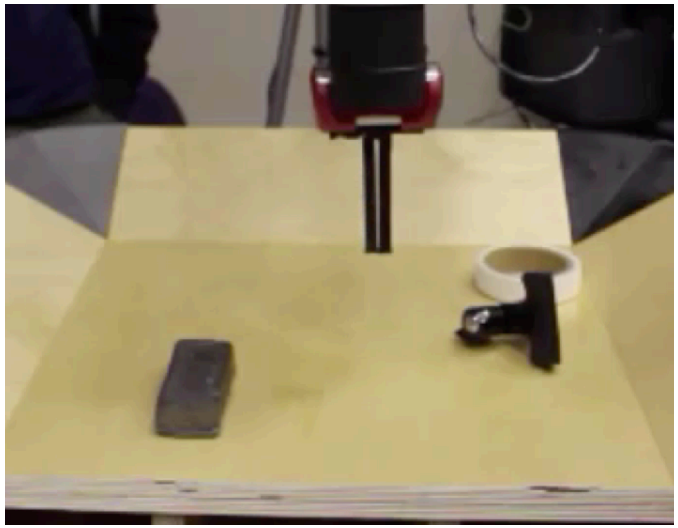
Hard to achieve

Super-Human Performance

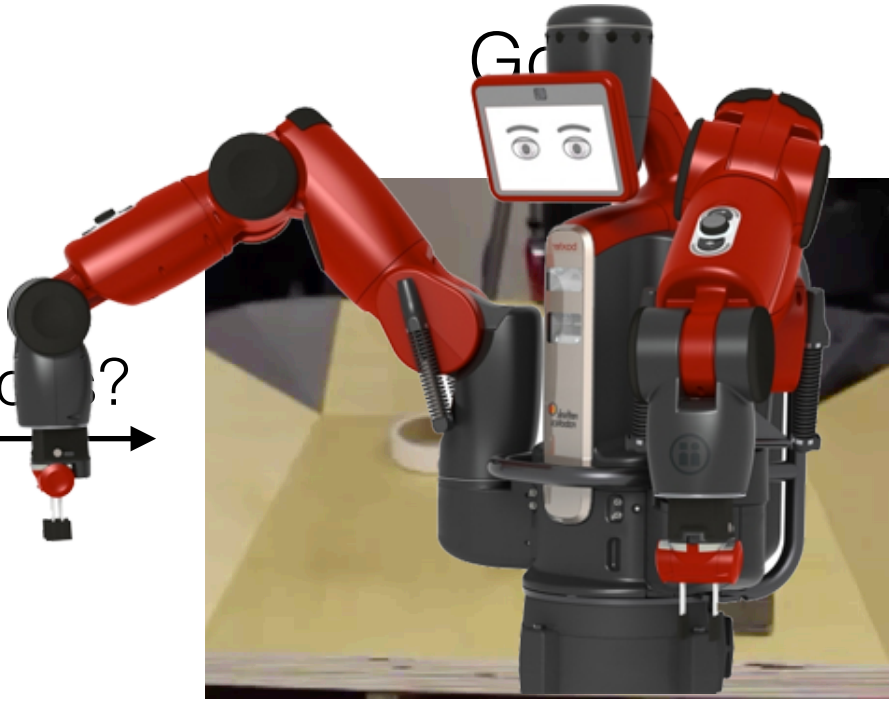
with
behavior cloning

Let the machine automatically figure out decision making rules!

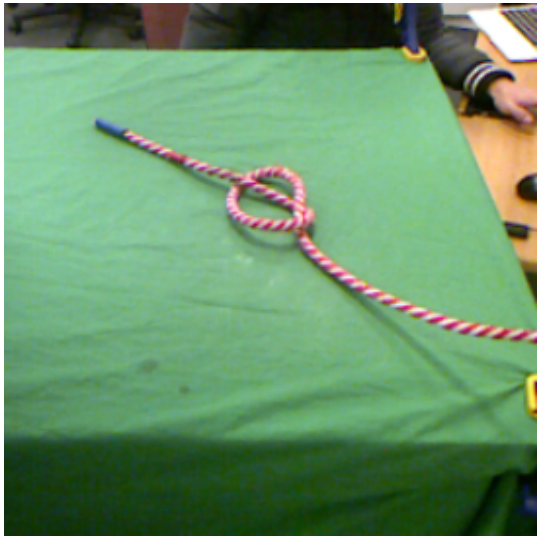
Current Observation



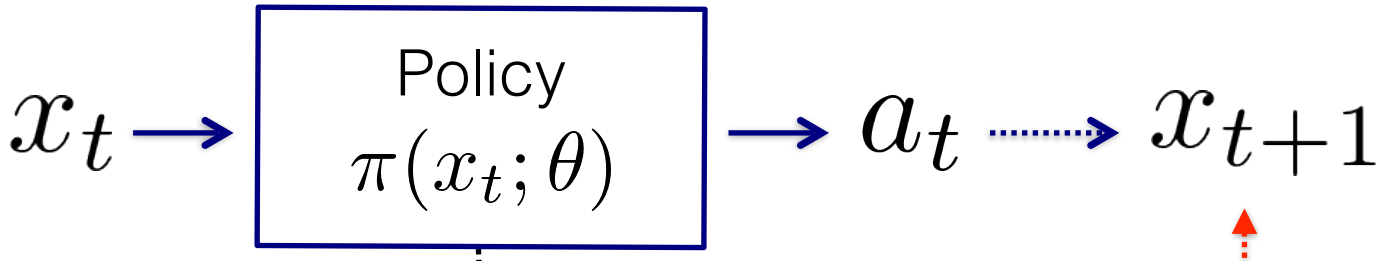
Actio?



Actions?



Brief Overview of Reinforcement Learning



initially random

(how to learn this?)

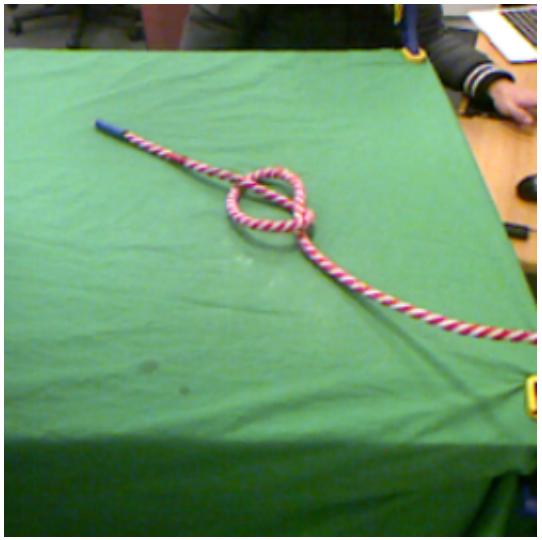
$$\max_\theta \mathbb{E}\left(\sum_{t=1}^{T} r_t\right)$$

r_{t+1}

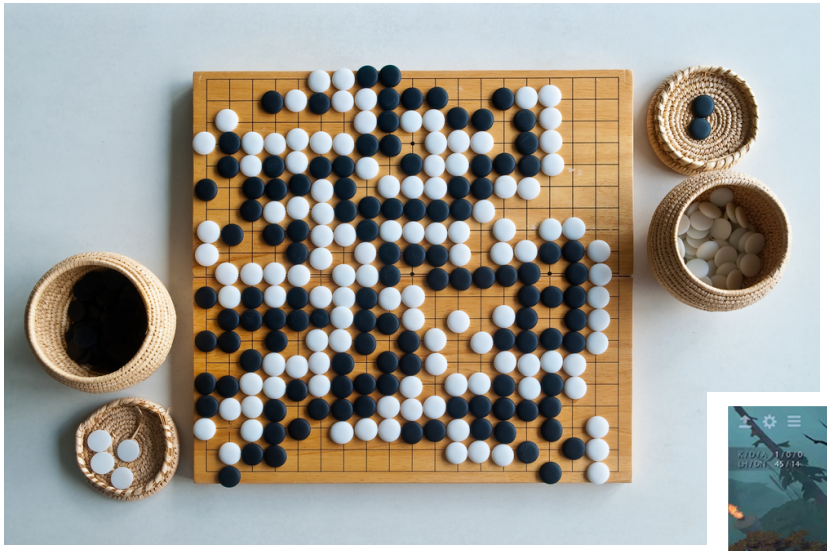
x_t

Actions?

x_G

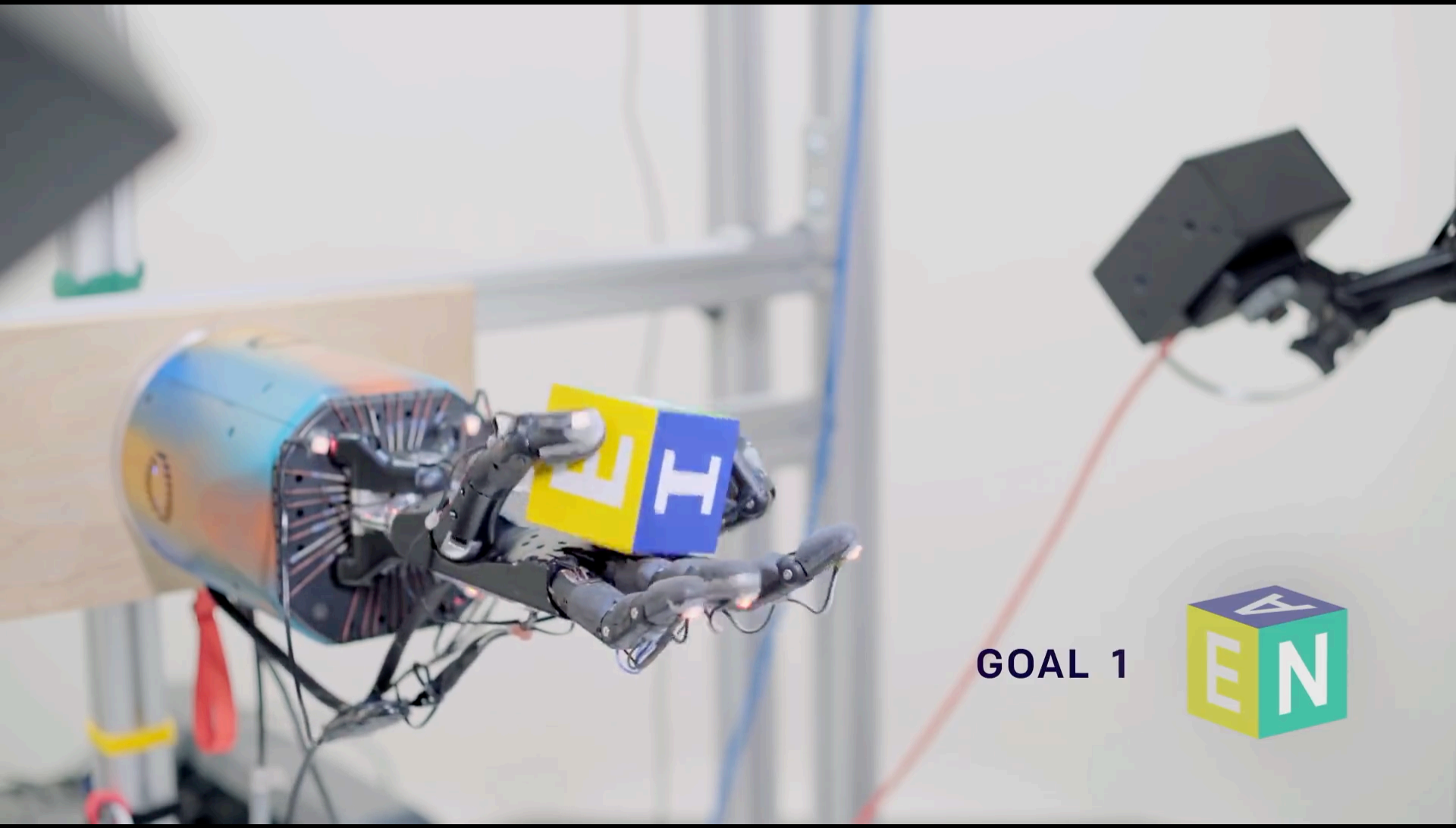


AlphaGo



Open AI Five playing DOTA

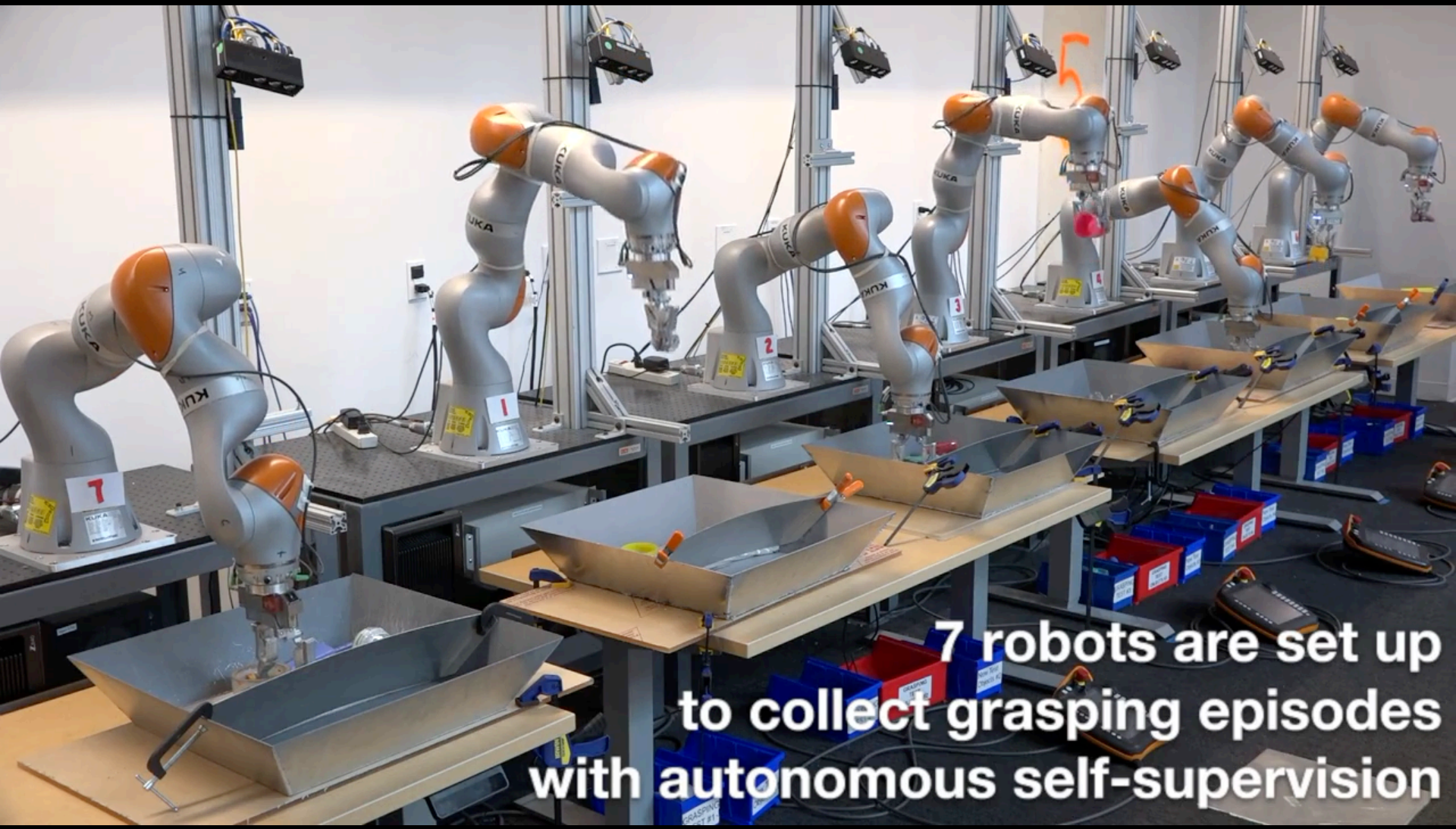
Learning Dexterity



Locomotion Strategies

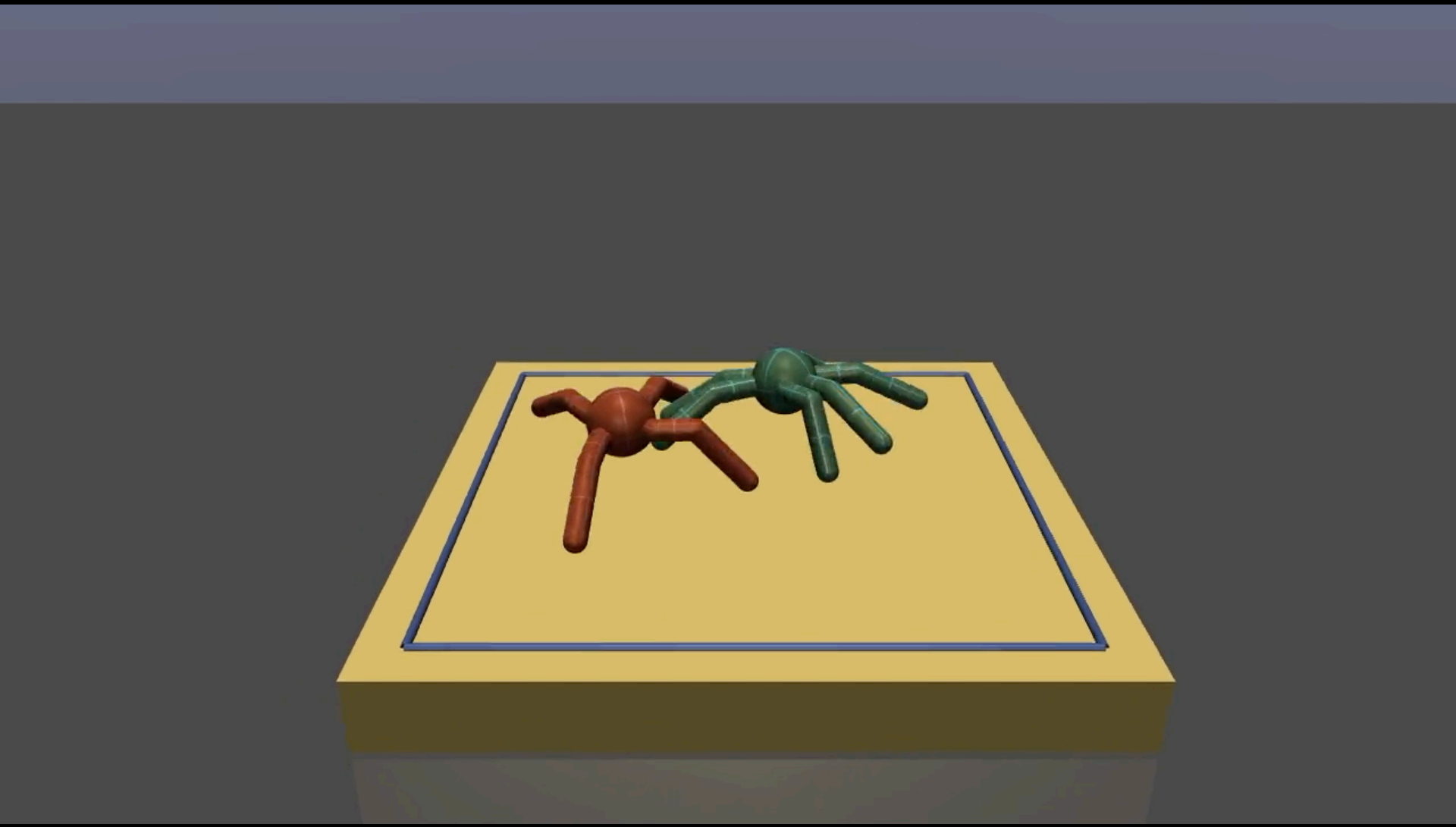
Learning Quadrupedal Locomotion over Challenging Terrain, Lee et al., 2020

Robots Learning to Grasp Objects



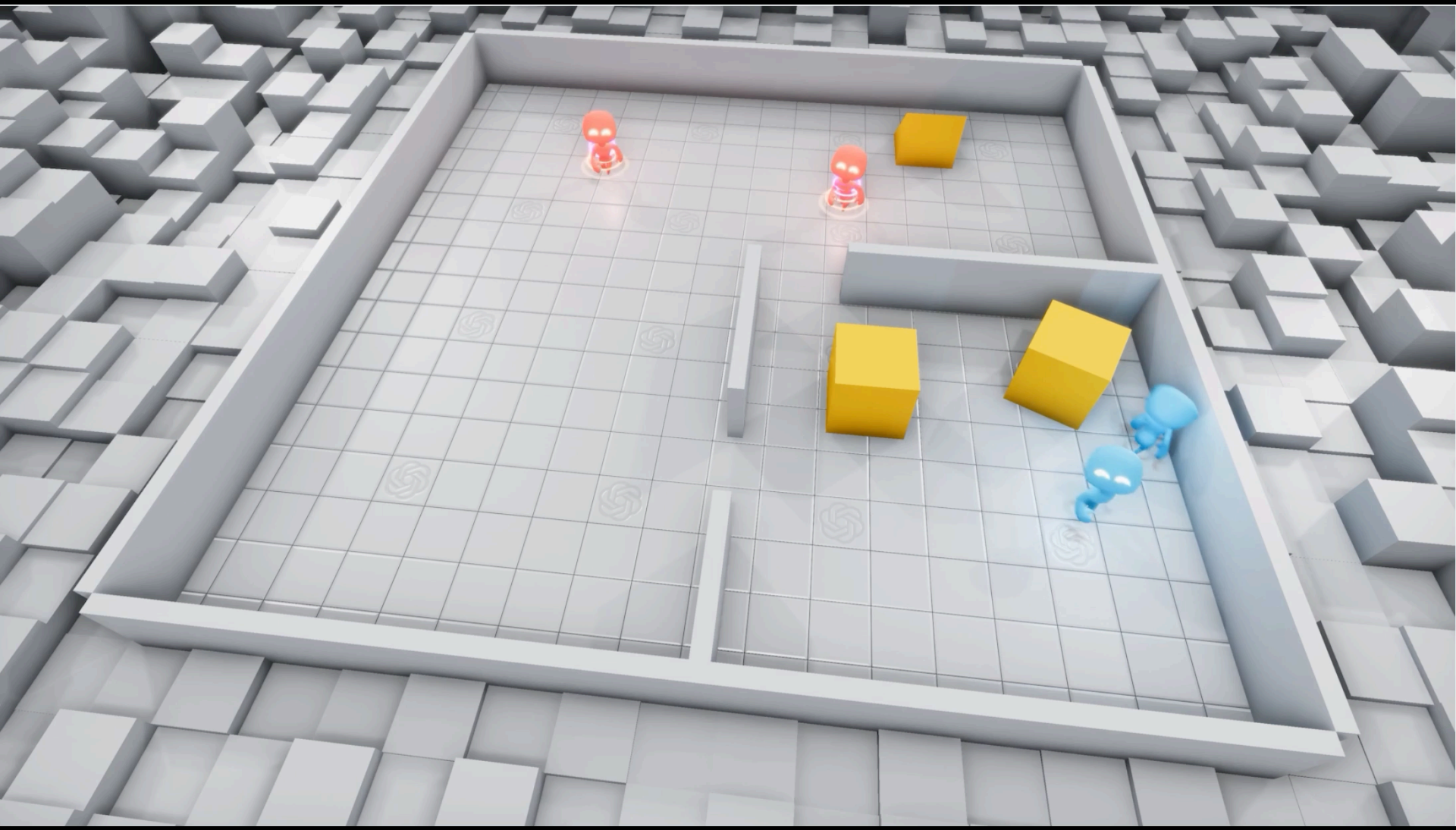
QT-OPT by Irpan et al. 2018 (Google)

Automatic Discovery of Skills



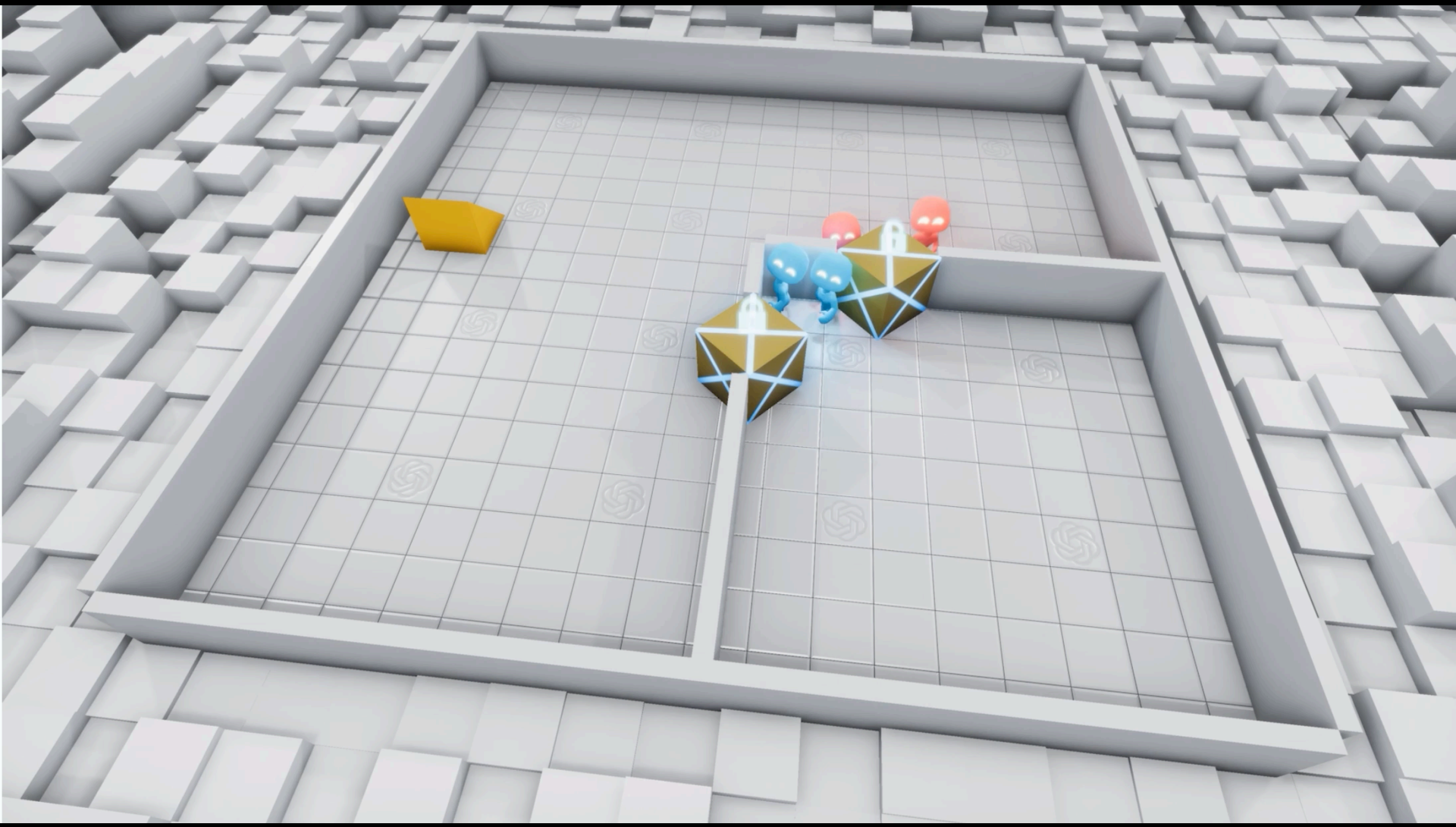
Continuous Adaptation via Meta-Learning, Al-Shedivat et al., 2018

Initial Random Exploration



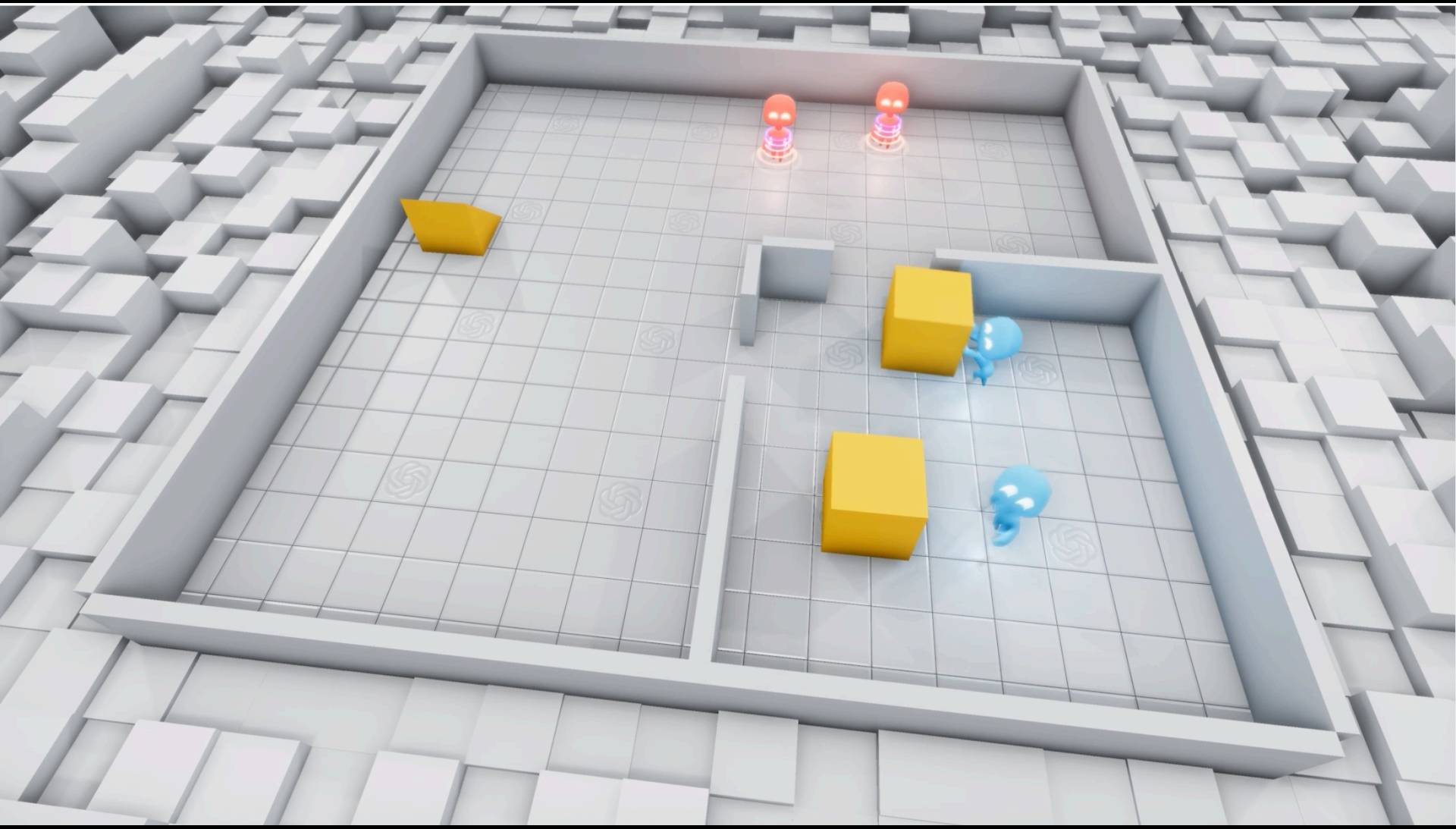
Emergent Tool Use from Multi-Agent Interaction, 2019

Use of blockers



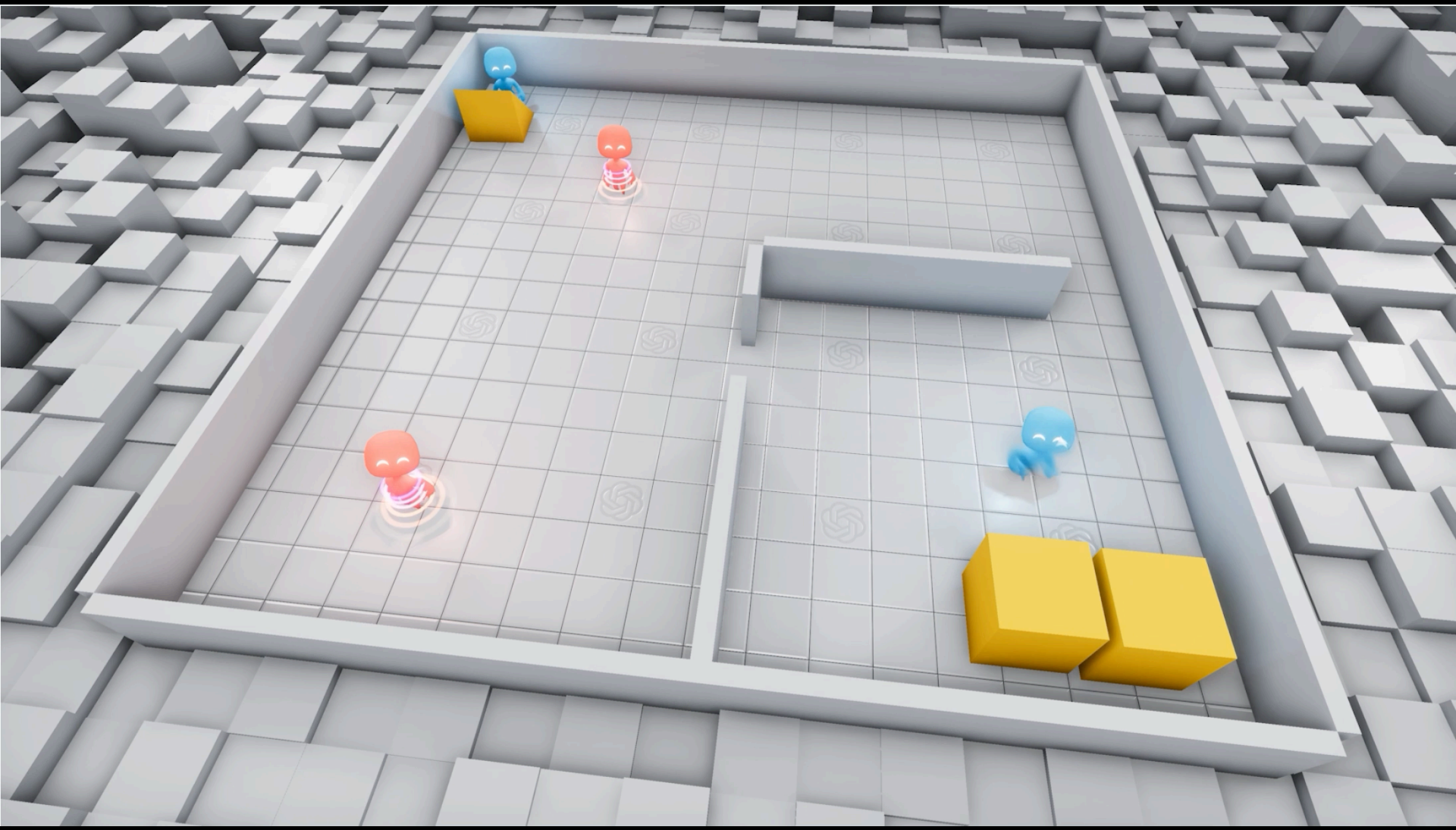
Emergent Tool Use from Multi-Agent Interaction, 2019

Using the Ramp



Emergent Tool Use from Multi-Agent Interaction, 2019

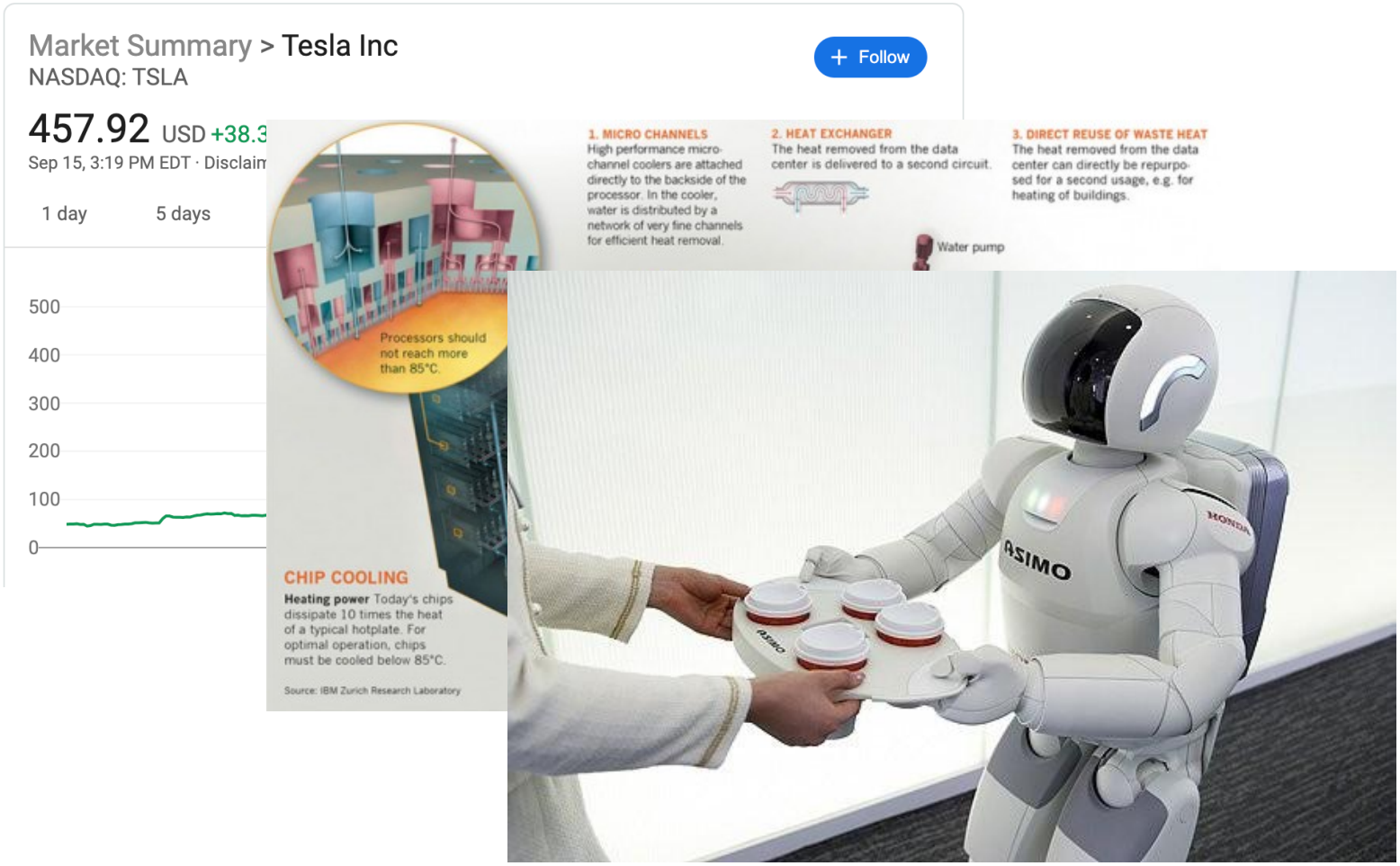
Hiding the Ramp



Emergent Tool Use from Multi-Agent Interaction, 2019

Internet to remote places: Balloon stabilization

And more problems

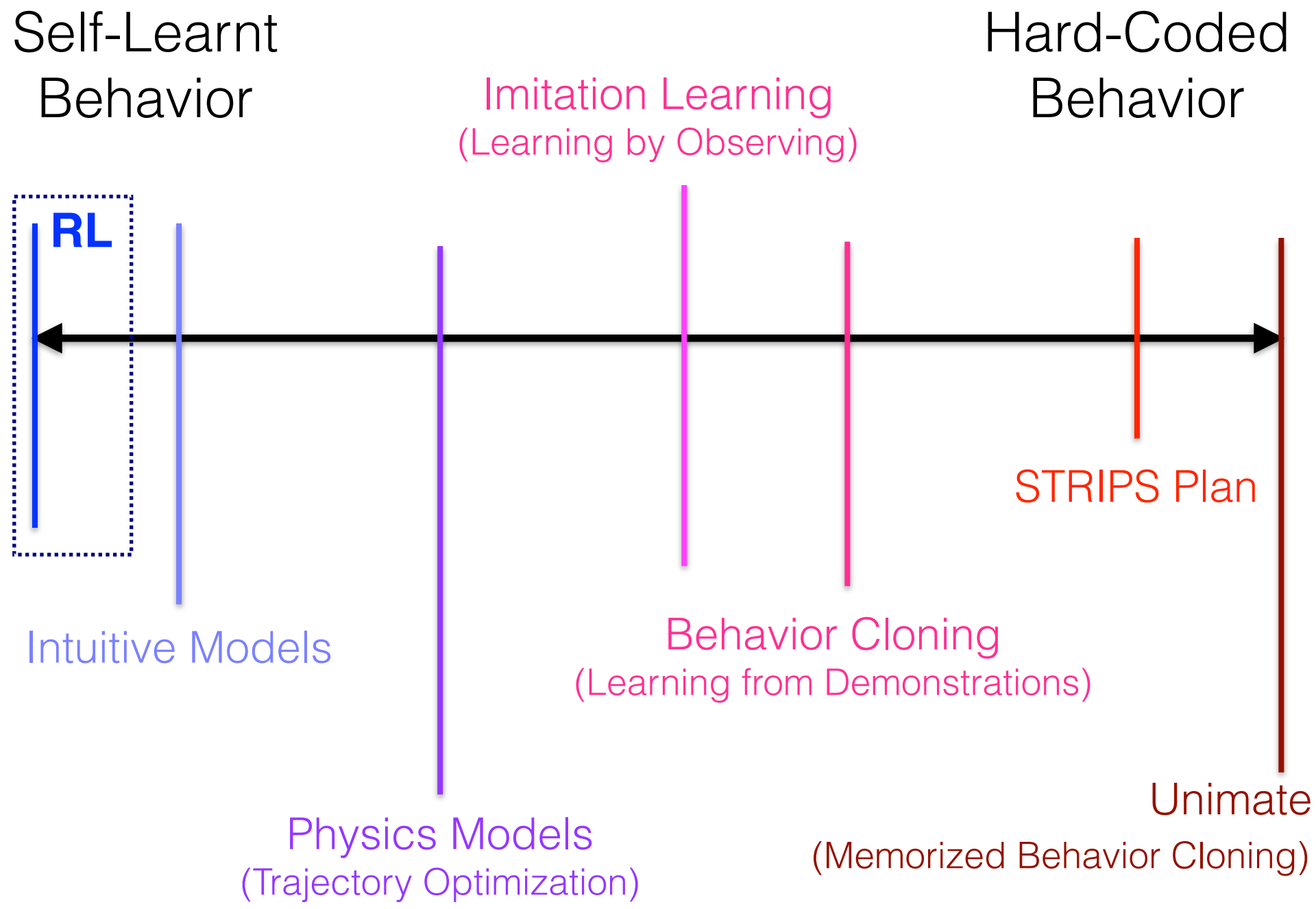


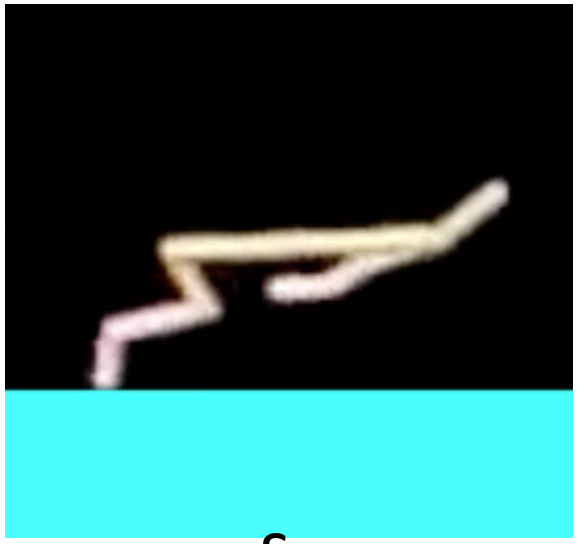
How does this all work?

Reinforcement Learning

Bandits

Contextual Bandits

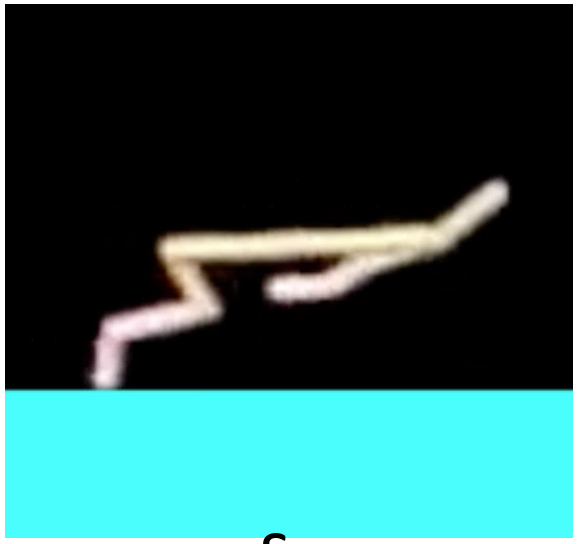

Self-Learnt Behavior
Hard-Coded Behavior
Imitation Learning
(Learning by Observing)
RL
Intuitive Models
Physics Models
(Trajectory Optimization)
Behavior Cloning
(Learning from Demonstrations)
STRIPS Plan
Unimate
(Memorized Behavior Cloning)



s_1

State (s_1, s_2 …)

- Location/rotation of joints
- Or, the image
- Or, both



s_1

State (s_1, s_2 …)

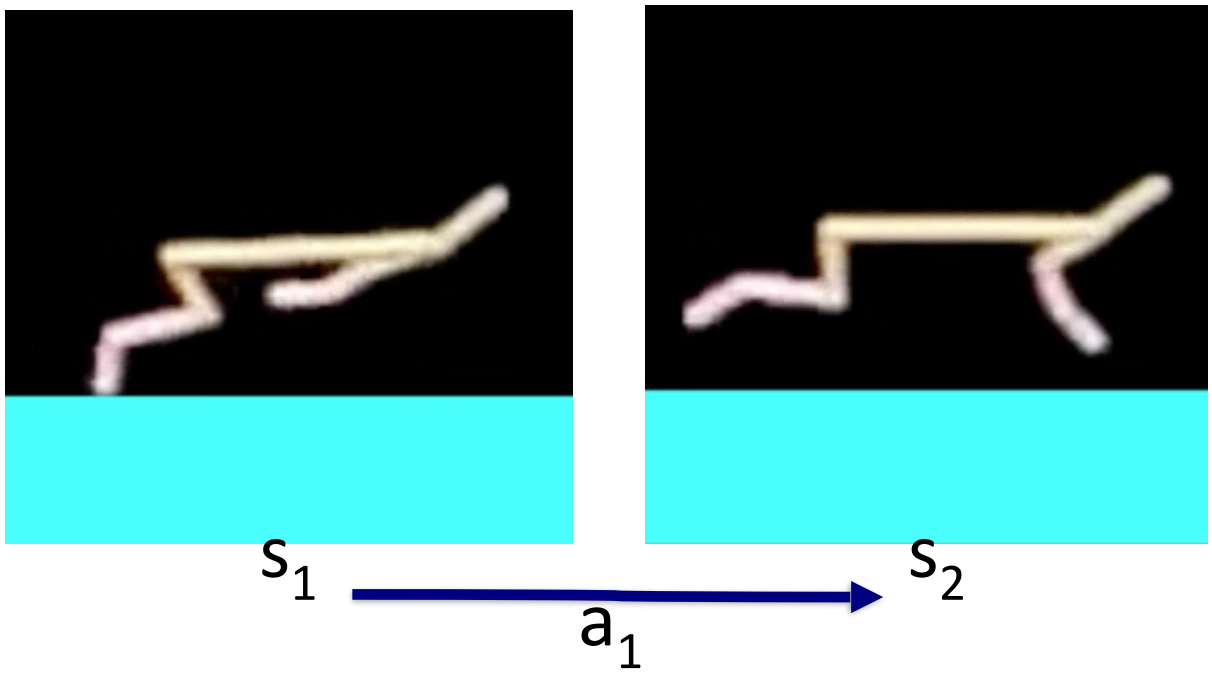
Action (a_1, a_2 …)

- Location/rotation of joints

Torques on each joint

- Or, the image

- Or, both



State (s_1, s_2 …)

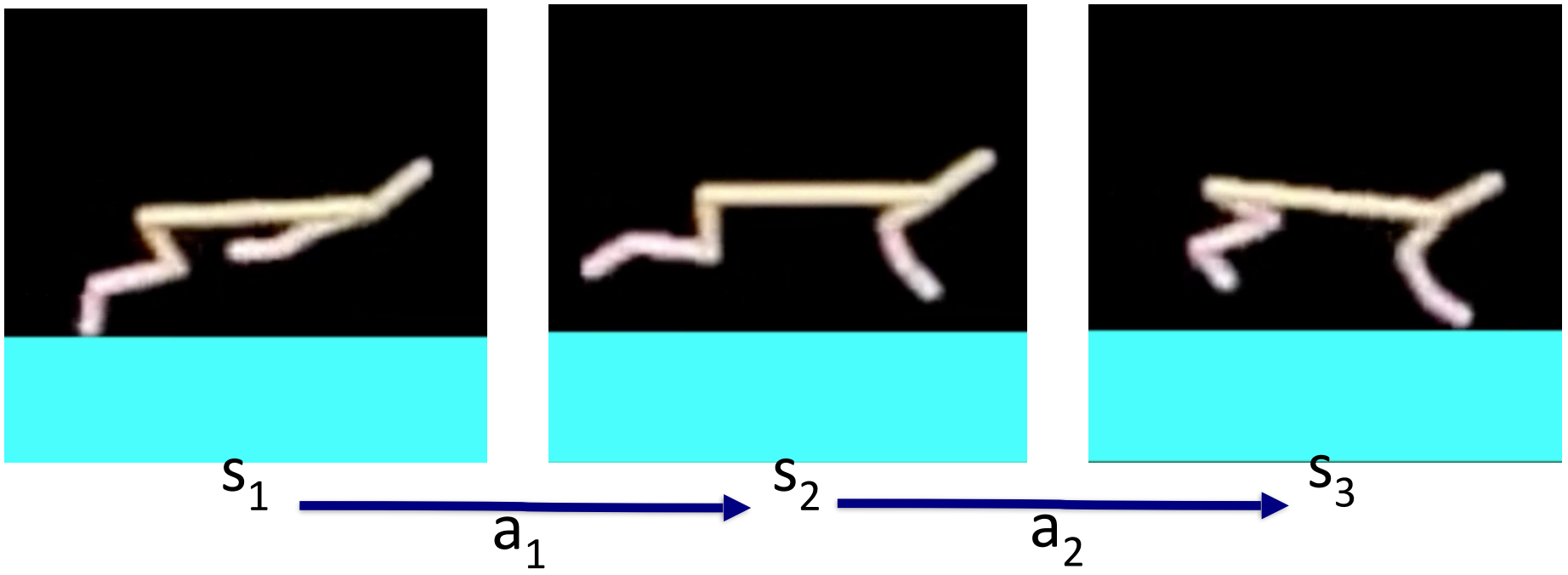
Action (a_1, a_2 …)

- Location/rotation of joints

Torques on each joint

- Or, the image

- Or, both



State (s_1, s_2 …)

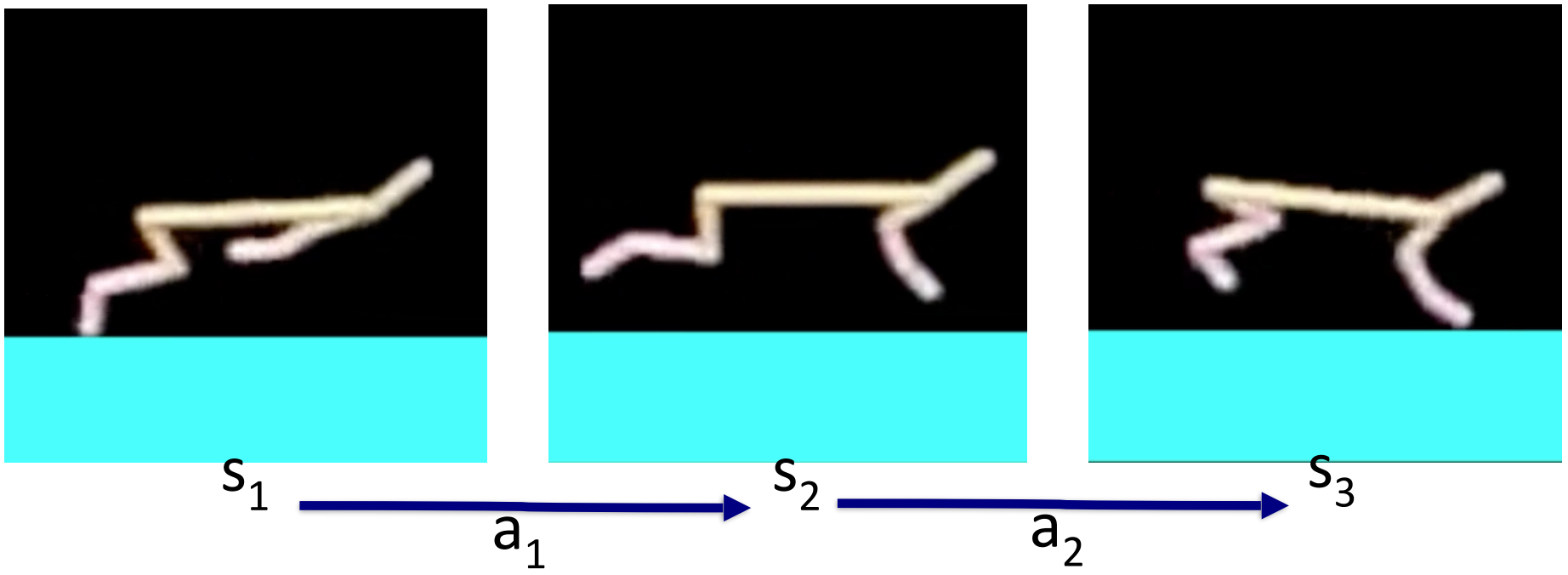
Action (a_1, a_2 …)

- Location/rotation of joints

Torques on each joint

- Or, the image

- Or, both



State (s_1, s_2 …)

Action (a_1, a_2 …)

Rewards (r_1, r_2 …)

- Location/rotation of joints

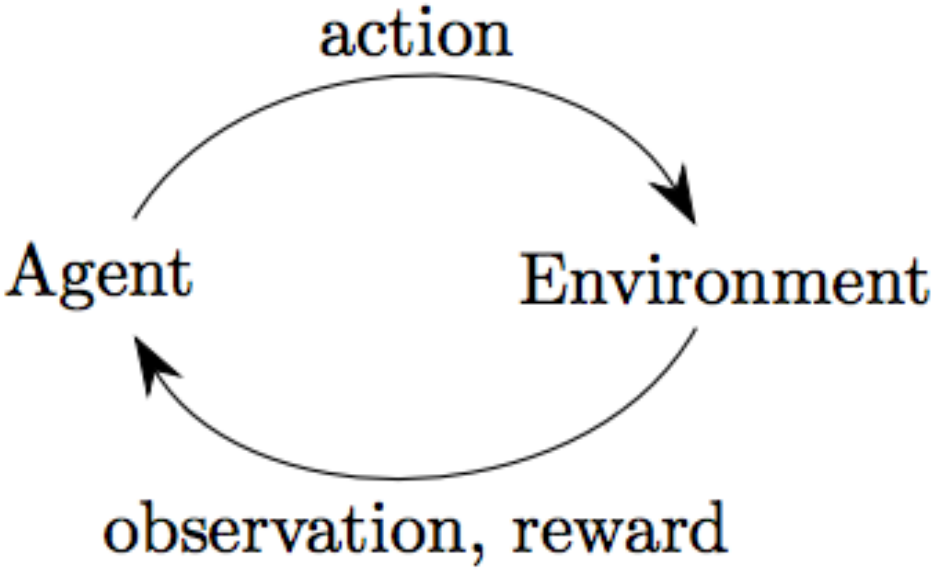
Torques on each joint

Speed of the Cheetah

- Or, the image

- Or, both

Problem Formulation



Problem Formulation

Do Actions: $a_1, a_2, a_3, \ldots, a_T$

Problem Formulation

Do Actions: $a_1, a_2, a_3, \ldots, a_T$

Get Rewards: $r_1, r_2, r_3, \ldots, r_T$

Problem Formulation

Do Actions: $a_1, a_2, a_3, \ldots, a_T$

Get Rewards: $r_1, r_2, r_3, \ldots, r_T$

$$\max \sum_{t=1}^{T} r_t$$

Problem Formulation

$$\max \sum_{t=1}^{T} r_t$$

environment model

s.t. $s_{t+1} = f(s_{0:t}, a_{0:t})$

given or learnt

Model Based Methods

policy

$a_t = \pi(s_{0:t}; \theta)$

Model Free Methods

The RL Problem

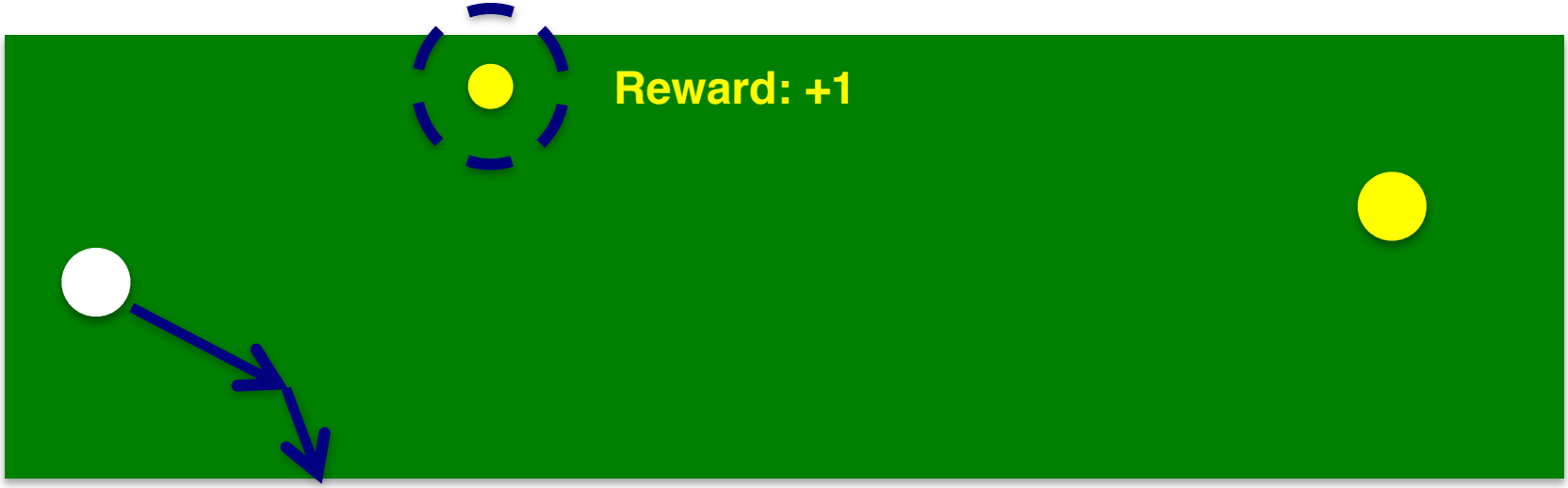
The RL Problem

The RL Problem

The RL Problem

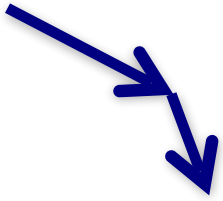
Reward: -1

Solving the MDP



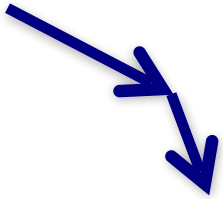
The RL Problem

Another Attempt



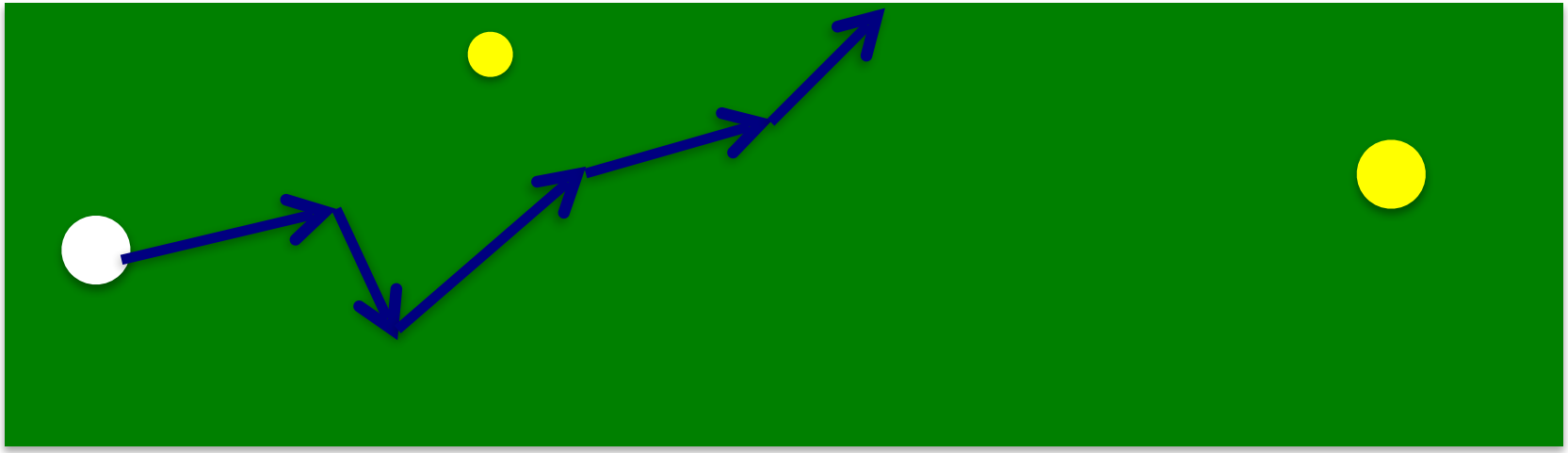
Reward: -1

Another Attempt

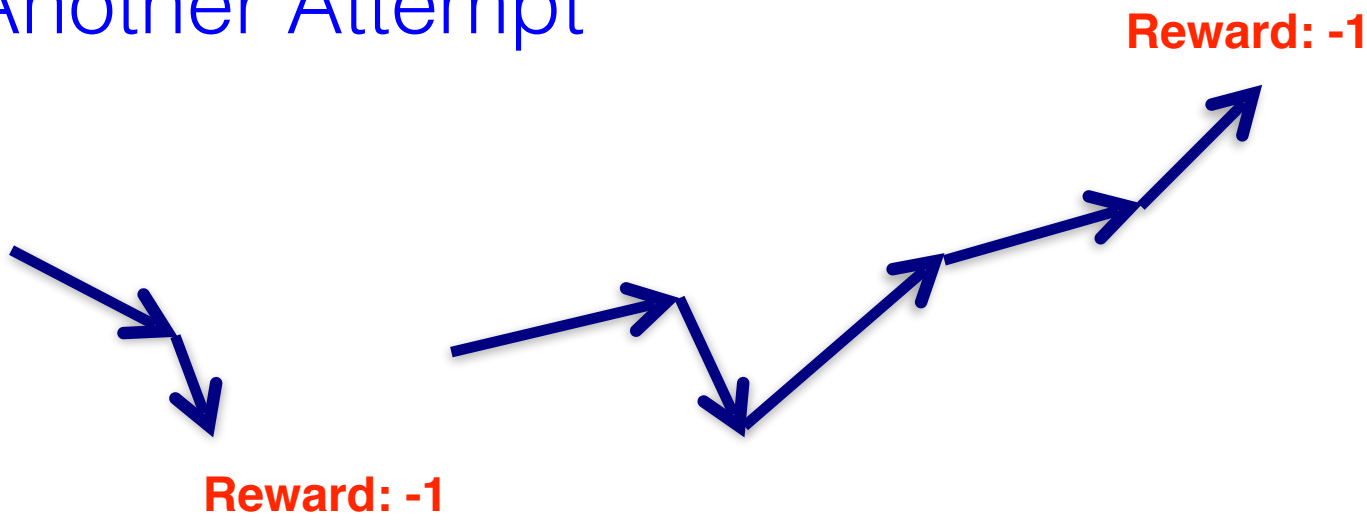


Reward: -1

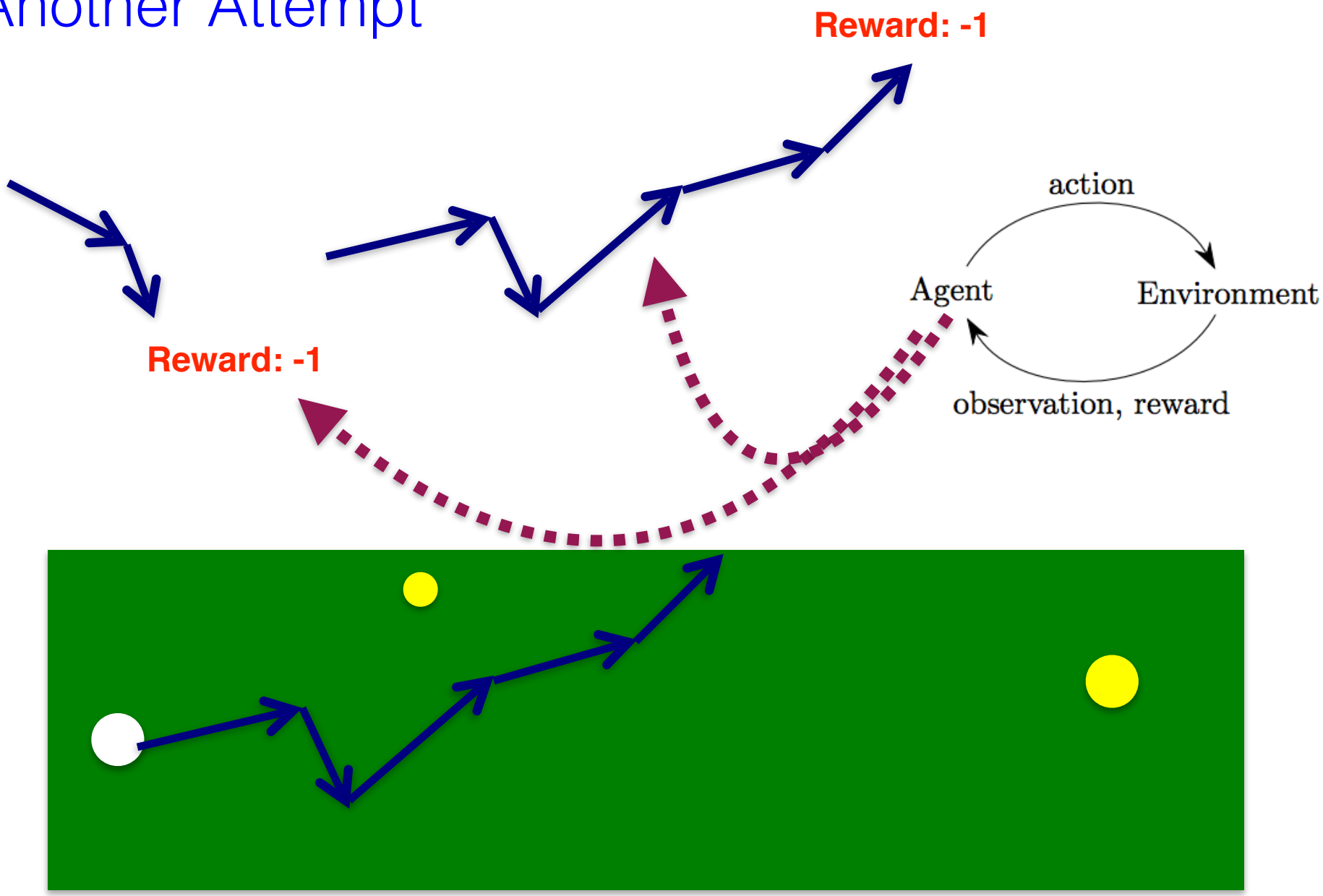
Reward: -1



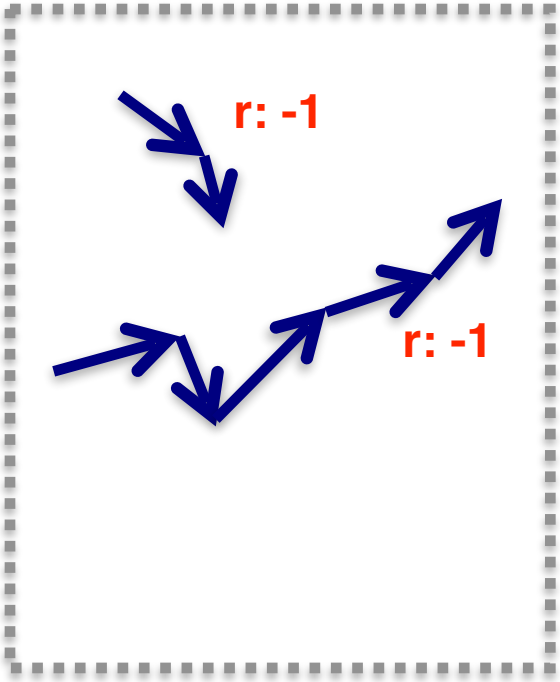
Another Attempt

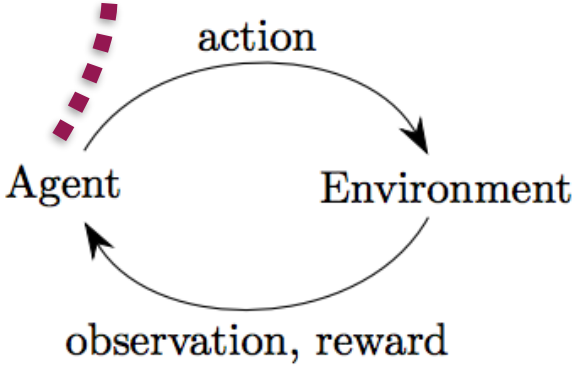


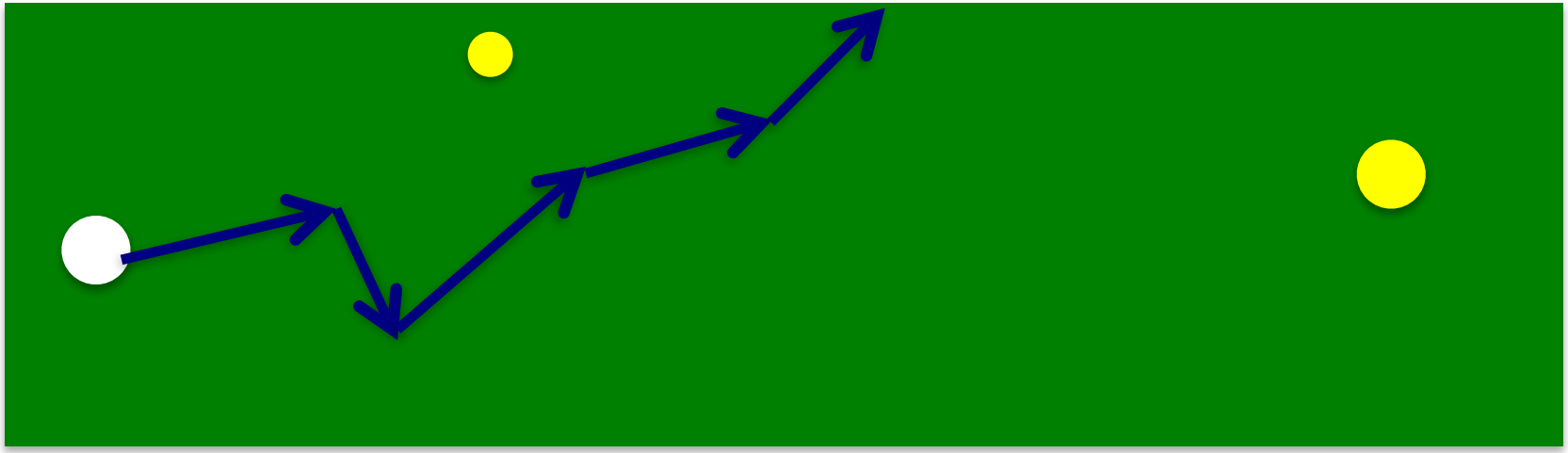
Another Attempt



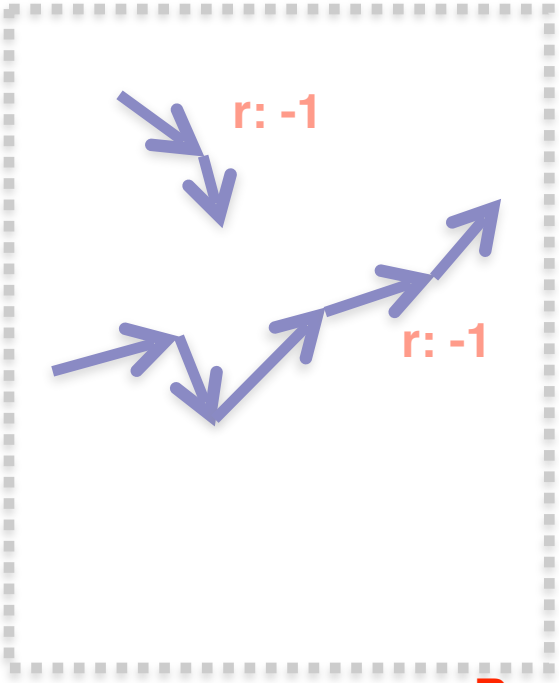
Dataset Generation

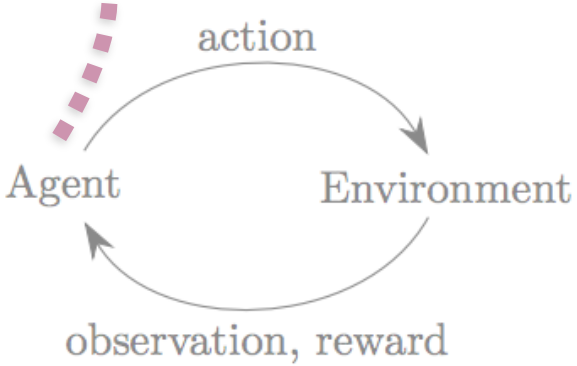






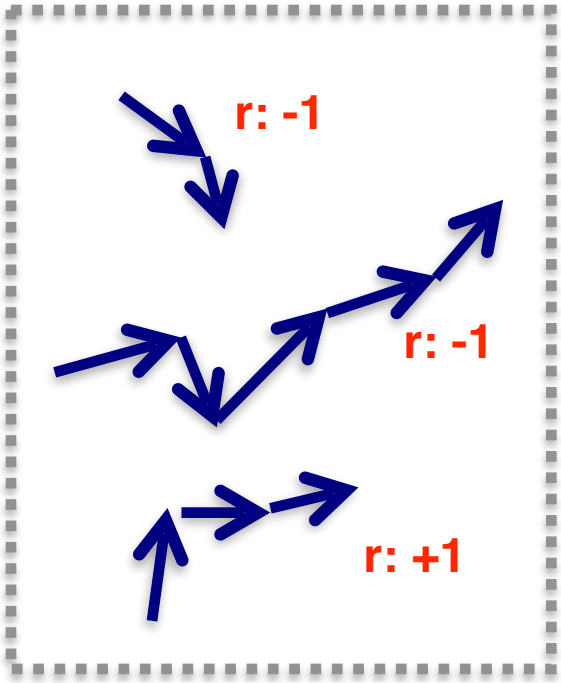
Dataset Generation

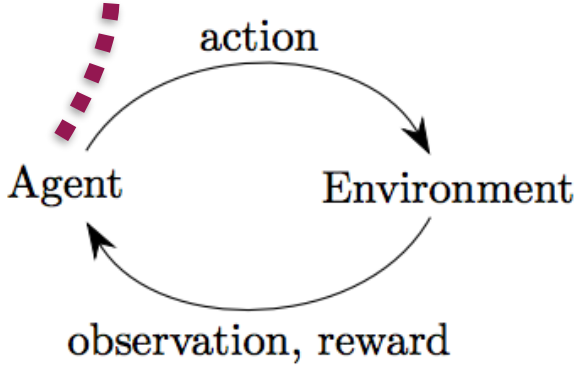




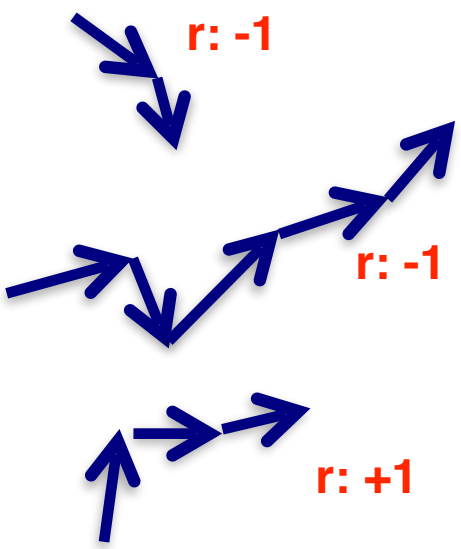
Reward: +1

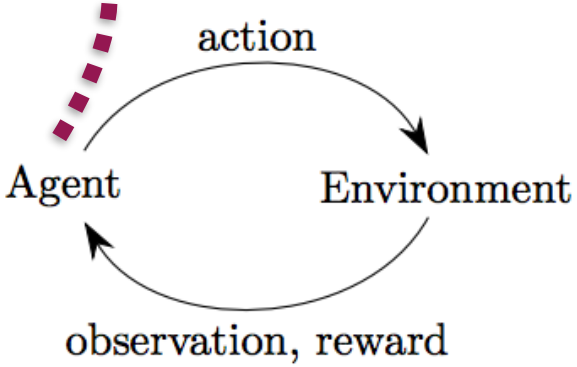
Dataset Generation





Dataset Generation





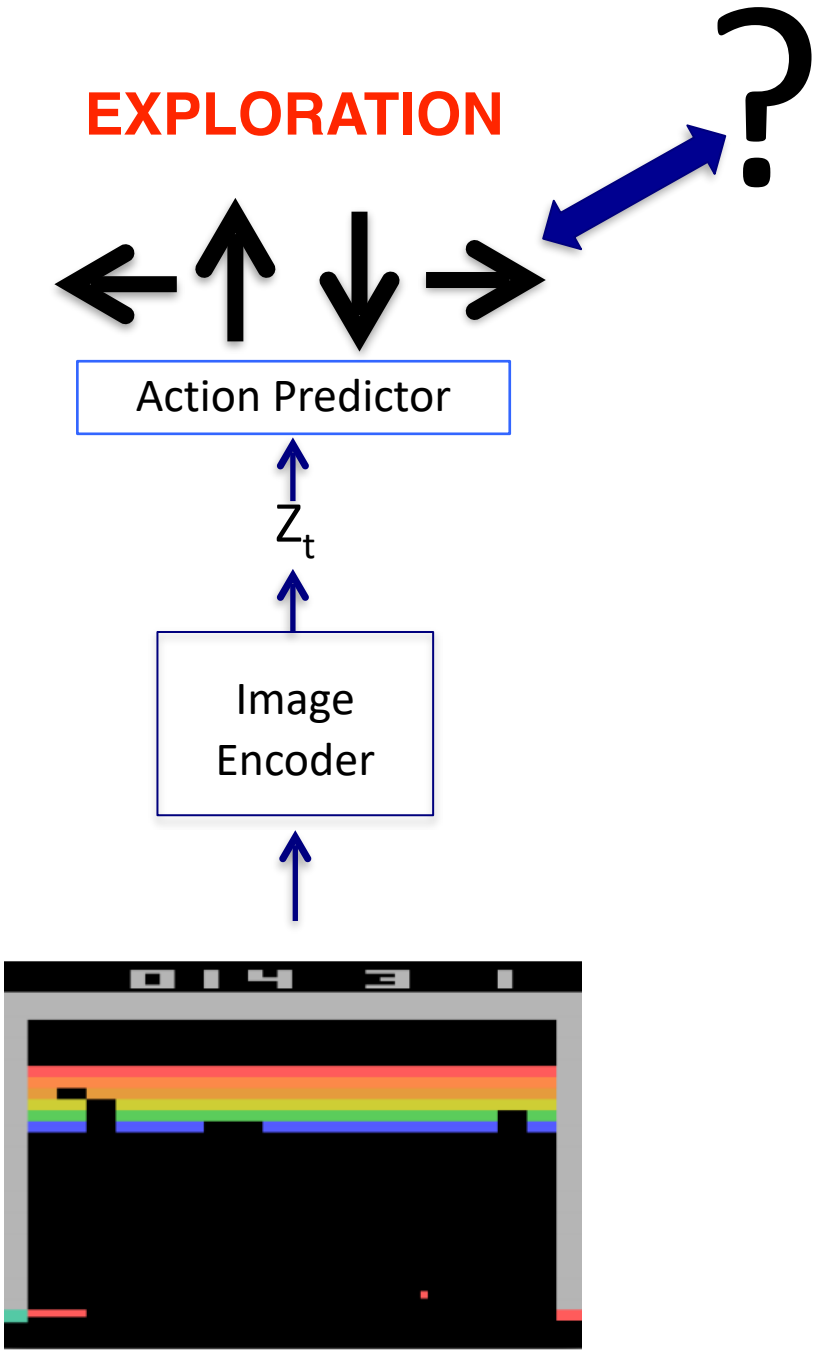
Supervised Learning

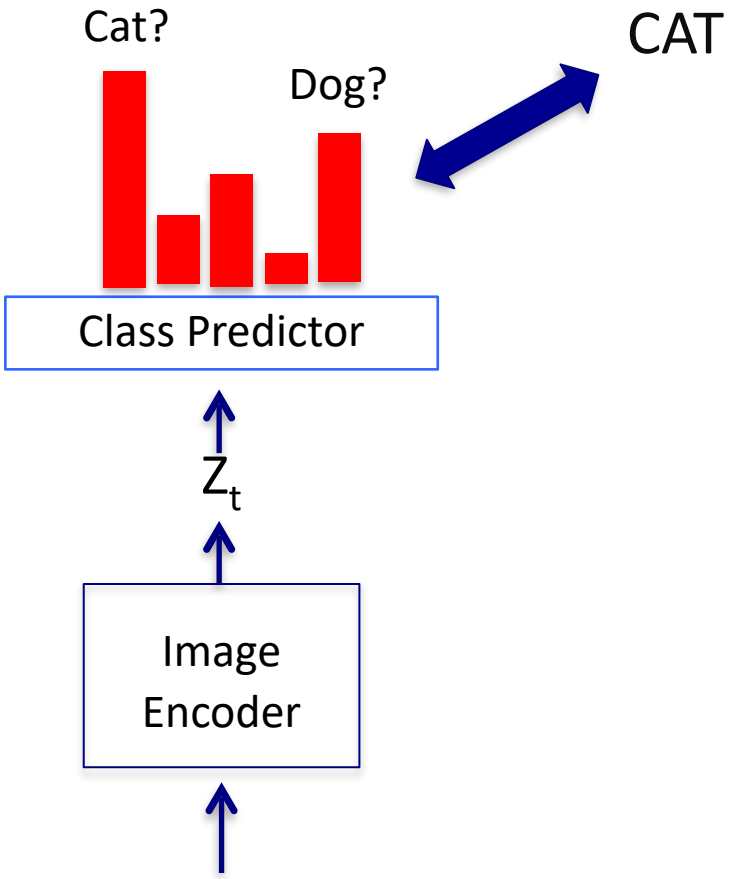
(cat)

In supervised learning, dataset is GIVEN

In reinforcement learning, the agent collects its own data

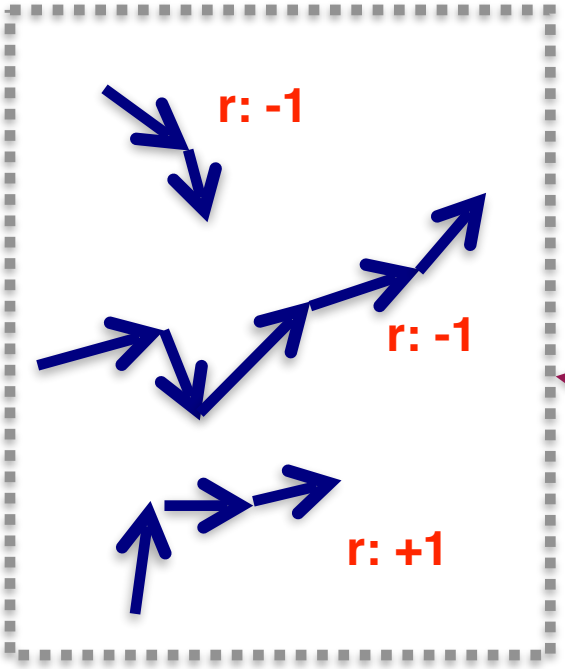
(i.e, it needs to explore)

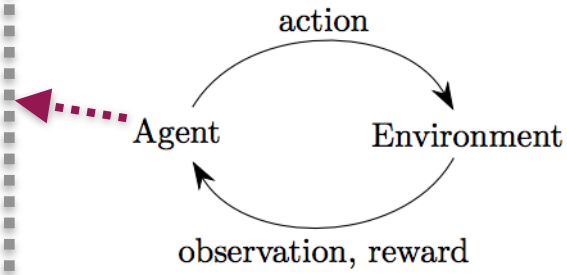

EXPLORATION
?
Action Predictor
Z_t
Image Encoder


Cat?
Dog?
CAT
Class Predictor
Z_t
Image Encoder

Dataset

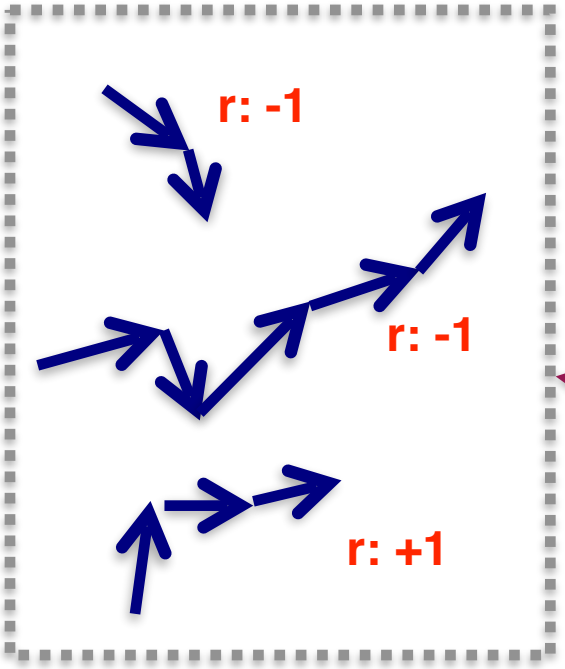
Is exploration a problem?



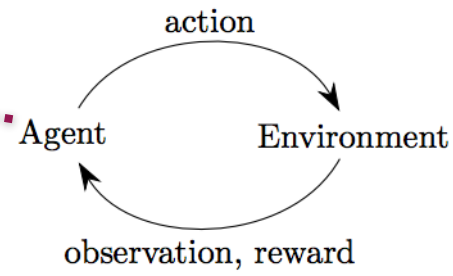


Dataset

Is exploration a problem?



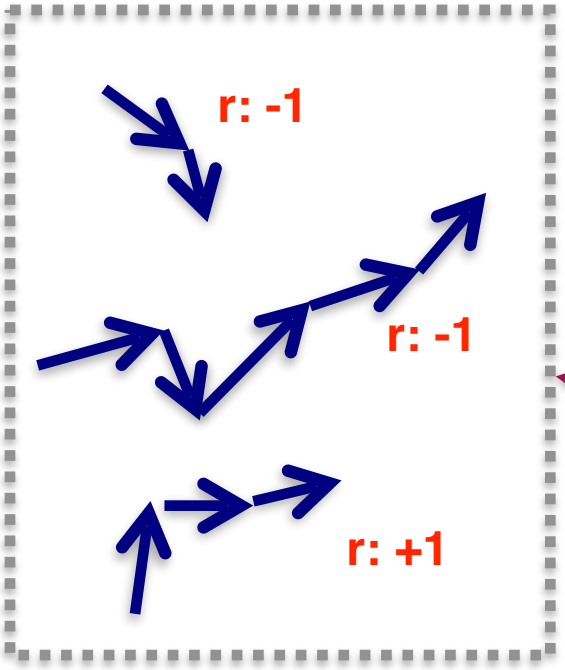
Learn



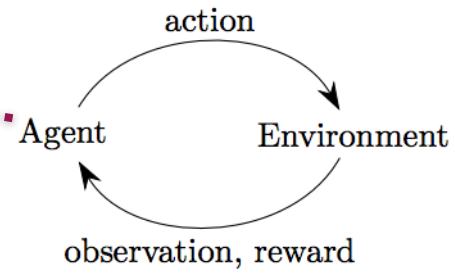
Goal

$$a_t = \pi(s_{0:t}; \theta)$$

Dataset



Learn



Is exploration a problem?

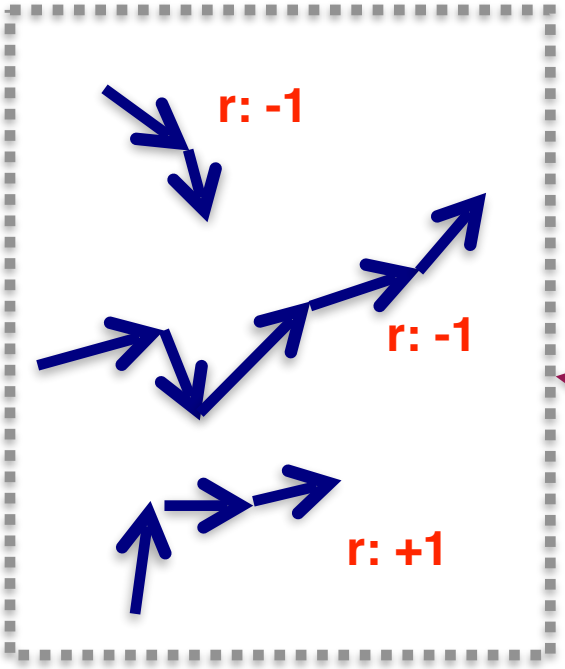
Goal

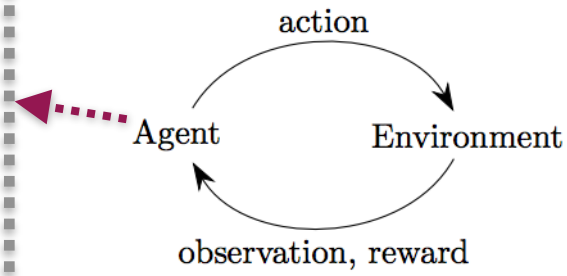
$$a_t = \pi(s_{0:t}; \theta)$$

Looks good!

(is there a problem?)

Dataset





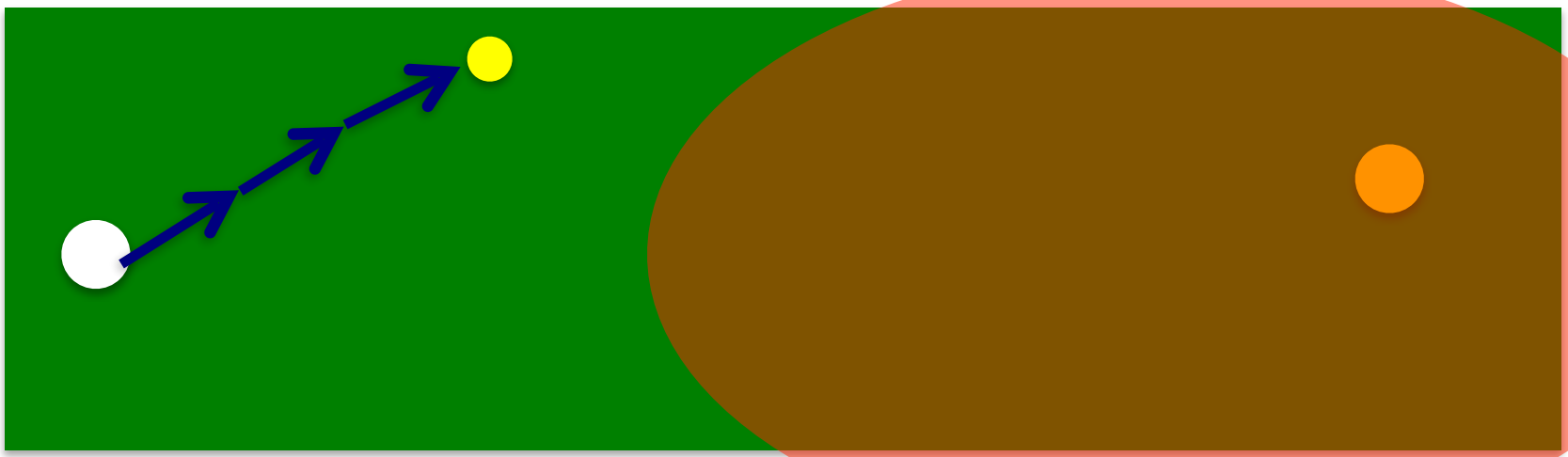
Is exploration a problem?

Goal

$$a_t = \pi(s_{0:t}; \theta)$$

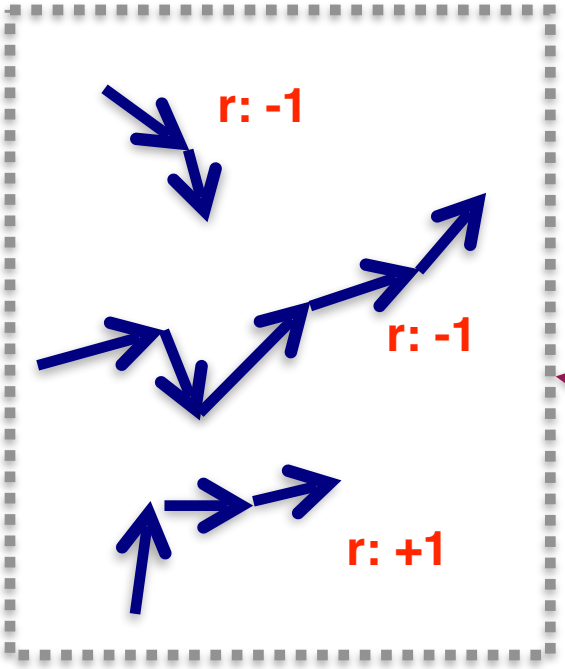
Looks good!

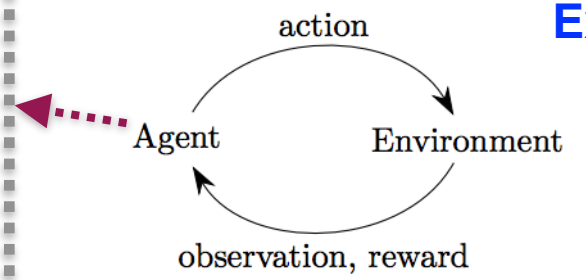
(is there a problem?)



Might not explore the state space!

Dataset

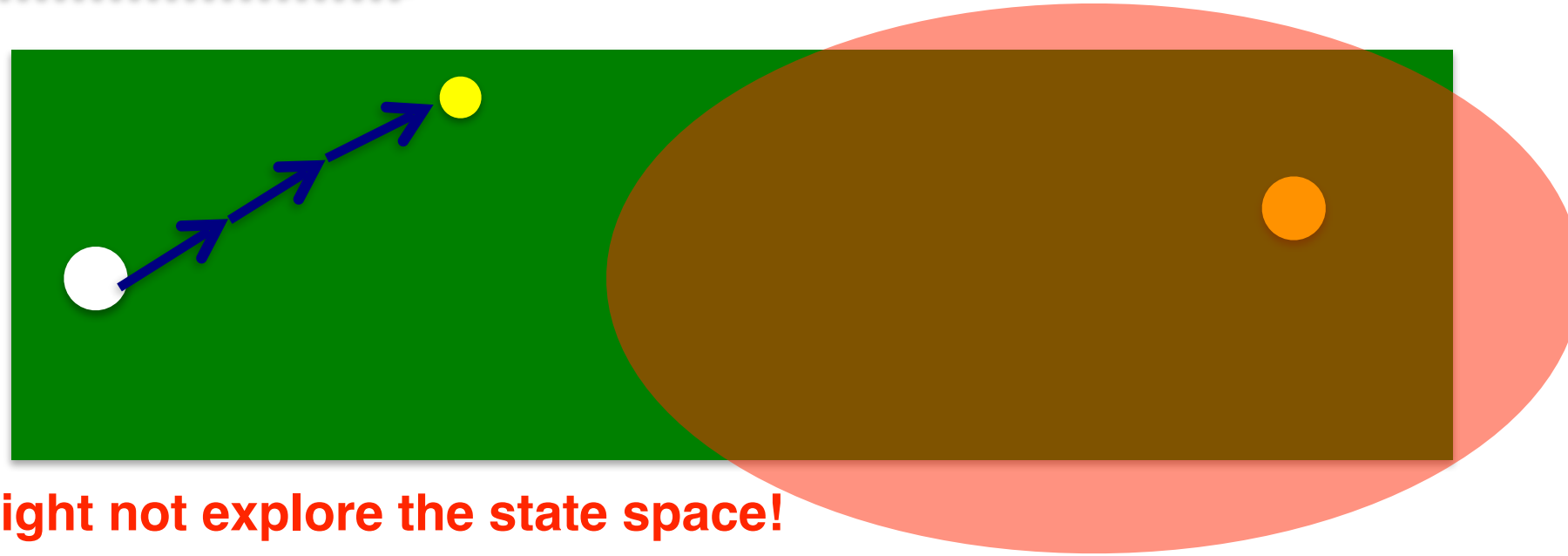




Is exploration a problem?

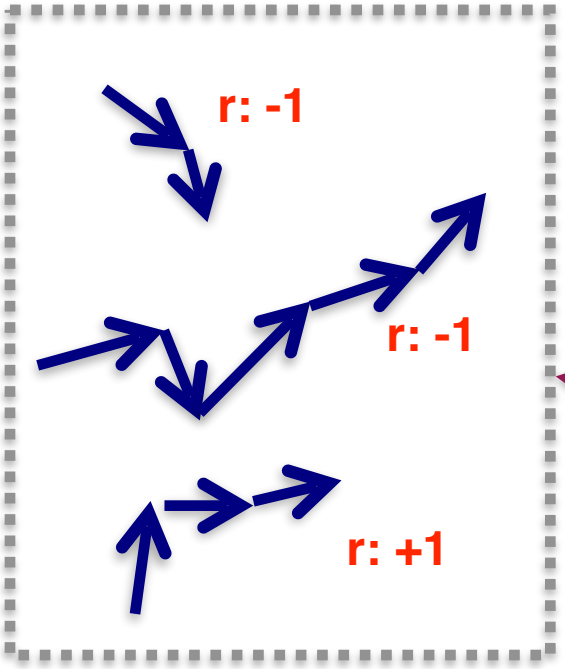
Yes!
(Can learn sub-optimal behavior)

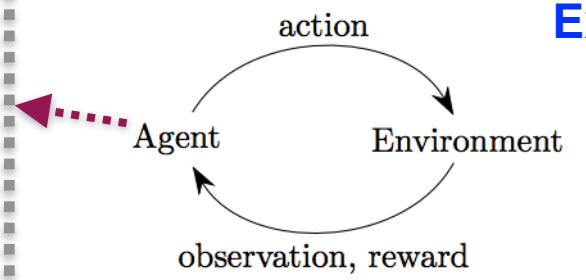
Exploration-Exploitation Dilemma



Might not explore the state space!

Dataset





Is exploration a problem?

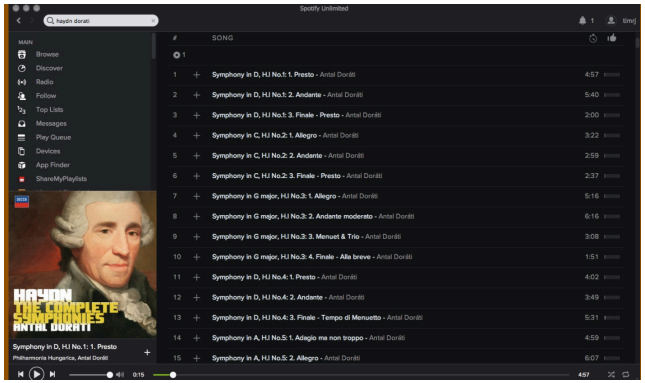
Yes!
(Can learn sub-optimal behavior)

Exploration-Exploitation Dilemma

Exploration can be quite hard!

When?

Imagine your favorite playlist



(they want you hooked)

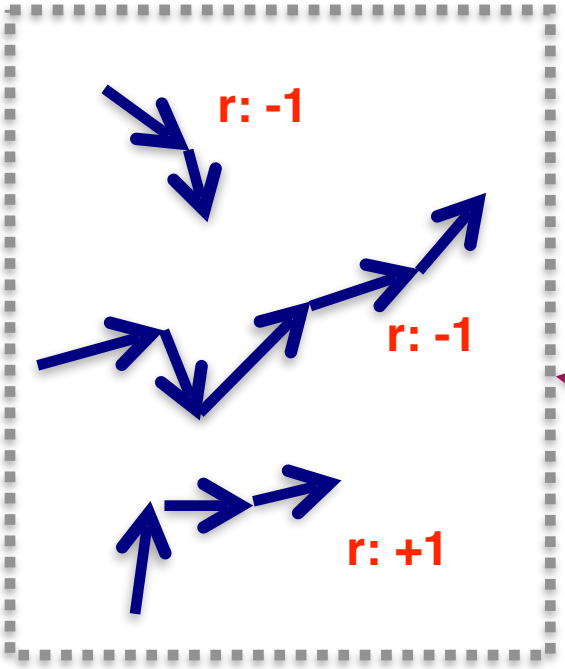
Explore by
Suggesting other music

Imagine your favorite playlist

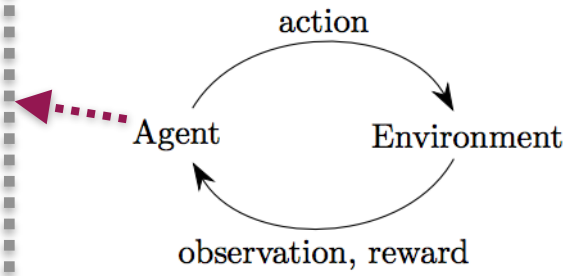
Sometimes Exploration can be very costly!

Suggesting other music

Dataset



What questions might be of interest?



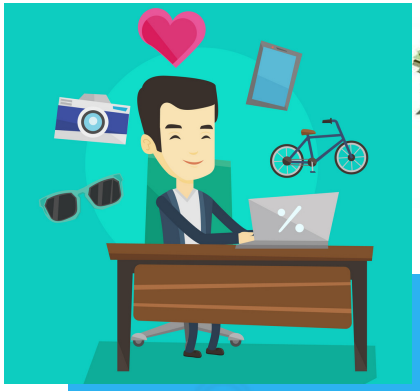
Is there a method that will achieve the highest reward?

How fast will reach it?

(i.e., what is the overall regret)

Consider landing pages for your website

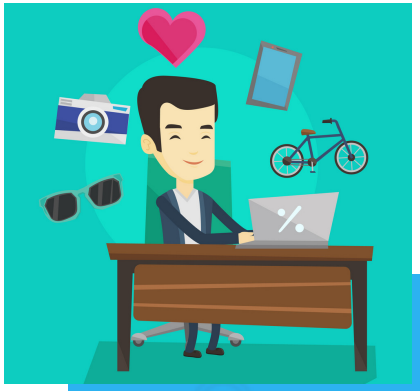
a_1 a_2 a_3



a_2

$r(a_2)$

a_1 a_2 a_3



a_2 a_3

$r(a_2)$ $r(a_3)$

a_1 a_2 a_3

Multi-Arm Bandit Problem

$$a_2, a_3, a_3, a_1, a_3, a_2, a_1, a_1, a_3 \ldots$$

$$r(a_2), r(a_3), r(a_3), r(a_1), r(a_3), r(a_2), r(a_1), r(a_1), r(a_3) \ldots$$

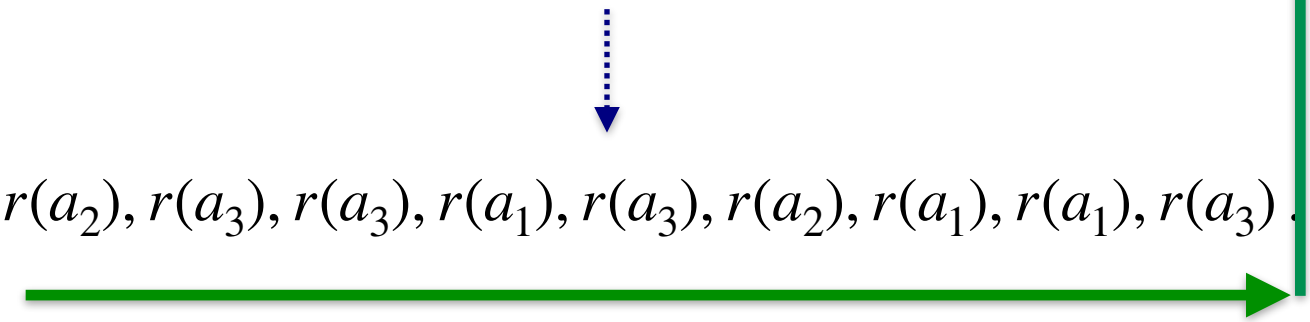
stochastic

time T

$$\sum_{t=1}^{T} r(a_i^t) \quad i \in [1,N]$$

Multi-Arm Bandit Problem

$$a_2, a_3, a_3, a_1, a_3, a_2, a_1, a_1, a_3 \ldots$$



$$r(a_2), r(a_3), r(a_3), r(a_1), r(a_3), r(a_2), r(a_1), r(a_1), r(a_3) \ldots$$

time T

Goal: maximize reward over time

$$\sum_{t=1}^{T} r(a_i^t) \quad i \in [1,N]$$

If rewards are deterministic can we find the best solution?

Return of i^{th} arm: $\mu_i = \frac{1}{k_i} \sum_{k_i} r(a_i)$

(i^{th} arm is pulled k_i times)

Which arm to choose?

$$a_2, a_3, a_3, a_1, a_3, a_2, a_1, a_1, a_3 \ldots$$

Explore phase

Exploit phase

$$r(a_2), r(a_3), r(a_3), r(a_1), r(a_3), r(a_2), r(a_1), r(a_1), r(a_3) \ldots$$

time K

time T

$$\sum_{t=1}^{T} r(a_i^t) \quad i \in [1,N]$$

(total reward)

$$\mu_i = \frac{1}{k_i} \sum_{k_i} r(a_i)$$

(mean reward of action i)

One Strategy: Explore-First

Sample each arm equally $\approx \frac{K}{N}$

After K rounds, choose arm with highest average reward μ_i

Only take the highest rewarding action for remaining T-K rounds

Is this the best we can do?

(slide co-designed with Cathy Wu)

What do we mean by best?

$$a_2, a_3, a_3, a_1, a_3, a_2, a_1, a_1, a_3 \ldots$$

$$r(a_2), r(a_3), r(a_3), r(a_1), r(a_3), r(a_2), r(a_1), r(a_1), r(a_3) \ldots$$

(total reward of selected actions)

$$R = \sum_{t=1}^{T} r(a_i^t)$$

(total reward of best actions)

$$R^* = \sum_{t=1}^{T} r(a_i^{t*})$$

oracle

time T

Why is this called oracle?

regret $\|R^* - R\|$

As in life, goal is to minimize regret

Assume $r \in [0, 1]$

Worst that we can do: T

Explore-First: $T^{2/3} \times O(N \log T)^{1/3}$

Not just the asymptotic performance, but how fast we reach!

(slide co-designed with Cathy Wu)

What do we mean by best?

$$a_2, a_3, a_3, a_1, a_3, a_2, a_1, a_1, a_3 \ldots$$

$$r(a_2), r(a_3), r(a_3), r(a_1), r(a_3), r(a_2), r(a_1), r(a_1), r(a_3) \ldots$$

Does there exist an optimal algorithm?

time T

$$R = \sum_{t=1}^{T} r(a_i^t) \qquad R^* = \sum_{t=1}^{T} r(a_i^{t^*}) \quad \text{oracle}$$

regret $\|R^* - R\|$

Not just the asymptotic performance, but how fast we reach!

What's the difference from supervised learning?

Online Decisions!

(i.e., can't wait to collect data first)

Profit per Click

significant difference!

Algo 1

Algo 2

data / time

What do we mean by best?

$$a_2, a_3, a_3, a_1, a_3, a_2, a_1, a_1, a_3 \ldots$$

$$r(a_2), r(a_3), r(a_3), r(a_1), r(a_3), r(a_2), r(a_1), r(a_1), r(a_3) \ldots$$

time T

Does there exist an optimal algorithm?

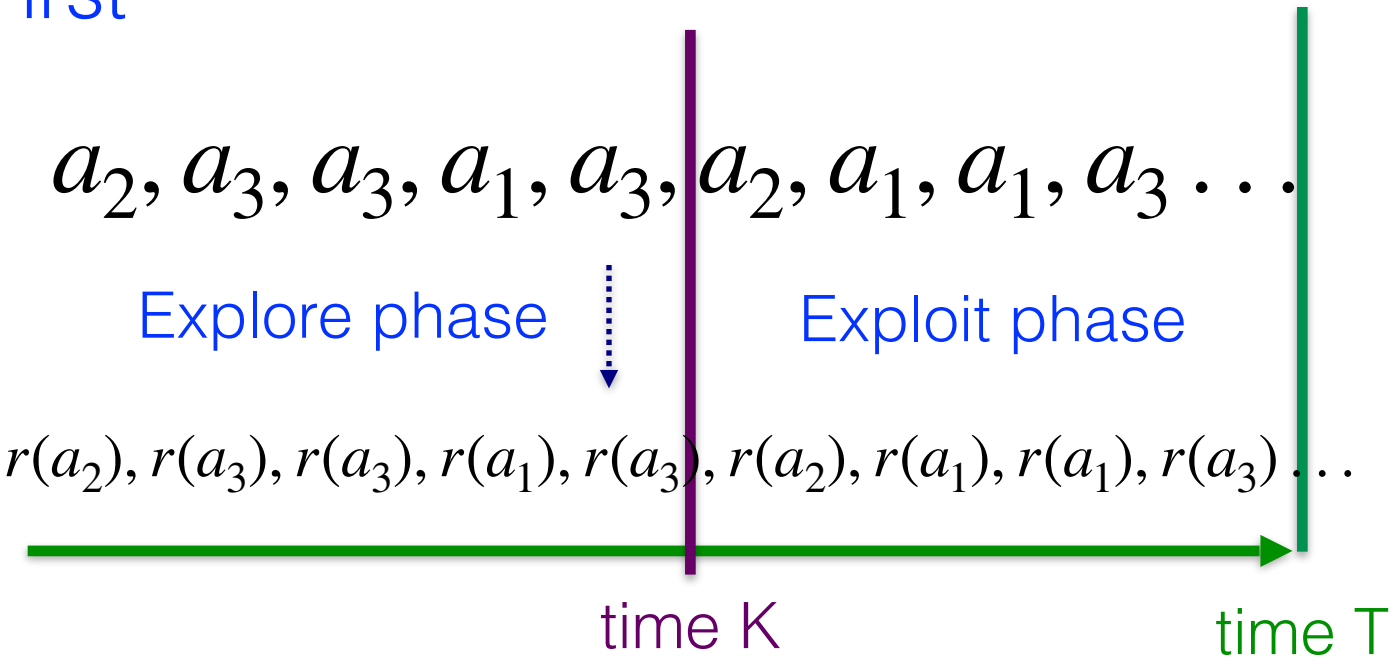
Upper Confidence Bound (UCB) Algorithm

$$R = \sum_{t=1}^{T} r(a_i) \qquad R^* = \sum_{t=1}^{T} r(a_i^*) \quad \text{oracle}$$

regret $\|R^* - R\|$

Not just the asymptotic performance, but how fast we reach!

Explore-First



Non-adaptive exploration

Explore + exploit separately

vs

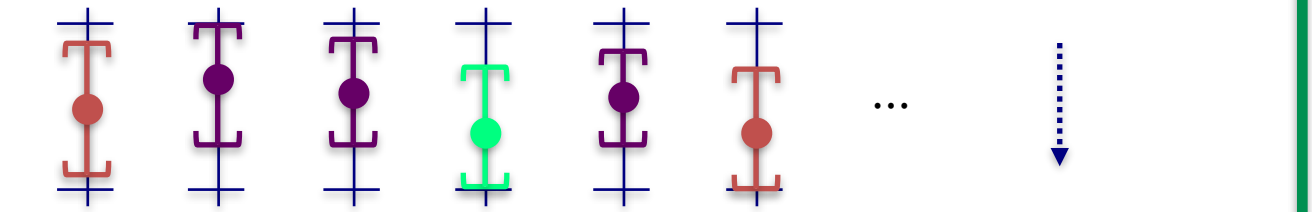
Adaptive exploration

Explore + Exploit simultaneously

(slide co-designed with Cathy Wu)

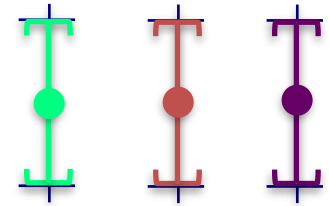
Upper Confidence Bound (UCB) Algorithm

$$a_2, a_3, a_3, a_1, a_3, a_2, a_1, a_1, a_3 \ldots$$



$$r(a_2), r(a_3), r(a_3), r(a_1), r(a_3), r(a_2), r(a_1), r(a_1), r(a_3) \ldots$$

Initial confidence intervals:



$a_1 \quad a_2 \quad a_3$

time T

Exploitation

$$a_{t+1} = \arg\max_i \mu_i(t) + \sqrt{\frac{4 \log t}{k_i}}$$

$$\mu_i = \frac{1}{k_i} \sum_{k_i} r(a_i)$$

(mean reward of action i)

Optimism in face of uncertainty

Exploration bonus for rare actions (optimism)

(slide co-designed with Cathy Wu)

How good is UCB?

$$a_2, a_3, a_3, a_1, a_3, a_2, a_1, a_1, a_3 \ldots$$

$$r(a_2), r(a_3), r(a_3), r(a_1), r(a_3), r(a_2), r(a_1), r(a_1), r(a_3) \ldots$$

(total reward of selected actions)

(total reward of best actions)

time T

$$R = \sum_{t=1}^{T} r(a_i^t)$$

$$R^* = \sum_{t=1}^{T} r(a_i^{t*})$$ oracle

regret $\|R^* - R\|$

As in life, goal is to minimize regret

Assume $r \in [0, 1]$

Worst that we can do: T

Explore-First: $T^{2/3} \times (N \log T)^{1/3}$

UCB: $(NT \log T)^{1/2}$

Optimal! (up to log factors)

(slide co-designed with Cathy Wu)

Upper Confidence Bound (UCB) Algorithm

$$\arg\max_i \hat{\mu}_i(t) + \sqrt{\frac{4 \log t}{k_i}}$$

$$\hat{\mu}_i = \frac{1}{k_i} \sum_{k_i} r(a_i)$$

Where this come from?

Empirical estimate of μ_i (unknown)

We want

$$\arg\max_i \mu_i$$

We have

$$\arg\max_i \hat{\mu}_i$$

Initially (with few samples)
This estimate is going to be bad

Construct

$$\arg\max_i \hat{\mu}'_i$$

Principle of optimism: find $\hat{\mu}'_i \geq \mu_i$

$$p\left(\mu_i \geq \hat{\mu}'_i\right) \leq \delta$$

Is true!

If $r(a_i)$ are 1-subgaussian and if $\hat{\mu}'_i : \hat{\mu}_i + \sqrt{\frac{2 \log \frac{1}{\delta}}{k_i}}$

Upper Confidence Bound (UCB) Algorithm

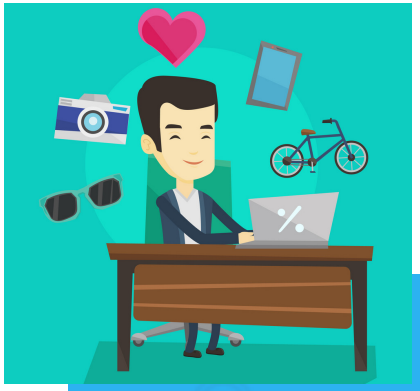
$$\arg\max_i \mu_i(t) + \sqrt{\frac{4 \log t}{k_i}} \qquad \mu_i = \frac{1}{k_i} \sum_{k_i} r(a_i)$$

Upper Bound on average number of sub-optimal actions

$$\frac{16 |A| \log T}{\Delta^2} + O(1)$$

$$\Delta = \mu_{best} - \mu_{second\_best}$$

$|A|$: number of actions



a_2 a_3

$r(a_2)$ $r(a_3)$

a_1 a_2 a_3

a_2 a_3

$r(a_2)$ $r(a_3)$

(female, 20s, computer-savvy)

(male, 30s, computer-savvy)

How to use these "features" in decision making?

Contextual Bandits

a_1 a_2 a_3

a_2 a_3

$r(a_2)$ $r(a_3)$

$r(a_2, s_2)$ $r(a_3, s_3)$

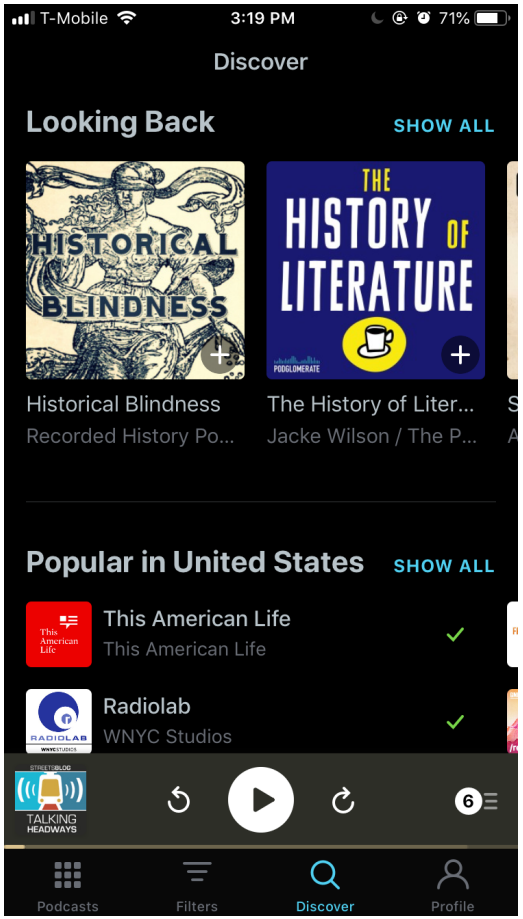
s_2

s_3

(male, 30s, computer-savvy)

(female, 20s, computer-savvy)

Example:
Podcast
recommendations



(slide co-designed with Cathy Wu)

a_2 a_3

$r(a_2)$ $r(a_3)$

$r(a_2, s_2)$ $r(a_3, s_3)$

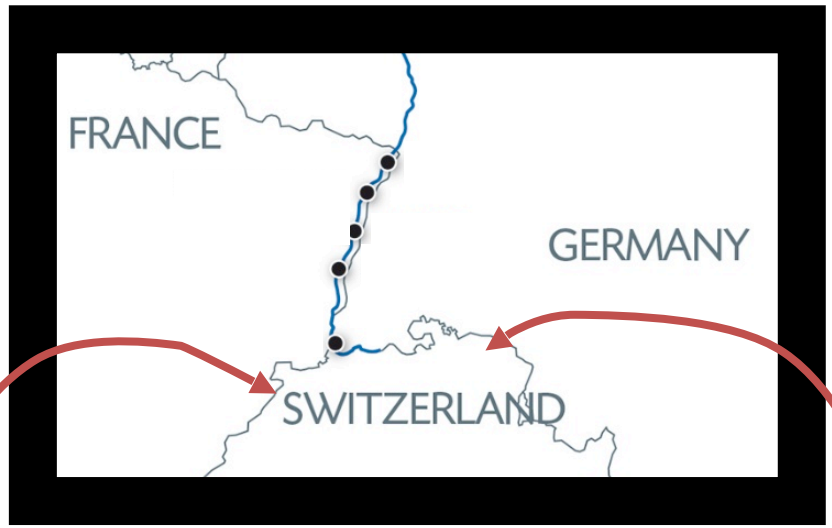
s_2 s_3

(male, 30s, computer-savvy)

(female, 20s, computer-savvy)

Example types of contexts

- User demographics
  - Discrete vs continuous



Switzerland

vs

(lat, long) coordinates

Wine or beer podcast?

vs

(slide co-designed with Cathy Wu)

a_2 a_3

$r(a_2)$ $r(a_3)$

$r(a_2, s_2)$ $r(a_3, s_3)$

s_3

s_2

(female, 20s, computer-savvy)

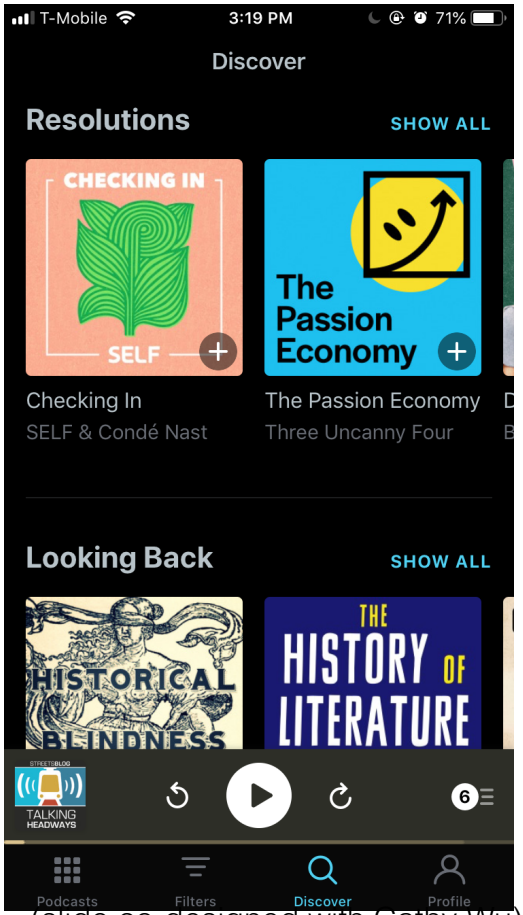
(male, 30s, computer-savvy)

Example types of contexts

- User demographics
- Time of day/year
- User mental state

Happy 2021!

New years resolutions?



(slide co-designed with Cathy Wu)

a_2 a_3

$r(a_2)$ $r(a_3)$

$r(a_2, s_2)$ $r(a_3, s_3)$

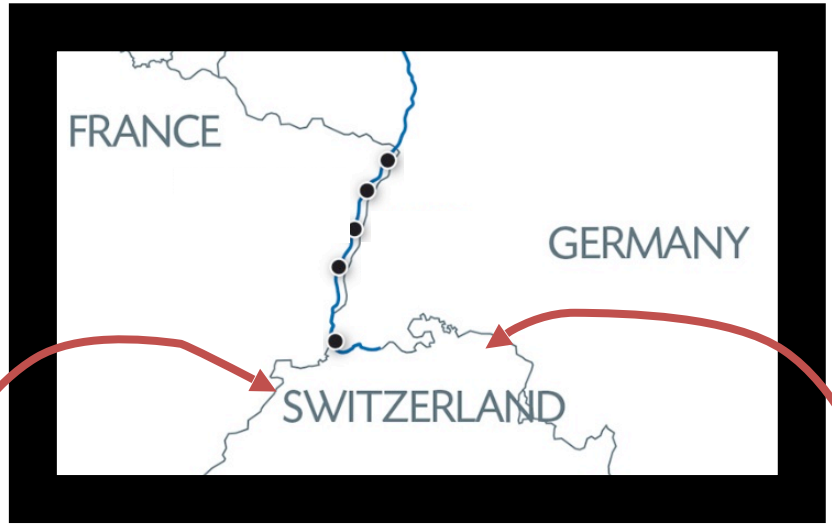
s_2 s_3

(male, 30s, computer-savvy)

(female, 20s, computer-savvy)

Naïve approach:
independent bandit problems
(one for each context)

Challenge:
may not handle continuous contexts well



Wine or beer podcast?

VS

(slide co-designed with Cathy Wu)
(slide co-designed with Cathy Wu)

a_2 a_3

$r(a_2)$ $r(a_3)$

$r(a_2, s_2)$ $r(a_3, s_3)$

s_2

(male, 30s, computer-savvy)

s_3

(female, 20s, computer-savvy)

Better approach:

LinUCB

Assume:
expected rewards are linear in context

$$\mu(a|s) = s_a \cdot \theta_a$$

θ_a fixed but unknown

Strategy:
select arm with highest

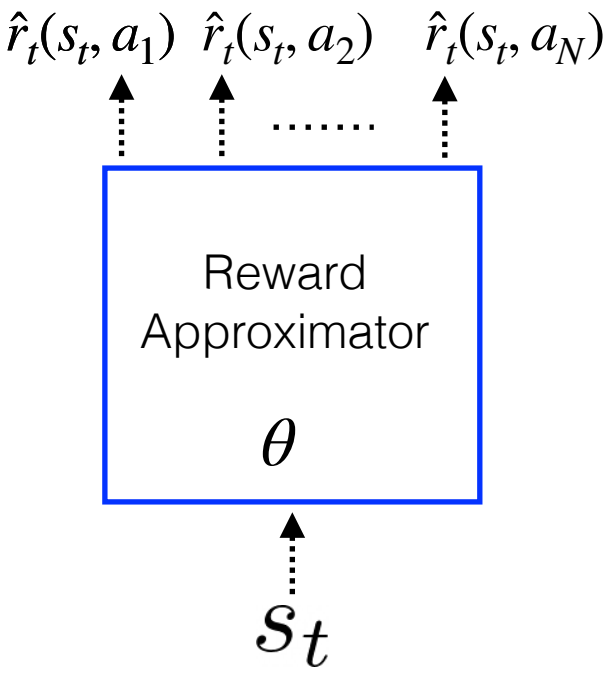
$$\mathrm{UCB}_t(a|s_t) = \max_{\theta \in C_t} s_{t,a} \cdot \theta_a$$

Confidence interval
(shrinks with each visit)

Optimal! (up to log factors)

(slide co-designed with Cathy Wu)

Expanding on LinUCB



Estimate reward for each action

$$\hat{r}_t^a = s_t \theta_t^a$$

Choose the best one (or sample)

$$a_t \leftarrow \max_a \hat{r}_t^a$$

All time steps until t

$$\hat{R}_{0:t}^a = S_{0:t} \theta^a$$

Solve for the parameters

$$\min_{\theta_a} \| R_{0:t}^a - \hat{R}_{0:t}^a \|_2^2$$

Ridge Regression

$$\theta_t^a = \left(S_{0:t}^T S_{0:t} + \lambda I \right)^{-1} S_{0:t}^T R_{0:t}^a$$

$$\theta_t^a = \underbrace{\left(X_{0:t}^T X_{0:t} + \lambda I \right)^{-1}}_{A_a} \underbrace{X_{0:t}^T R_{0:t}^a}_{b_a}$$

(X: features of S)

Need to solve Online!

$$\theta_t^a = \underbrace{\left(X_{0:t}^T X_{0:t} + \lambda I\right)^{-1}}_{A_a} \underbrace{X_{0:t}^T R_{0:t}^a}_{b_a}$$

Algorithm 1 LinUCB with disjoint linear models.

0: Inputs: $\alpha \in \mathbb{R}_+$
1: **for** $t = 1, 2, 3, \ldots, T$ **do**
2: Observe features of all arms $a \in \mathcal{A}_t$: $\mathbf{x}_{t,a} \in \mathbb{R}^d$
3: **for all** $a \in \mathcal{A}_t$ **do**
4:
5:
6:
7:
8: $\boldsymbol{\theta}_a \leftarrow \mathbf{A}_a^{-1}\mathbf{b}_a$
9:
10: **end for**
11:
12:
13:
14: **end for**

Exploration Bonus

Online Update

A Contextual-Bandit Approach to Personalized News Article Recommendation, Li et al. 2012

$$\theta_t^a = \underbrace{\left(X_{0:t}^T X_{0:t} + \lambda I\right)^{-1}}_{A_a} \underbrace{X_{0:t}^T R_{0:t}^a}_{b_a}$$

Algorithm 1 LinUCB with disjoint linear models.

0: Inputs: $\alpha \in \mathbb{R}_+$
1: **for** $t = 1, 2, 3, \ldots, T$ **do**
2: Observe features of all arms $a \in \mathcal{A}_t$: $\mathbf{x}_{t,a} \in \mathbb{R}^d$
3: **for all** $a \in \mathcal{A}_t$ **do**
4:
5:
6:
7:
8: $\boldsymbol{\theta}_a \leftarrow \mathbf{A}_a^{-1}\mathbf{b}_a$
9: $p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^\top \mathbf{x}_{t,a} + \alpha\sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1}\mathbf{x}_{t,a}}$ Exploration Bonus
10: **end for**
11: Choose arm $a_t = \arg\max_{a \in \mathcal{A}_t} p_{t,a}$ with ties broken arbitrarily, and observe a real-valued payoff r_t
12: $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_{t,a_t}\mathbf{x}_{t,a_t}^\top$
13: $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_t\mathbf{x}_{t,a_t}$ Online Update
14: **end for**

Pros and Cons of "disjoint models" (separate θ_a for each action) ?

A Contextual-Bandit Approach to Personalized News Article Recommendation, Li et al. 2012

$$\theta_t^a = \underbrace{\left(X_{0:t}^T X_{0:t} + \lambda I\right)^{-1}}_{A_a} \underbrace{X_{0:t}^T R_{0:t}^a}_{b_a}$$

Algorithm 1 LinUCB with disjoint linear models.

0: Inputs: $\alpha \in \mathbb{R}_+$
1: **for** $t = 1, 2, 3, \ldots, T$ **do**
2: Observe features of all arms $a \in \mathcal{A}_t$: $\mathbf{x}_{t,a} \in \mathbb{R}^d$
3: **for all** $a \in \mathcal{A}_t$ **do**
4:
5:
6:
7:
8: $\boldsymbol{\theta}_a \leftarrow \mathbf{A}_a^{-1}\mathbf{b}_a$
9: $p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^\top \mathbf{x}_{t,a} + \alpha\sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1}\mathbf{x}_{t,a}}$ Exploration Bonus
10: **end for**
11: Choose arm $a_t = \arg\max_{a \in \mathcal{A}_t} p_{t,a}$ with ties broken arbitrarily, and observe a real-valued payoff r_t
12: $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_{t,a_t}\mathbf{x}_{t,a_t}^\top$
13: $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_t\mathbf{x}_{t,a_t}$ Online Update
14: **end for**

What if there are new news articles?

A Contextual-Bandit Approach to Personalized News Article Recommendation, Li et al. 2012

$$\theta_t^a = \underbrace{\left(X_{0:t}^T X_{0:t} + \lambda I\right)^{-1}}_{A_a} \underbrace{X_{0:t}^T R_{0:t}^a}_{b_a}$$

Algorithm 1 LinUCB with disjoint linear models.

0: Inputs: $\alpha \in \mathbb{R}_+$
1: **for** $t = 1, 2, 3, \ldots, T$ **do**
2: Observe features of all arms $a \in \mathcal{A}_t$: $\mathbf{x}_{t,a} \in \mathbb{R}^d$
3: **for all** $a \in \mathcal{A}_t$ **do**
4: **if** a is new **then**
5: $\mathbf{A}_a \leftarrow \mathbf{I}_d$ (d-dimensional identity matrix)
6: $\mathbf{b}_a \leftarrow \mathbf{0}_{d \times 1}$ (d-dimensional zero vector)
7: **end if**
8: $\hat{\boldsymbol{\theta}}_a \leftarrow \mathbf{A}_a^{-1} \mathbf{b}_a$
9: $p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^\top \mathbf{x}_{t,a} + \alpha \sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1} \mathbf{x}_{t,a}}$ Exploration Bonus
10: **end for**
11: Choose arm $a_t = \arg\max_{a \in \mathcal{A}_t} p_{t,a}$ with ties broken arbitrarily, and observe a real-valued payoff r_t
12: $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_{t,a_t} \mathbf{x}_{t,a_t}^\top$
13: $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_t \mathbf{x}_{t,a_t}$ Online Update
14: **end for**

Context: News articles + User Features

A Contextual-Bandit Approach to Personalized News Article Recommendation, Li et al. 2012