Markov Decision Process

Sanjiban Choudhury

Cornell Bowers CIS
Computer Science

Announcements

1. Thanks for finishing Assignment 0!

2. Assignment 1 released!

3. Slides, Python notebook released

Learning

Robot

Decision

Making

Today!

Question from last class:

"Will we *only* look at discrete actions?"

Calculus to the rescue

Develop ideas in discrete space, extend to continuous space

Generalized Weighted Majority → Normalized Exponentiated Gradient Descent

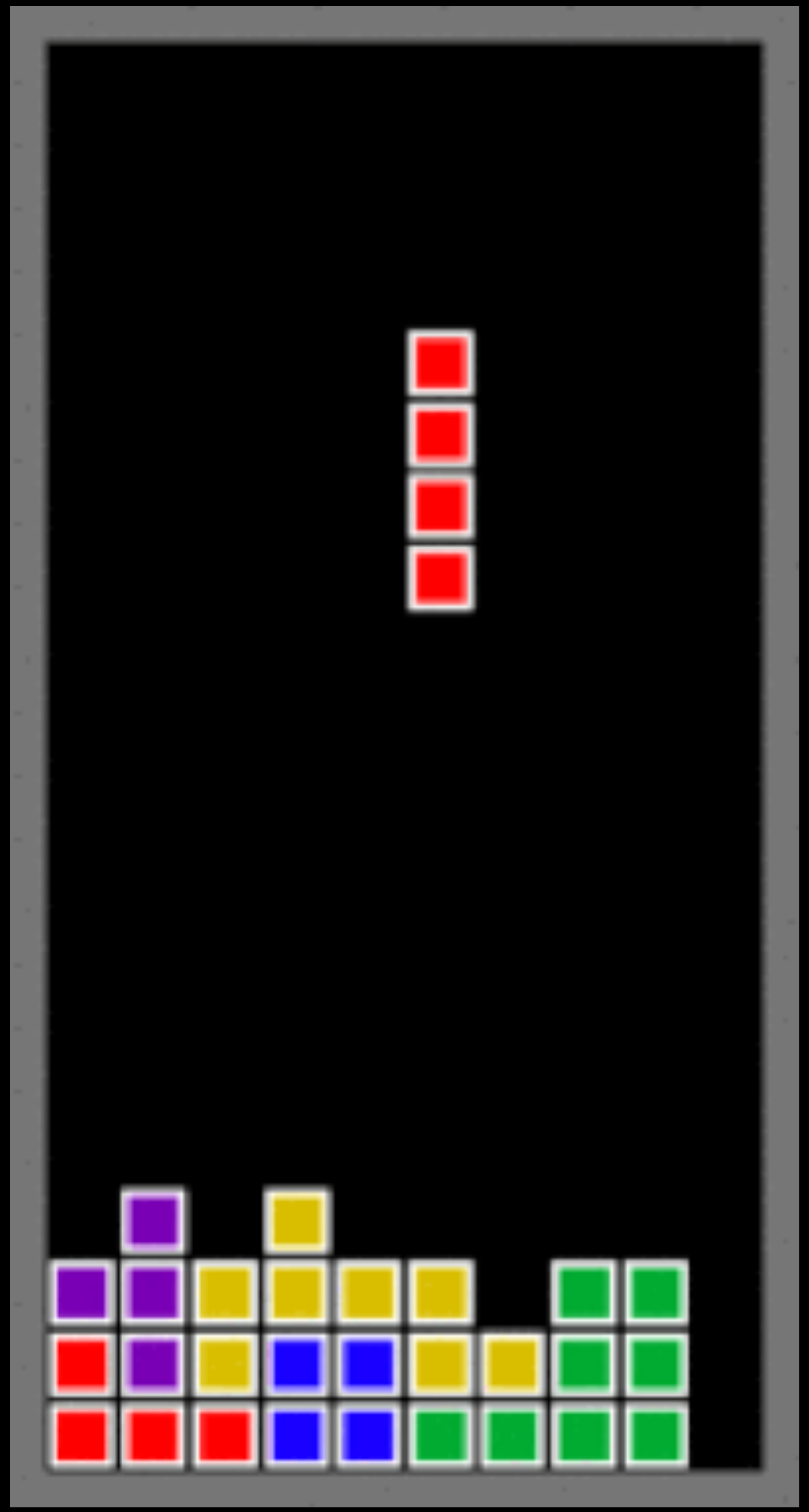
Discrete Value Iteration → Algebraic Ricatti Equations

Learning

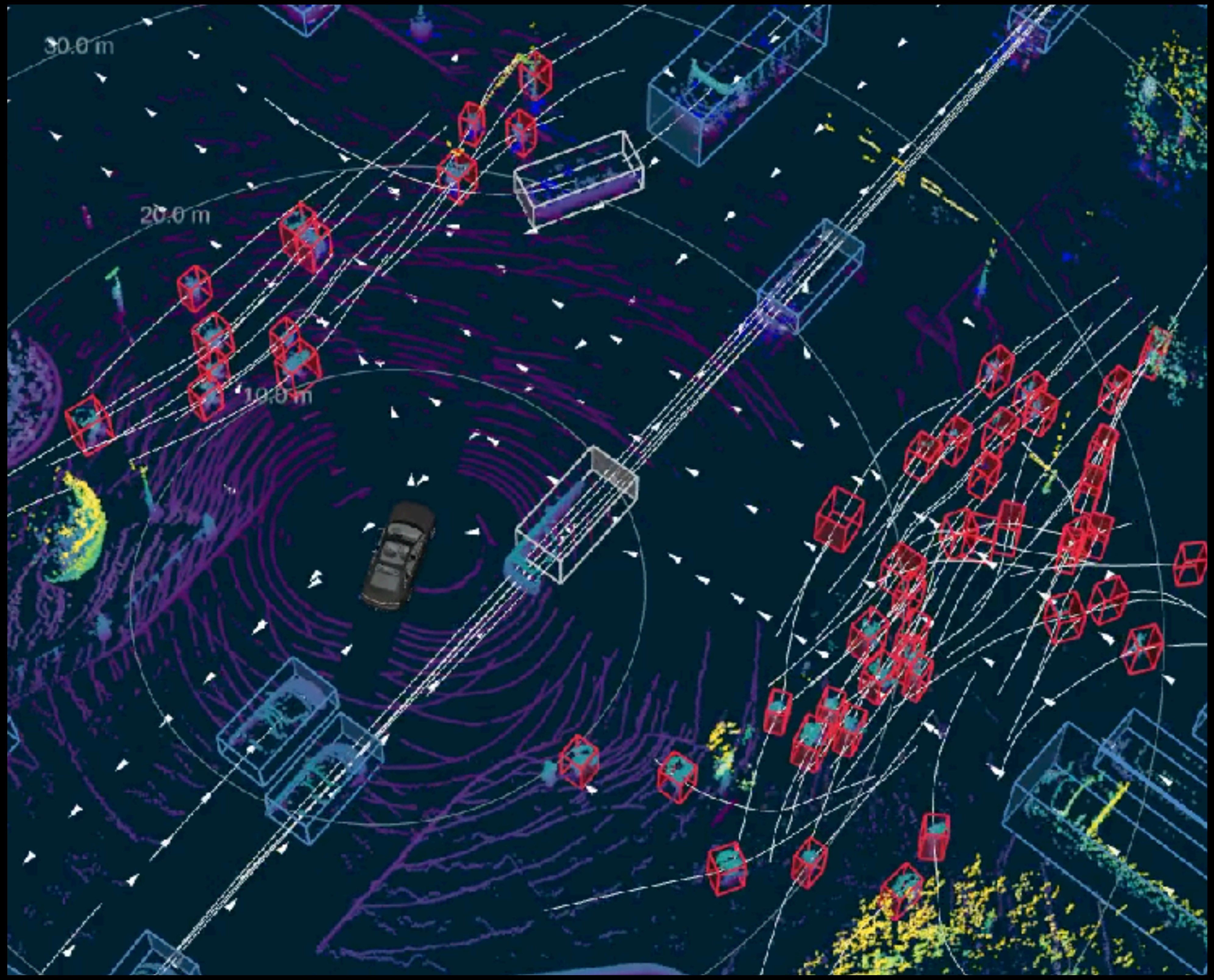
Robot
Decision
Making

Today!

Decisions, decisions!



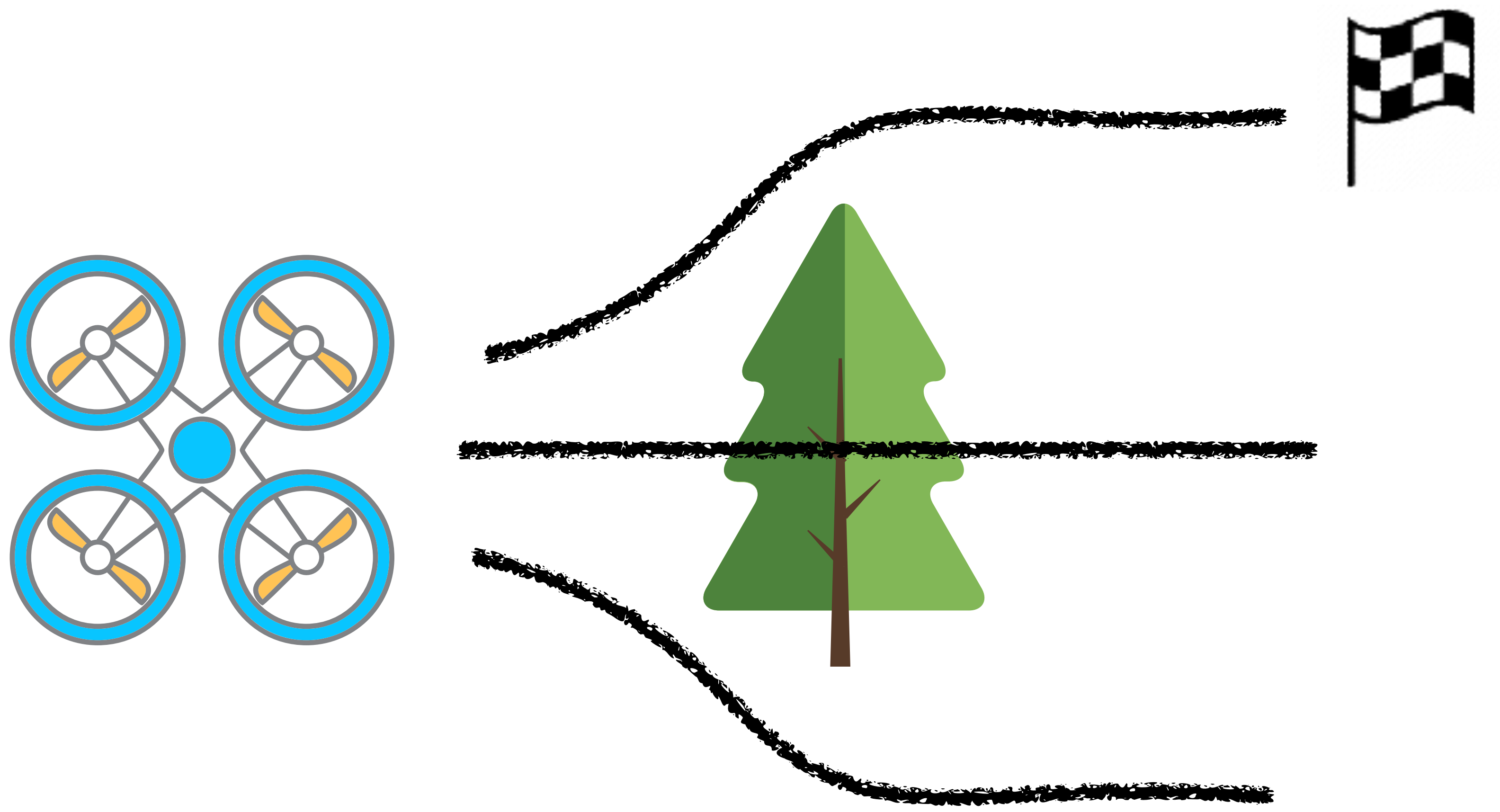
Tetris



Self-driving

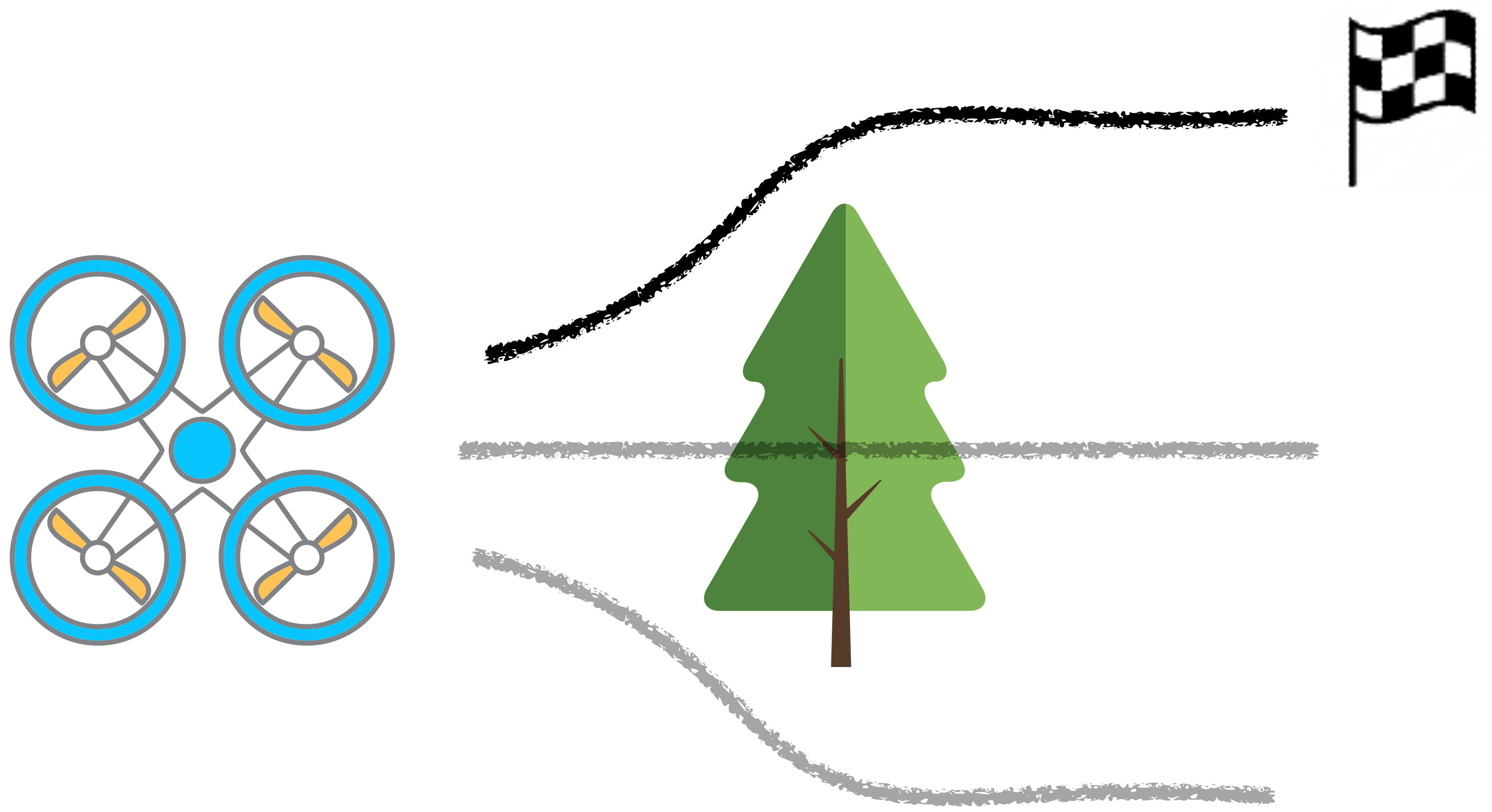
Robot Baristas

What makes decision making hard?



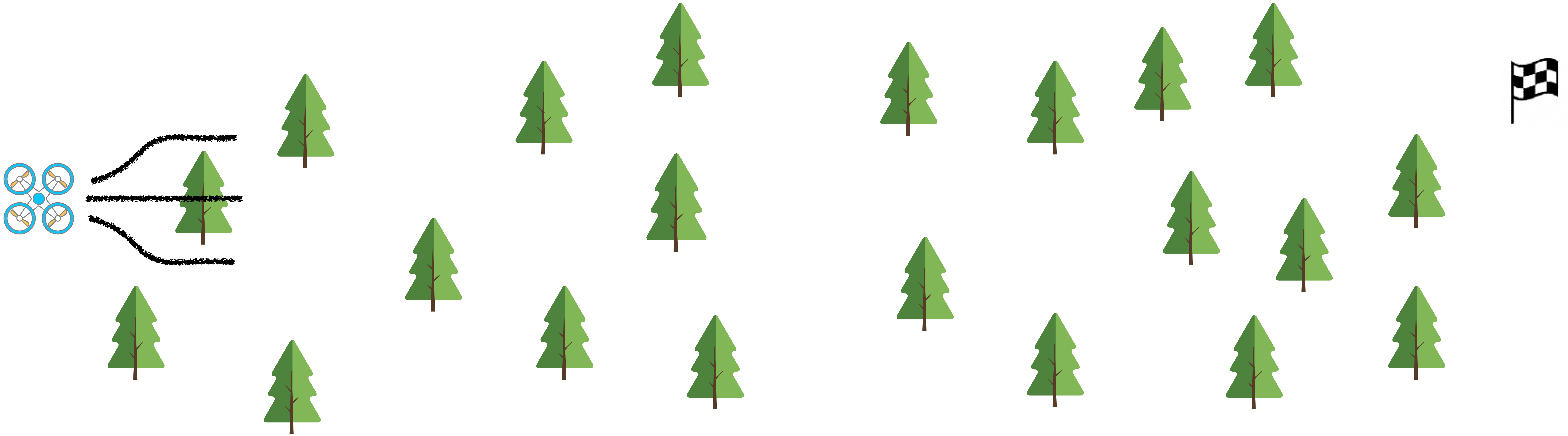
Single shot decision making

What makes decision making hard?



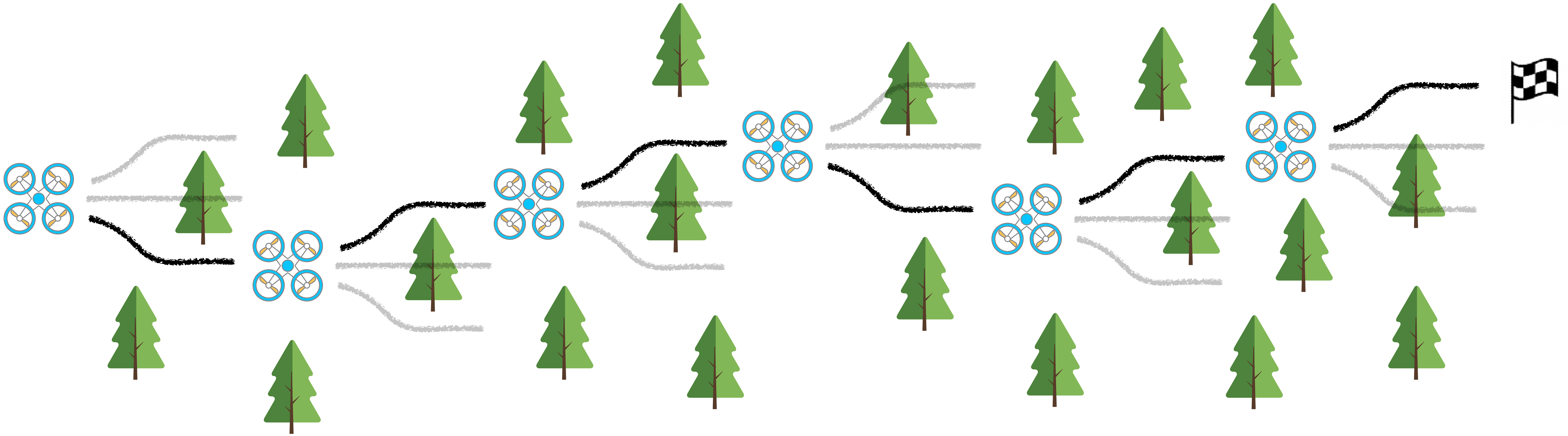
Single shot decision making

What makes decision making hard?



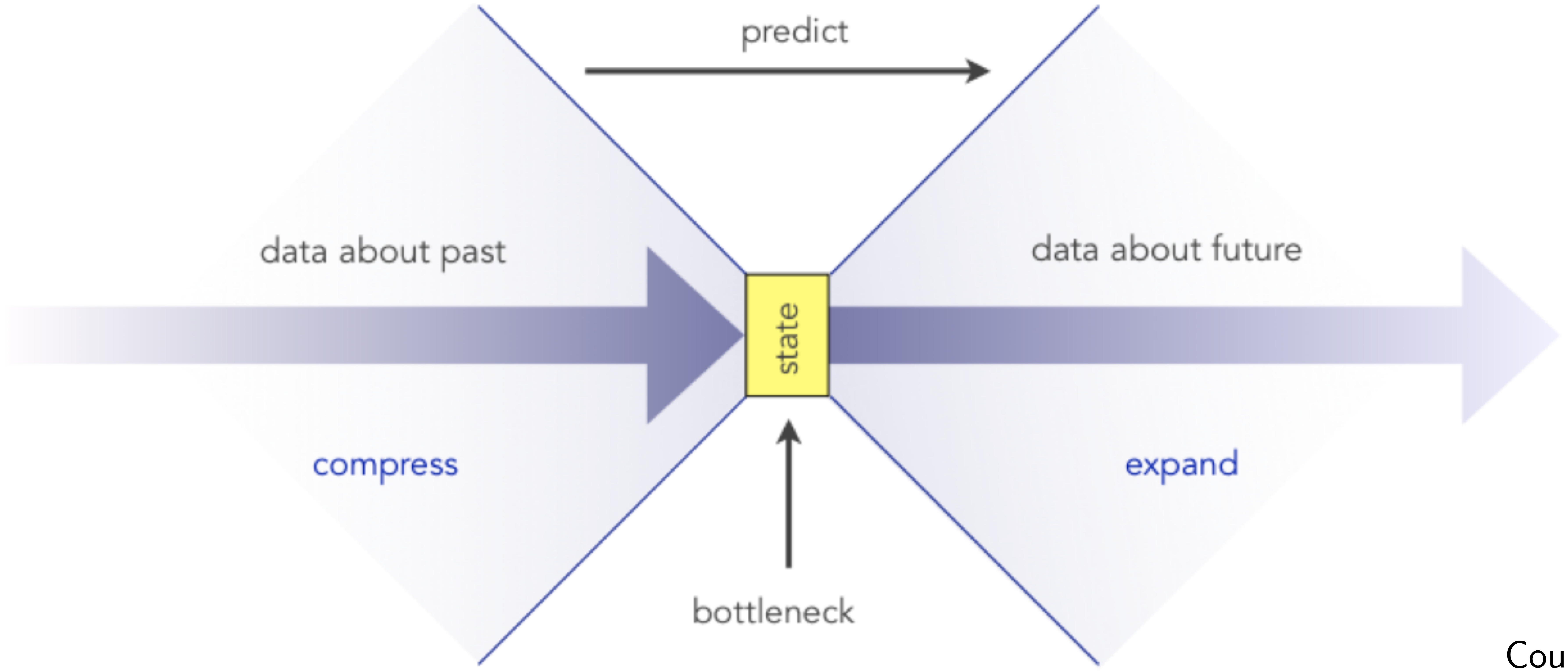
Sequential decision making

What makes decision making hard?



How do we tractably reason over a sequence of decisions?

Markov to the rescue!



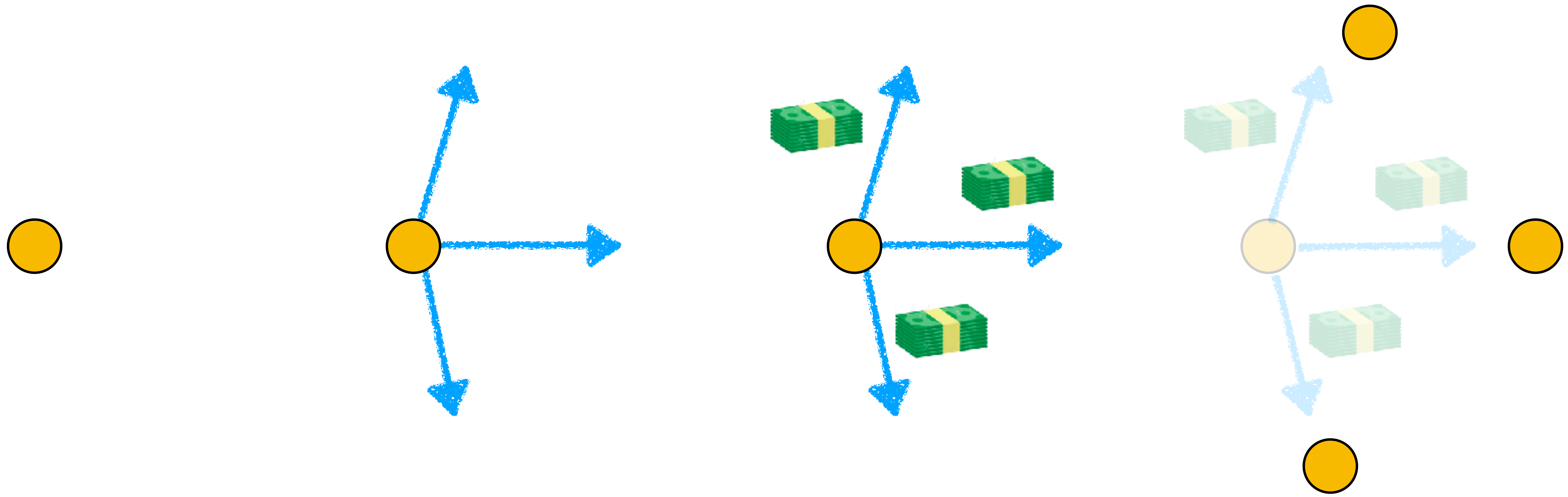
Courtesy: Byron Boots

State: statistic of history sufficient to predict the future

Markov Decision Process

A mathematical framework for modeling sequential decision making

$$< S \, , A \, , C \, , \mathcal{T} >$$

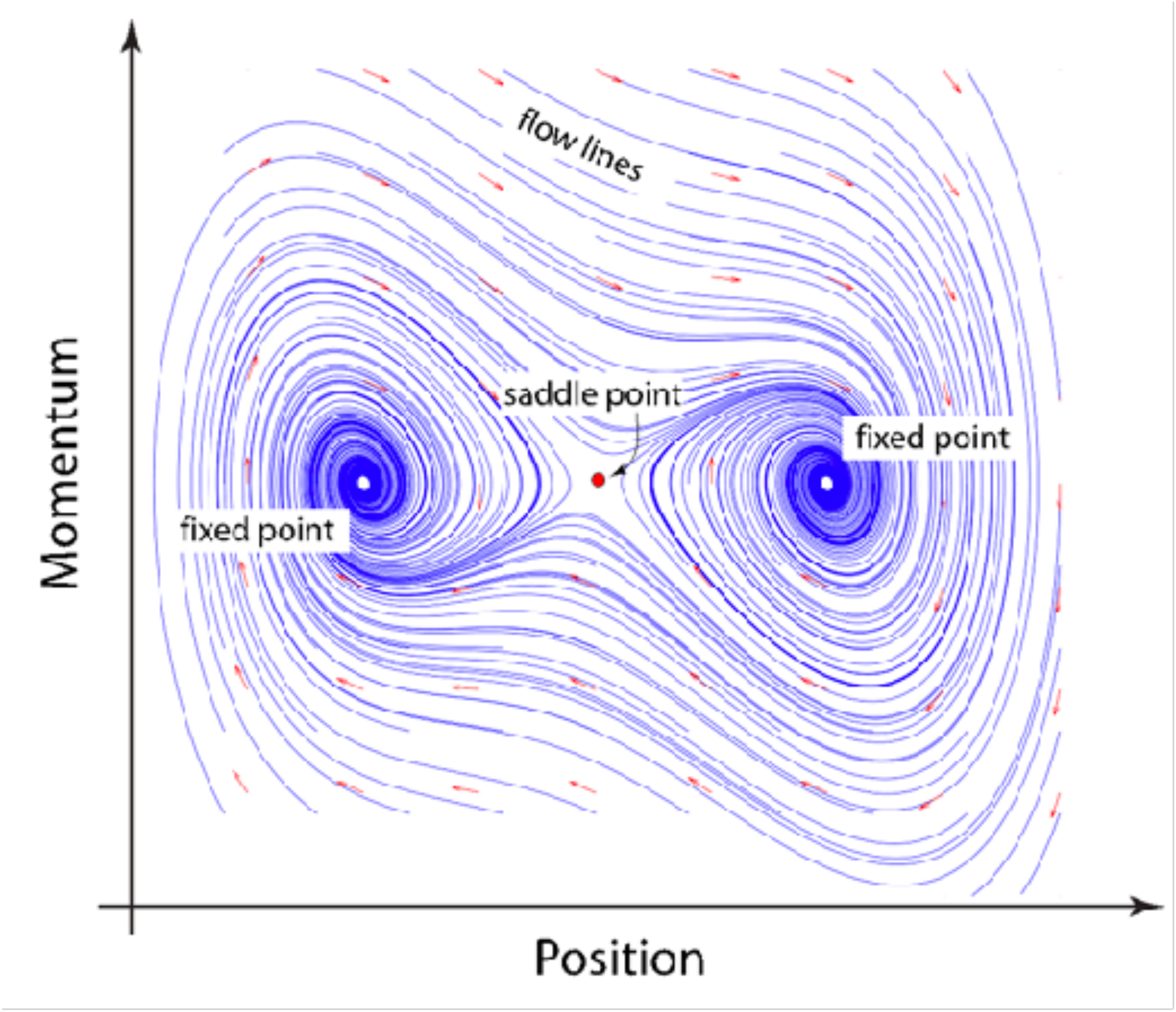


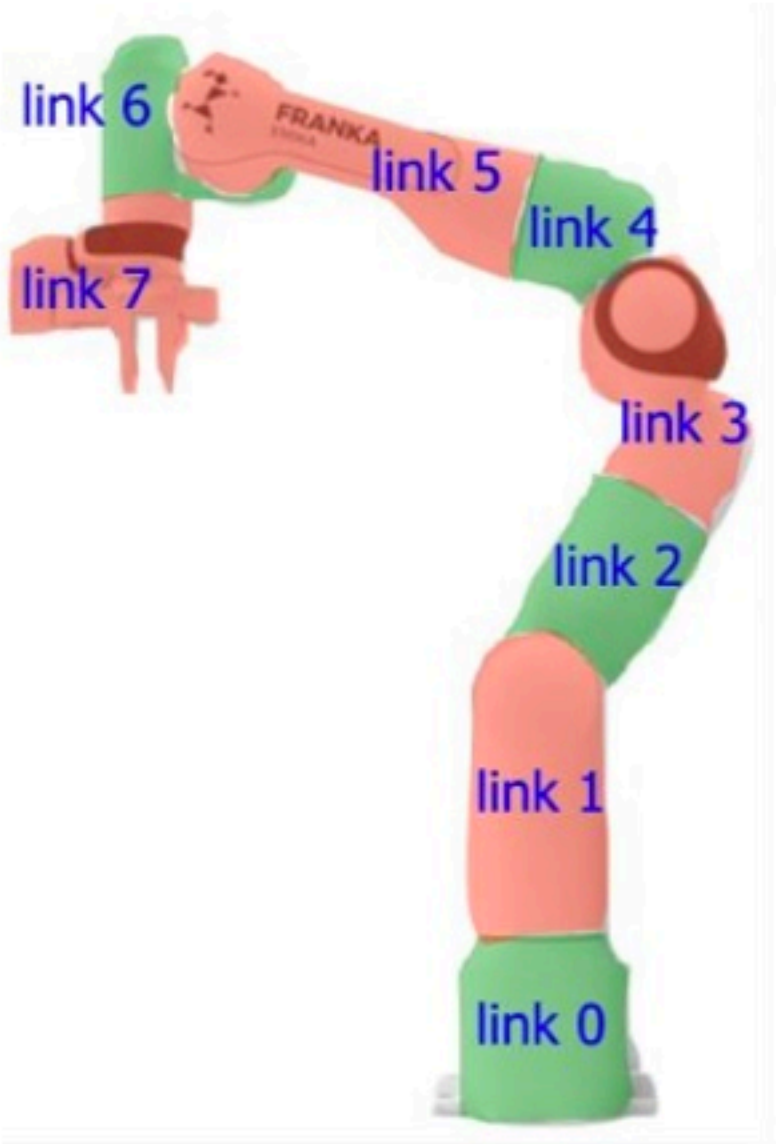
State

$< S , A , C , \mathcal{T} >$

Sufficient statistic of the system to predict future disregarding the past

- $s \in S$





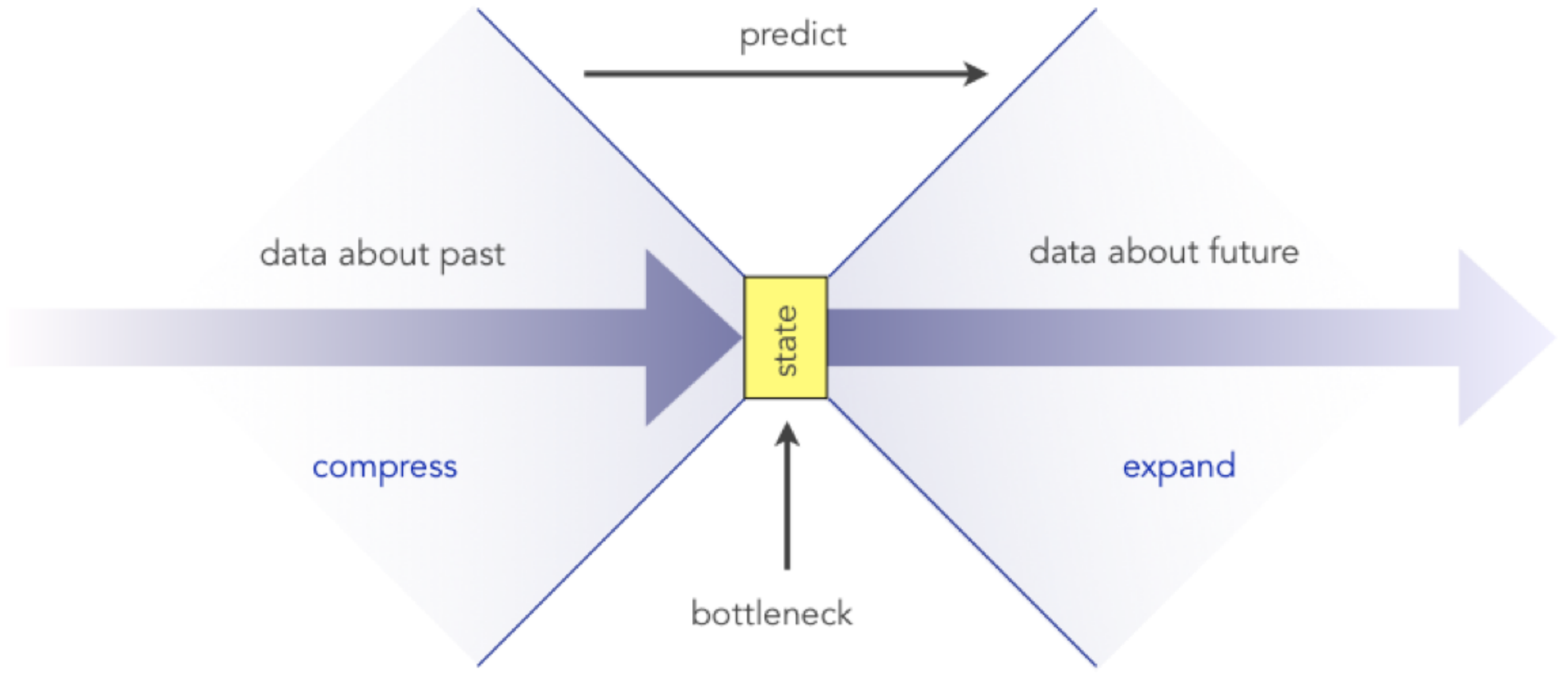
Activity!

Think-Pair-Share

Think (30 sec): Example of MDPs with shallow state?
(Current observation good enough)
Example of MDPs with deep state?

Pair: Find a partner

Share (45 sec): Partners exchange ideas

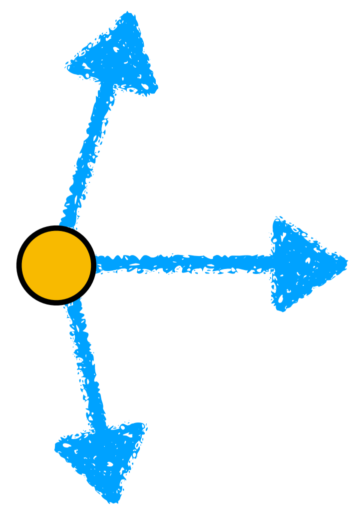


State: statistic of history sufficient to predict the future

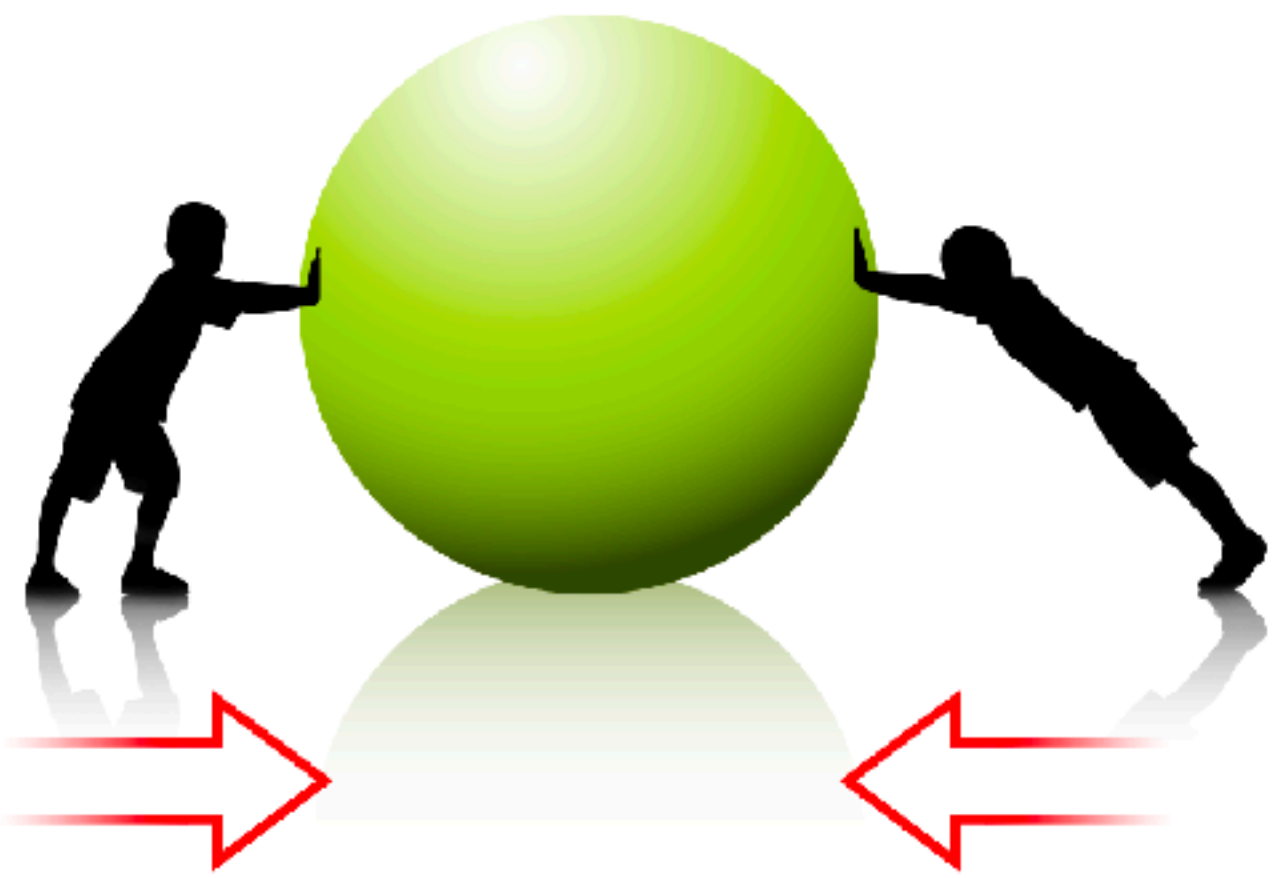
Action

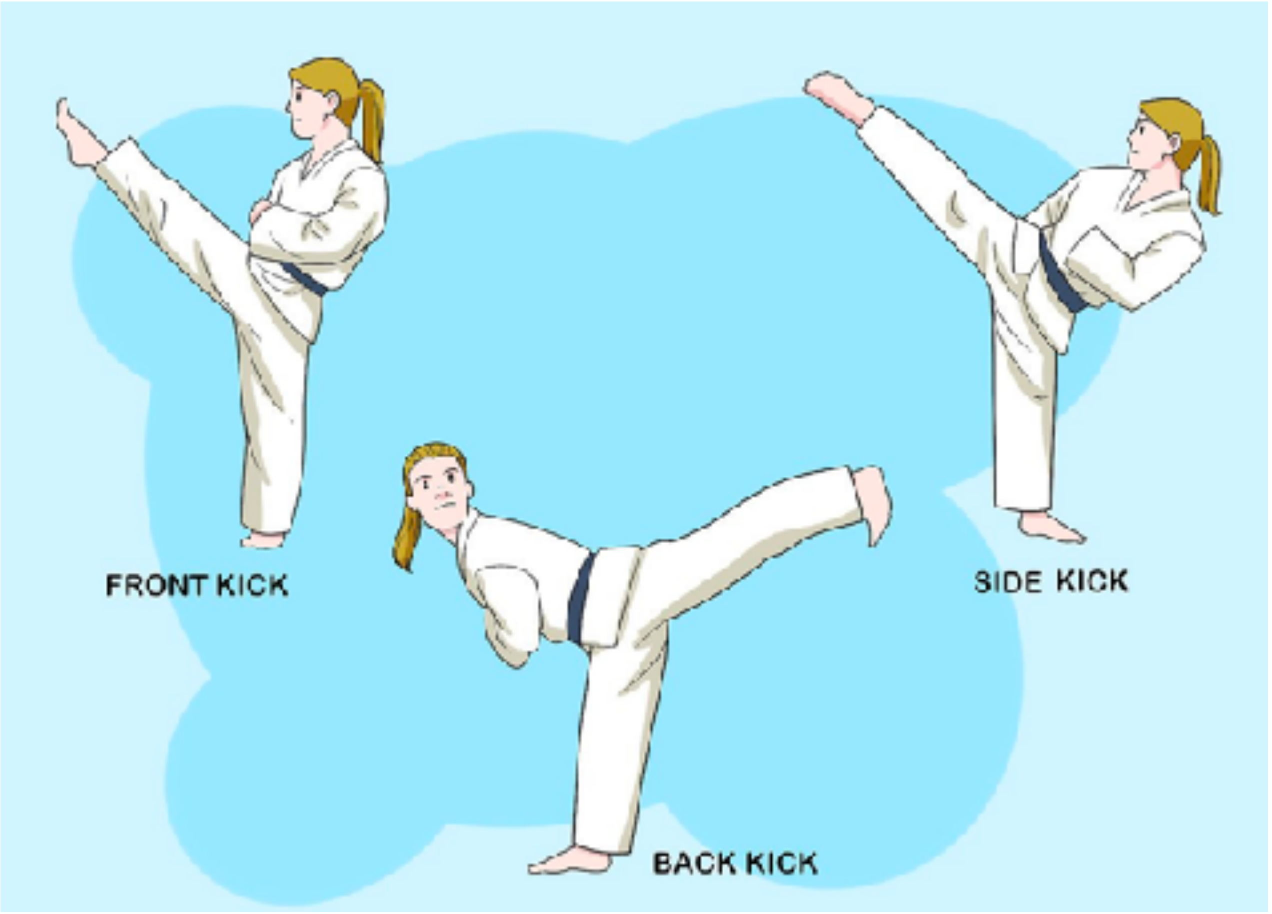
$< S , A , C , \mathcal{T} >$

Doing something:
Control action / decisions



$a \in A$



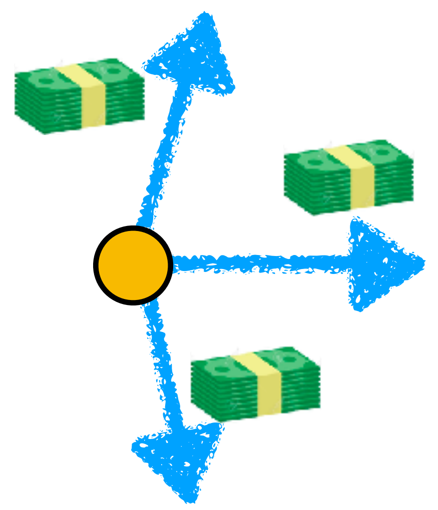


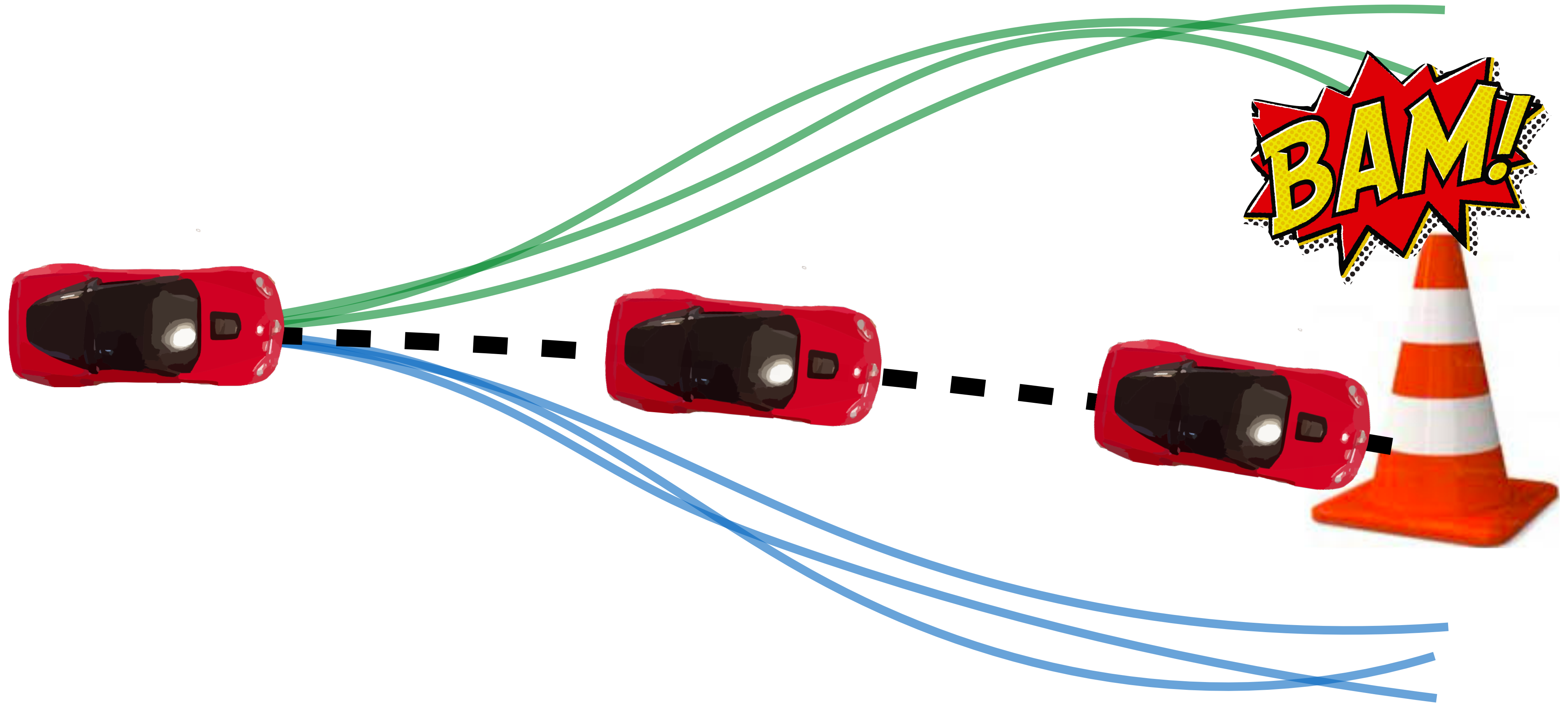
Cost

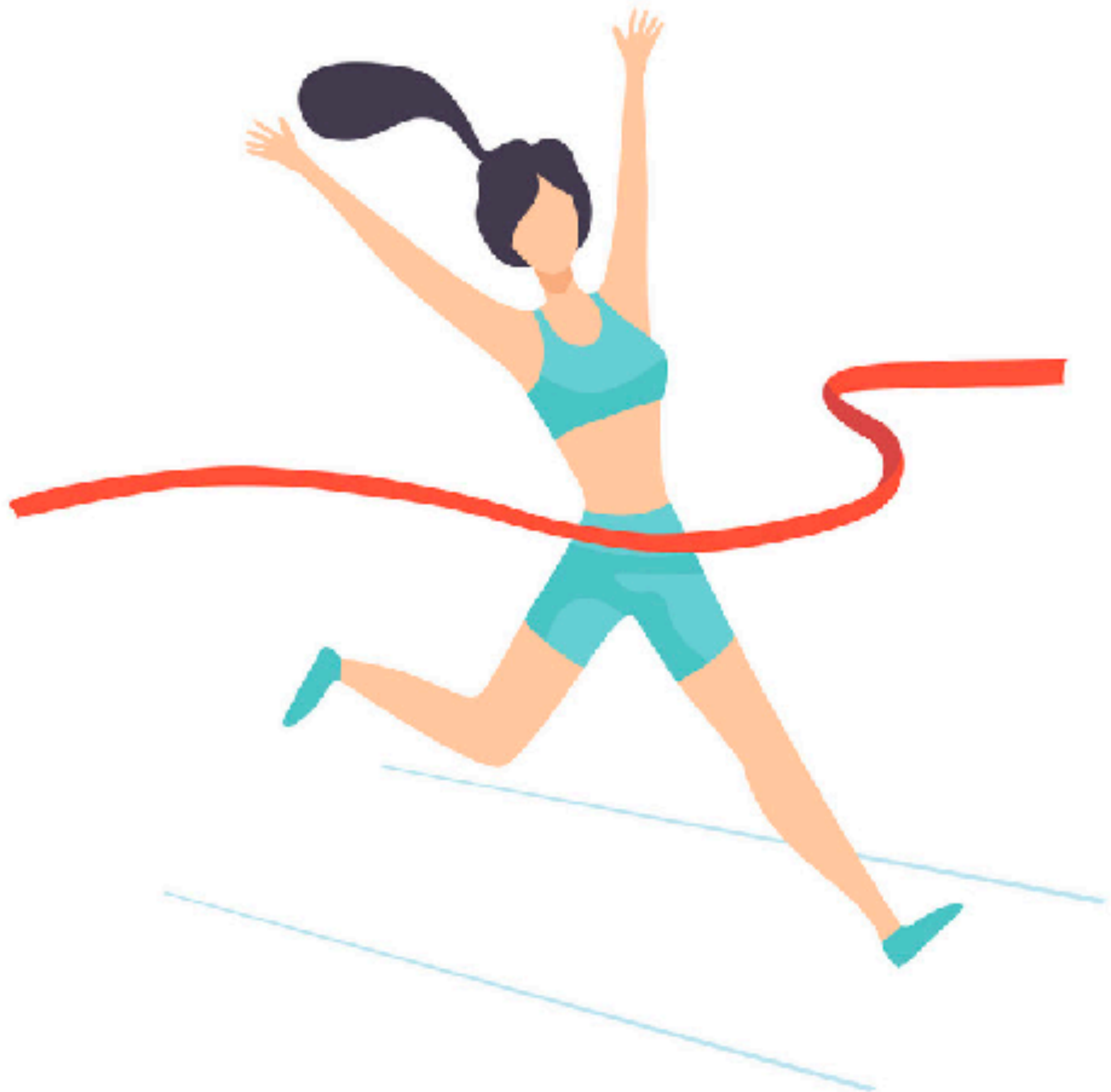
$< S, A, C, \mathcal{T} >$

The instantaneous cost of taking an action in a state

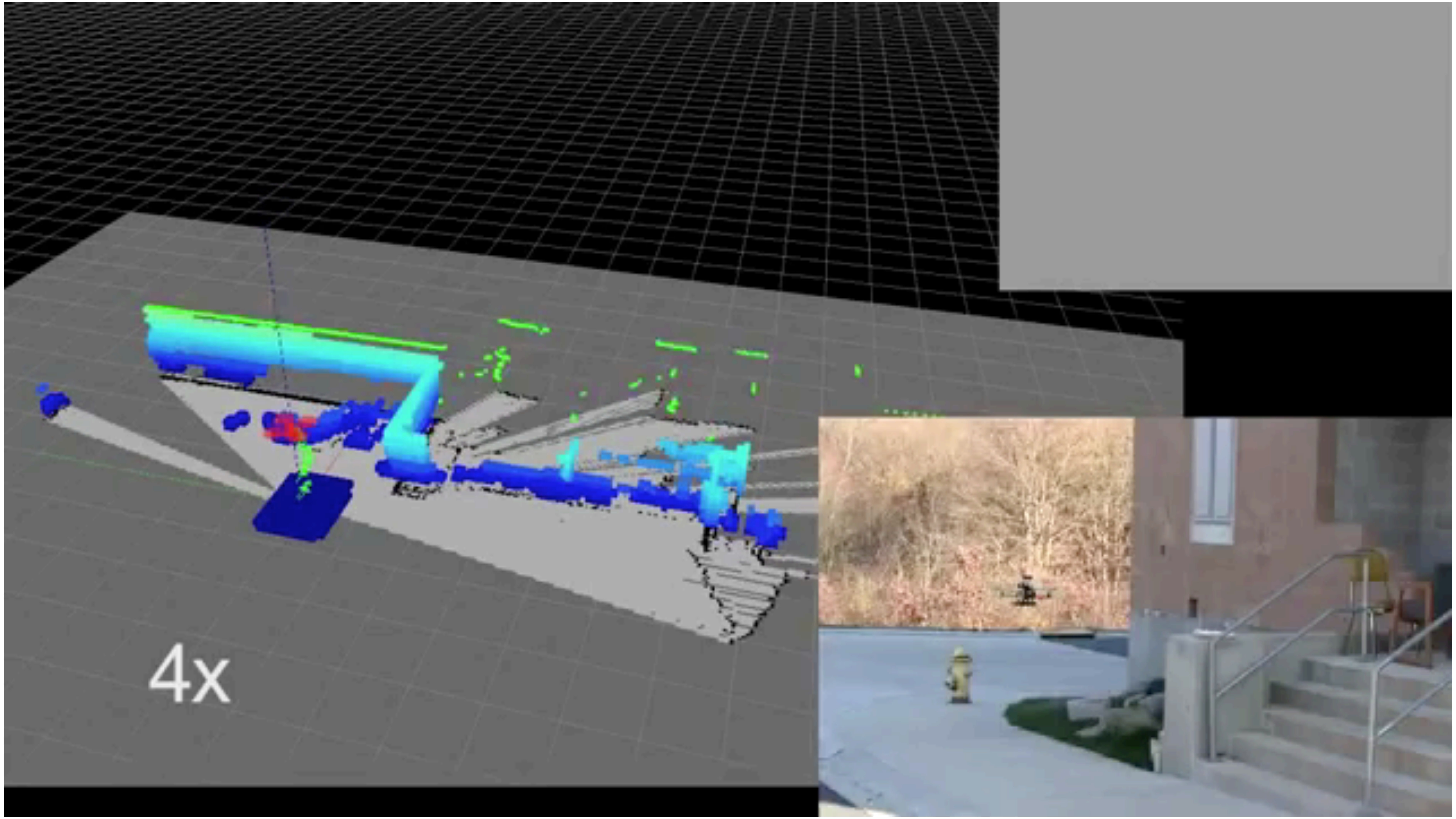
$$c(s, a)$$







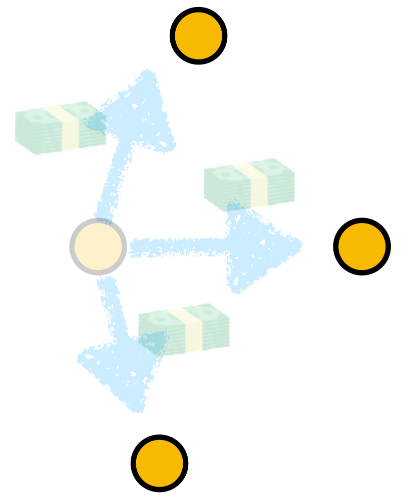
Examples of *non-Markovian* cost?



“Autonomous Multi-Floor Indoor Navigation with a Computationally Constrained MAV”, S. Shen, N. Michael, V.Kumar, 2010

Transition

$< S , A , C , \mathcal{T} >$

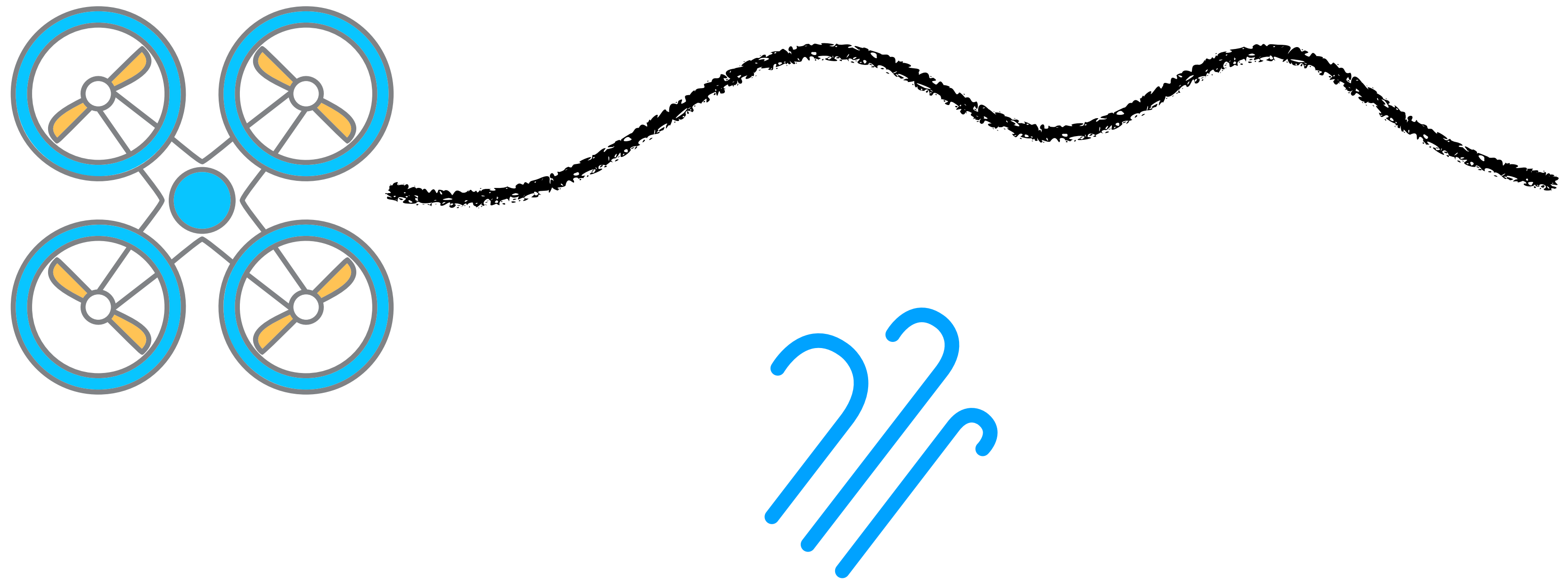


The next state given state and action

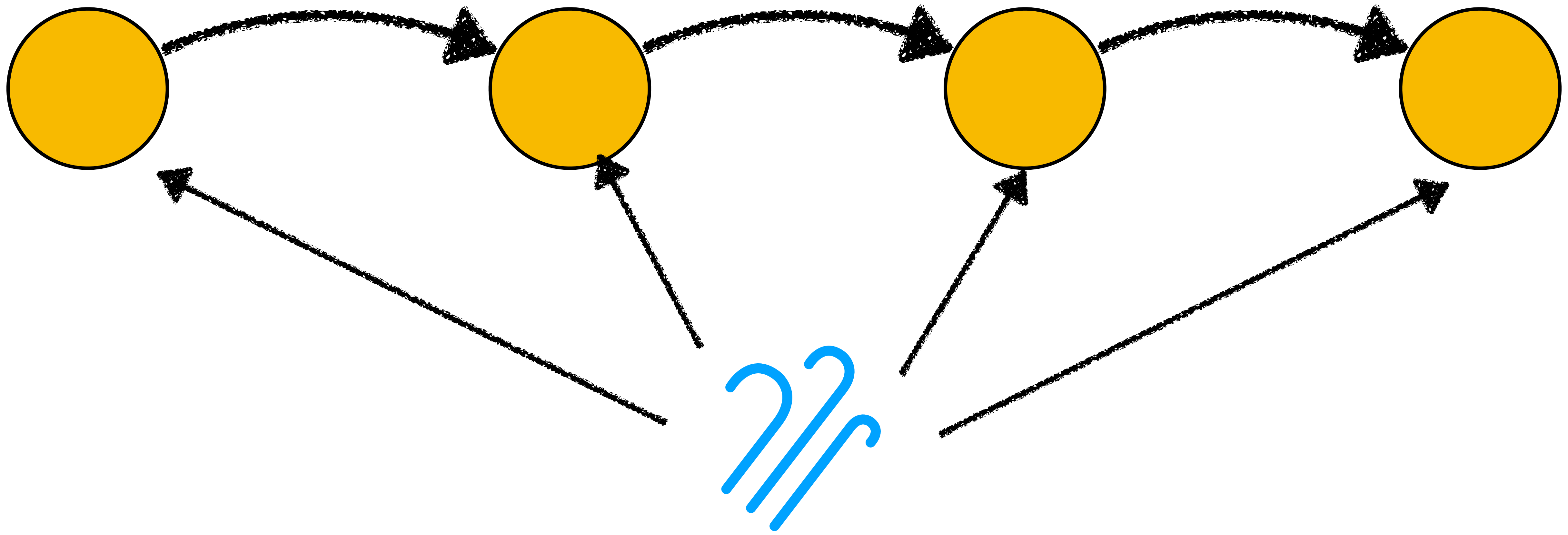
$$s' = \mathcal{T}(s, a) \qquad\qquad s' \sim \mathcal{T}(s, a)$$

Deterministic — Stochastic

Examples of *non-Markovian* dynamics?



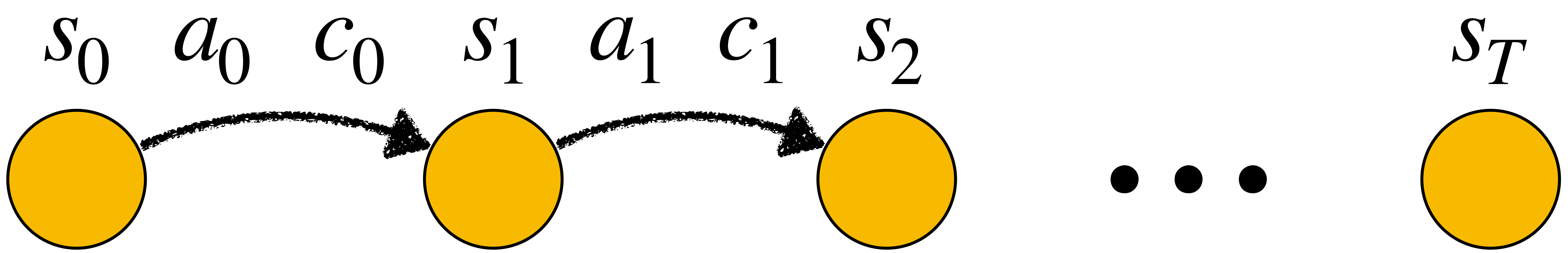
Wind correlates disturbance across time



Markov Decision ~~Process~~ → Problem

Includes things to define an optimization problem

Horizon $T \in \mathbb{N}$



Discount $0 \leq \gamma \leq 1$

Return: $c_0 + \gamma c_1 + \cdots \gamma^{T-1} c_{T-1}$

(Costs are more valuable if they happen soon)

Markov Decision ~~Process~~ → Problem

Policy

$$\pi \in \Pi$$

$$\pi : s_t \rightarrow a_t$$ (Deterministic)

$$\pi : s_t \rightarrow P(a_t)$$ (Stochastic)

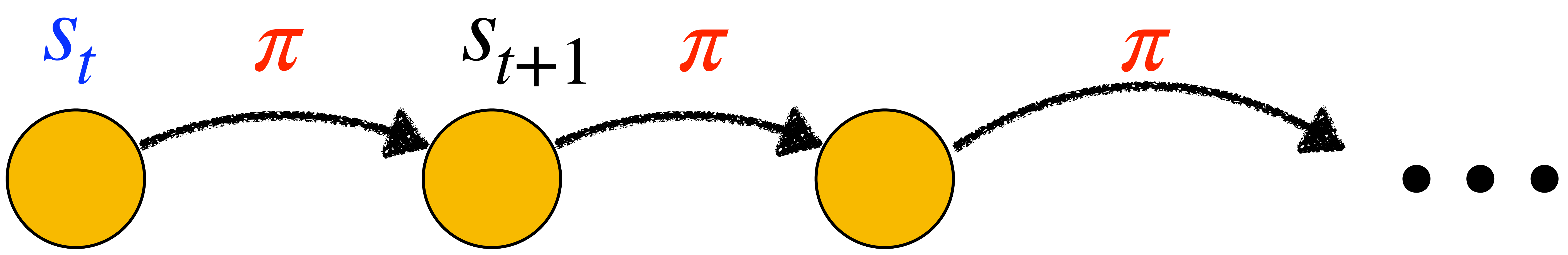
A function that maps state (and time) to action

Objective Function

$$\min_{\pi} \mathop{\mathbb{E}}_{\substack{a_t \sim \pi(s_t) \\ s_{t+1} \sim \mathcal{T}(s_t, a_t)}} [\sum_{t=0}^{T-1} \gamma^t c(s_t, a_t)]$$

Find policy that minimizes sum of discounted future costs

Value of a state

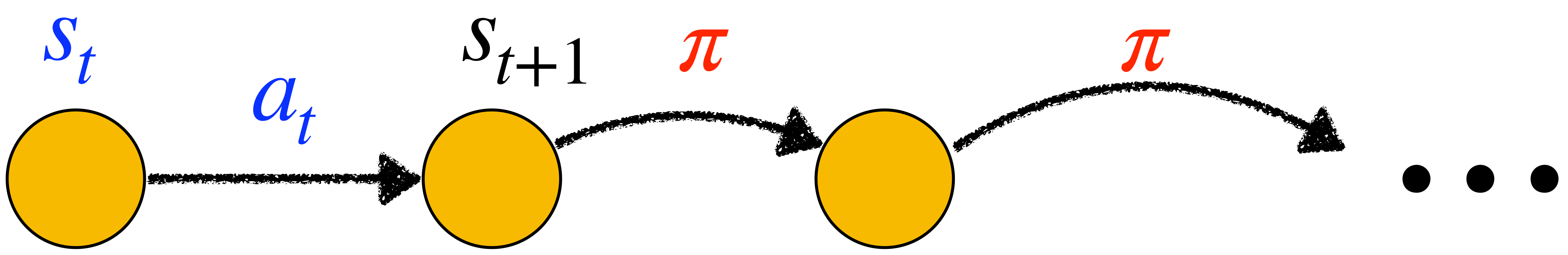


$$V^{\pi}(s_t) = c_t + \gamma c_{t+1} + \gamma^2 c_{t+2} + \dots$$

Expected discounted sum of cost from starting at a state and following a policy from then on

$$\pi^* = \arg\min_{\pi} \mathbb{E}_{s_0} V^{\pi}(s_0)$$

Value of a state-action



$$Q^{\pi}(s_t, a_t) = c_t + \gamma c_{t+1} + \gamma^2 c_{t+2} + \cdots$$

Expected discounted sum of cost from starting at a state, executing action and following a policy from then on

---

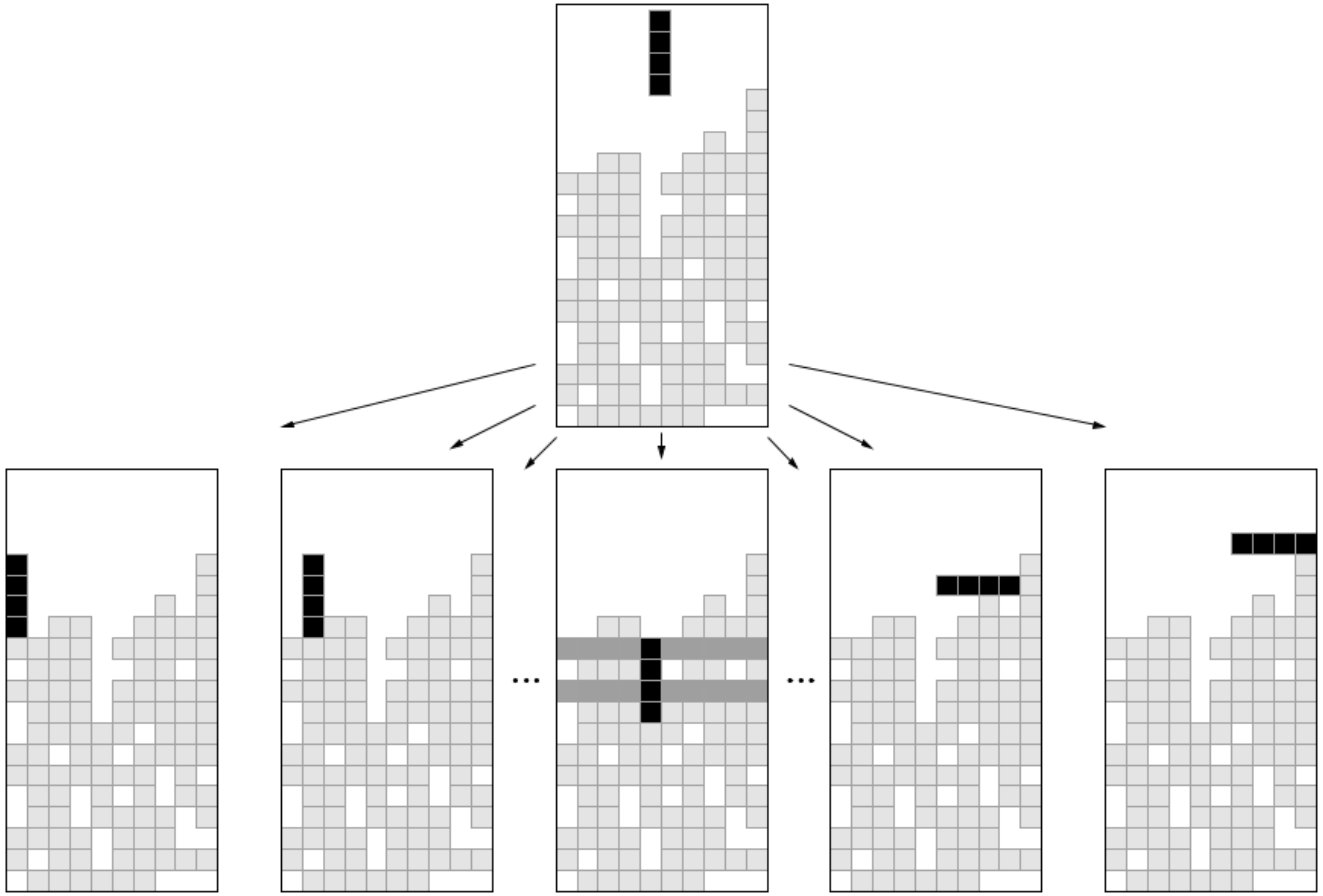
$$Q^{\pi}(s_t, a_t) = c(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim \mathcal{T}(s_t, a_t)} V^{\pi}(s_{t+1})$$

Values matter

Let's build some intuition!

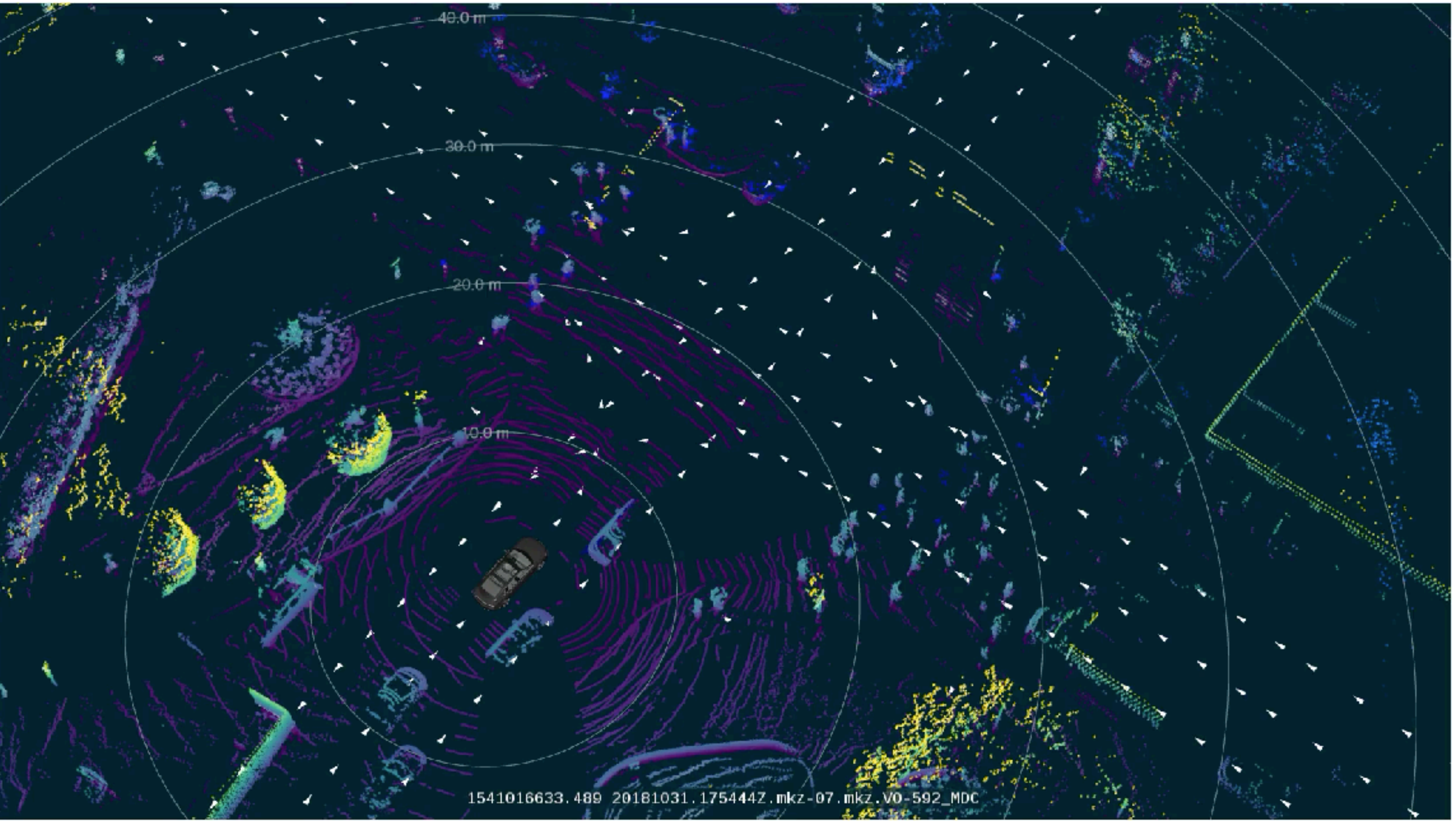
Case studies

Example 1: Tetris!



$$< S, A, C, \mathscr{T} >$$

Example 2: Self-driving



$$< S , A , C , \mathcal{T} >$$

?

Example 3: Coffee making robot

$$< S\,, A\,, C\,, \mathcal{T} >$$

?

Solving MDPs

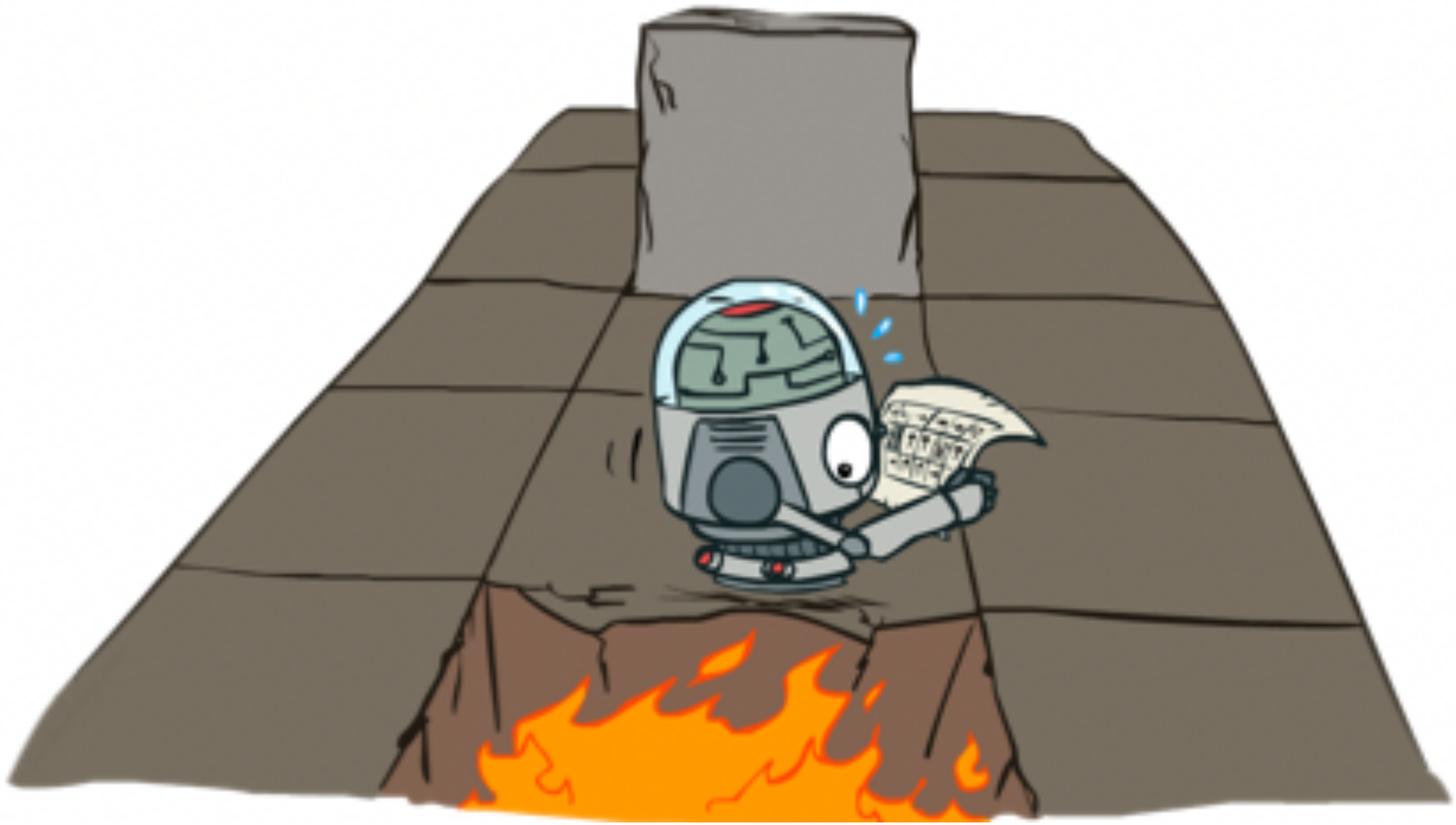
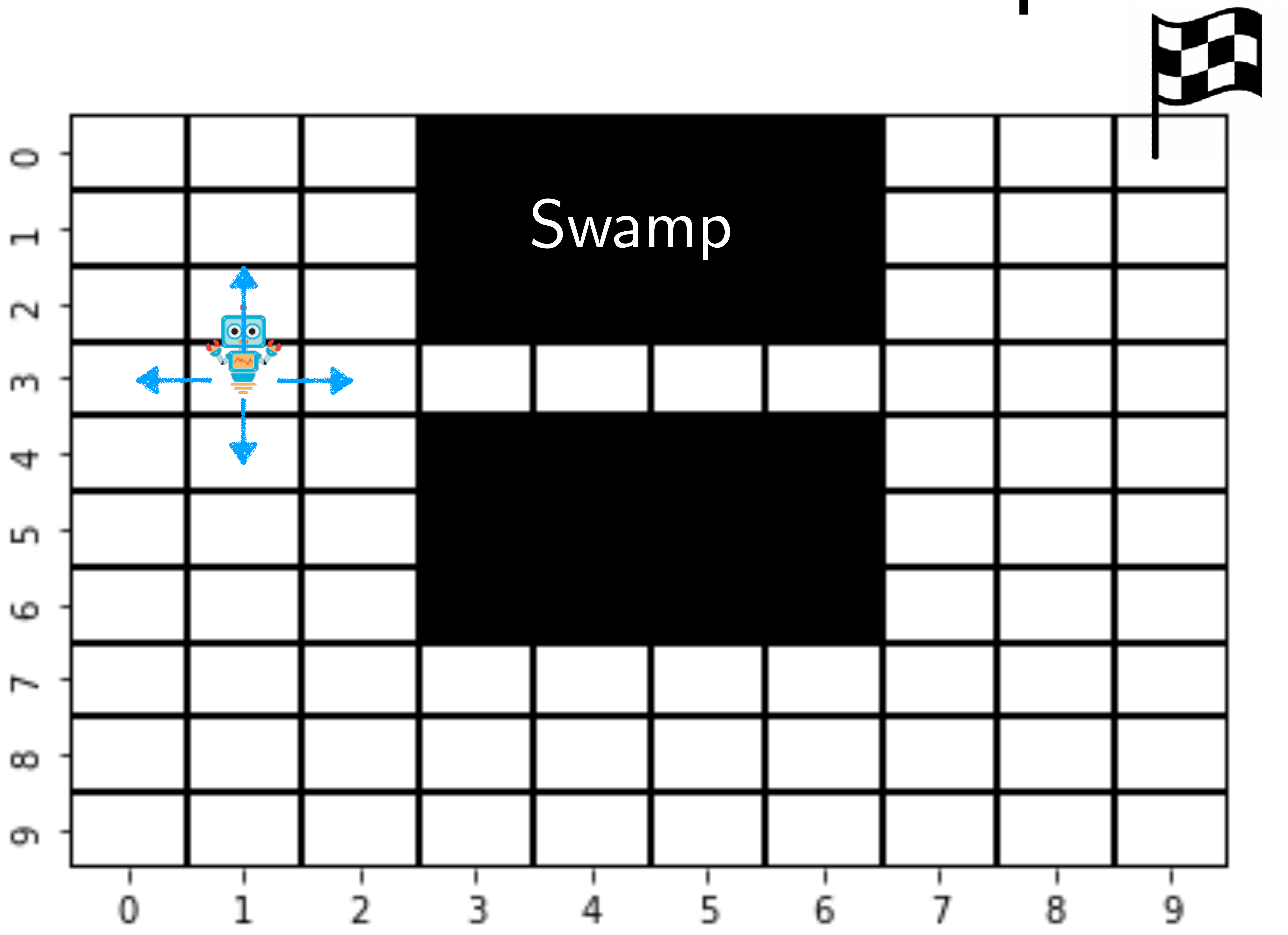


Image courtesy Dan Klein

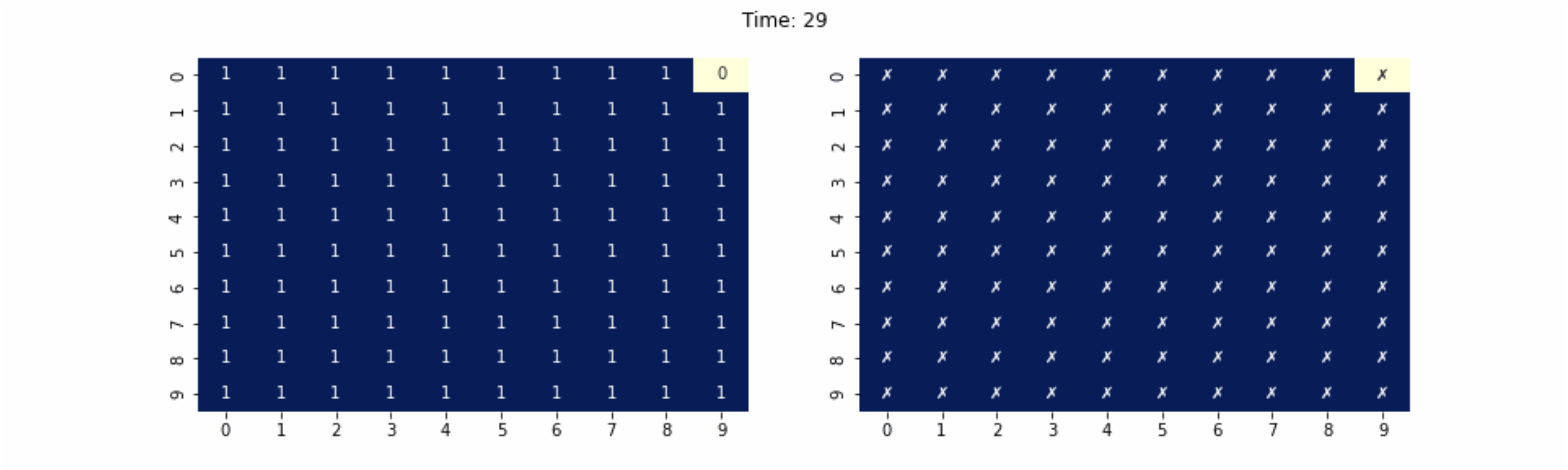
Setup



$$< S , A , C , \mathcal{T} >$$

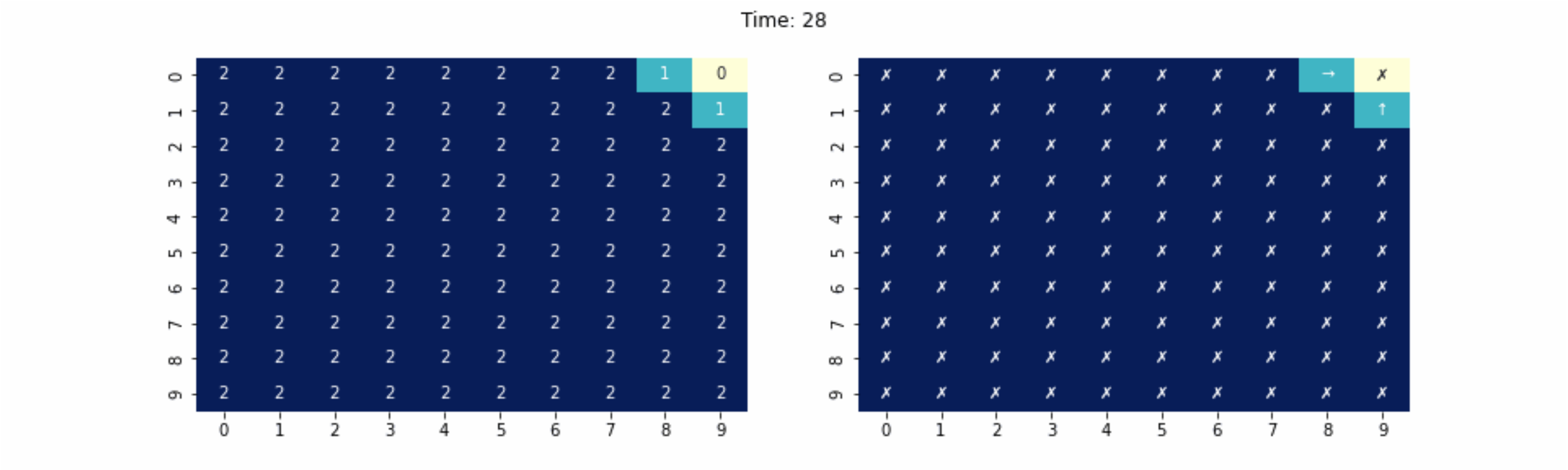
- Two absorbing states: Goal and Swamp
- Cost of each state is 1 till you reach the goal
- Let's set T = 30

What is the optimal value at T-1?



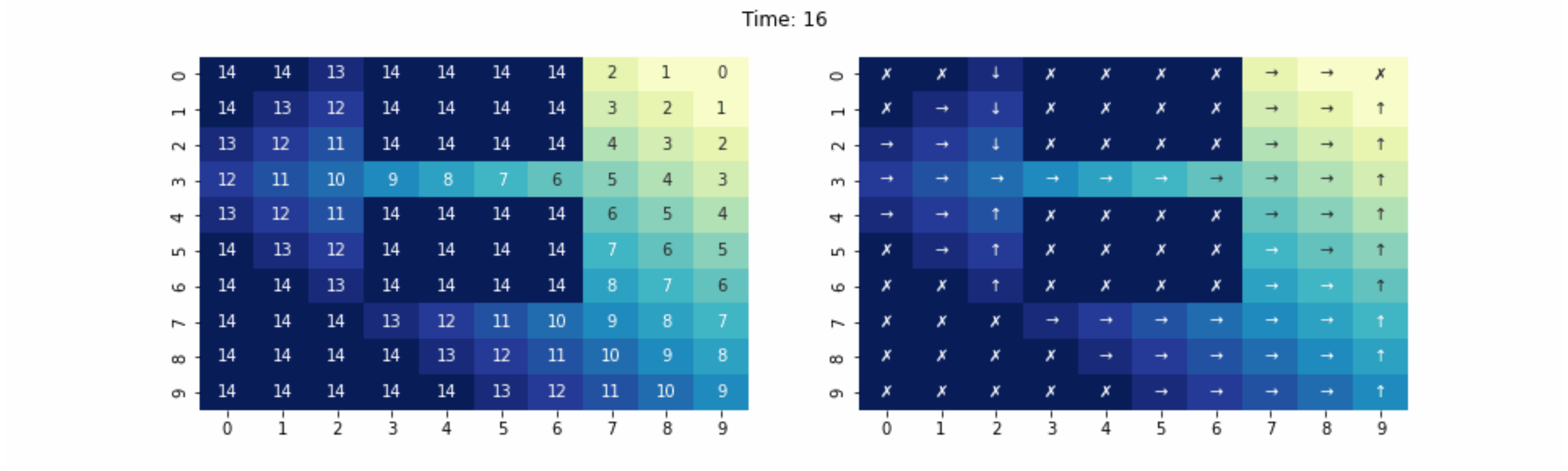
$$V^*(s_{T-1}) = \min_a c(s_{T-1}, a) \qquad \pi^*(s_{T-1}) = \arg\min_a c(s_{T-1}, a)$$

What is the optimal value at T-2?



$$V^*(s_{T-2}) = \min_a [c(s_{T-2}, a) + V^*(s_{T-1})] \qquad \pi^*(s_{T-2}) = \arg\min_a [c(s_{T-2}, a) + V^*(s_{T-1})]$$

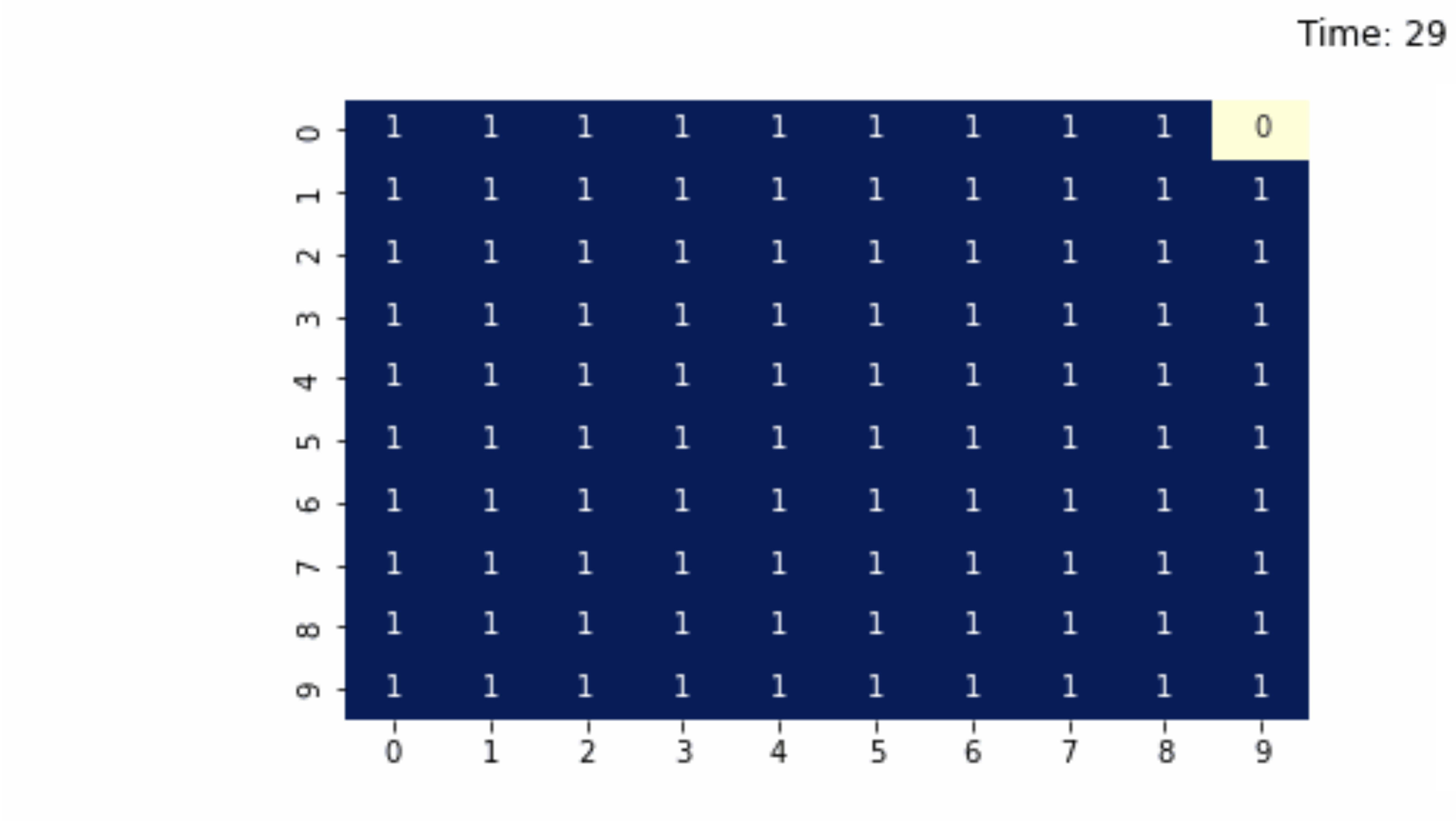
Dynamic Programming all the way!



$$V^*(s_t) = \min_a [c(s_t, a) + V^*(s_{t+1})]$$

$$\pi^*(s_t) = \arg\min_a [c(s_t), a) + V^*(s_{t+1})]$$

Value Iteration



Algorithm 4: Dynamic Programming Value Iteration for computing the optimal value function.

Algorithm `OptimalValue`(x, T)
- **for** $t = T-1, \ldots, 0$ **do**
  - **for** $x \in \mathbb{X}$ **do**
    - **if** $t = T-1$ **then**
      - $V(x,t) = \min_a c(x,a)$
    - **end**
    - **else**
      - $V(x,t) = \min_a c(x,a) + \sum_{x' \in \mathbb{X}} p(x'|x,a)V(x,t+1)$
    - **end**
  - **end**
- **end**

What is the complexity?

Deterministic	Stochastic	Efficient
$S \times A \times T$	$S^2 \times A \times T$	$k \times S \times A \times T$

Why is the optimal policy a function of time?

Pulling the goalie
when you
are losing and have
seconds left ..

To infinity!

Infinite horizon cases

$$V^*(s_t) = \min_{a_t} [c(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim \mathcal{T}(s_t, a_t)} V^*(s_{t+1})]$$

↓ Fixed point as $t \to \infty$

$$V^*(s) = \min_{a} [c(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s, a)} V^*(s)]$$

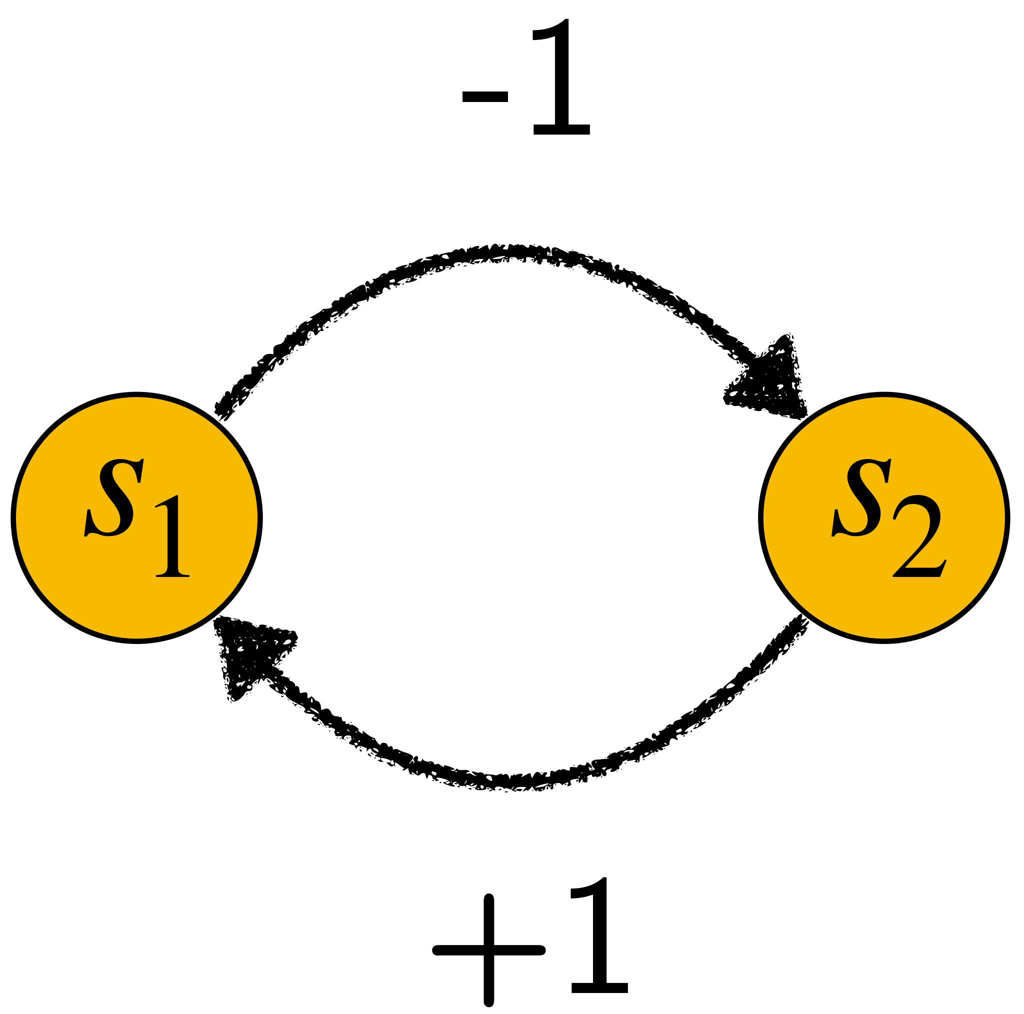
Bellman Equation

$$V^*(s) = \min_a \left[c(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s,a)} V^*(s)\right]$$

Does this converge?

How fast does it converge?

Does value iteration converge?



$\gamma = 1$

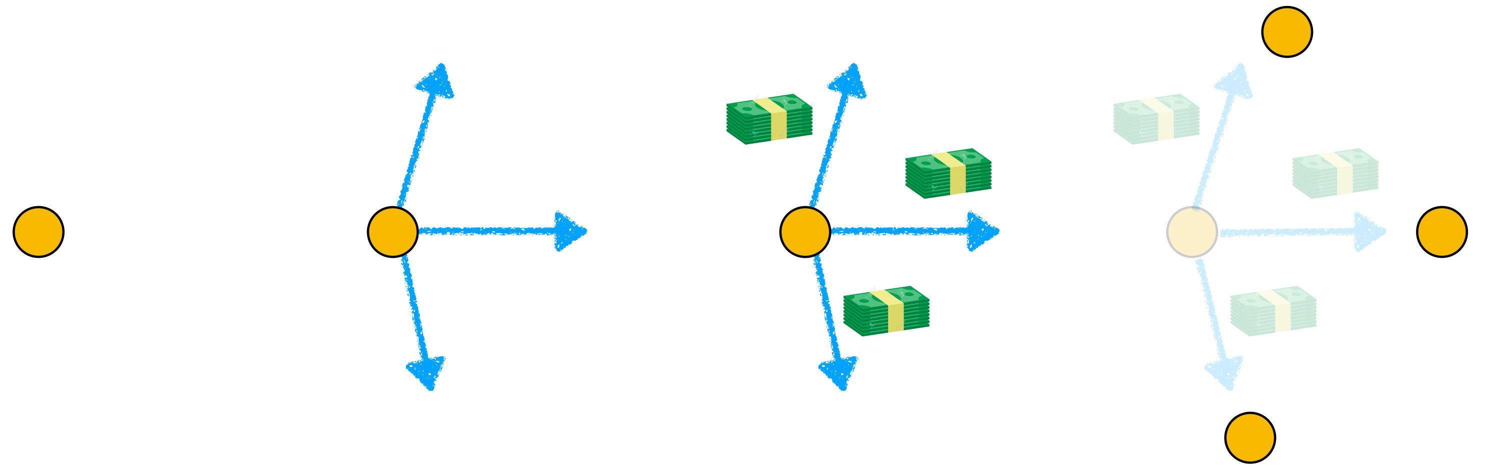
What is $V^*(s_1)$? What is $V^*(s_2)$?

tl;dr

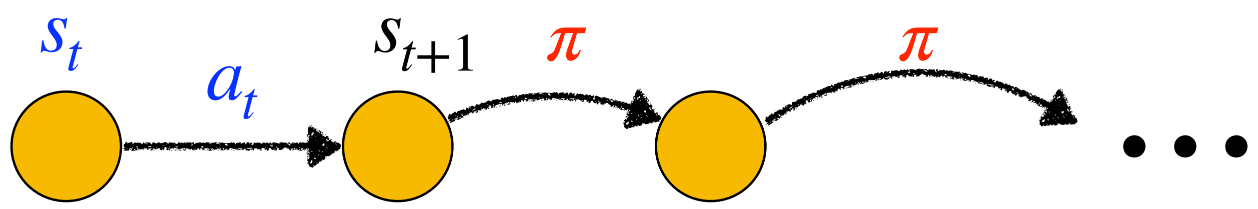
Markov Decision Process

A mathematical framework for modeling sequential decision making

$$< S , A , C , \mathcal{T} >$$



Value of a state-action

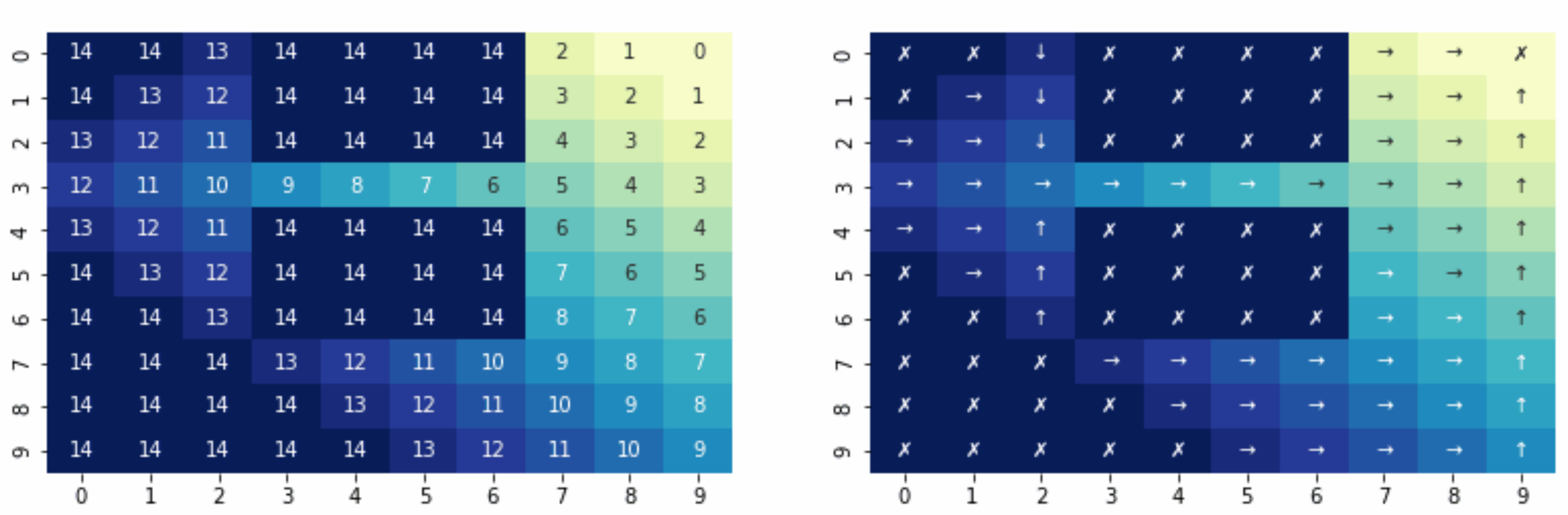


$$Q^{\pi}(s_t, a_t) = c_t + \gamma c_{t+1} + \gamma^2 c_{t+2} + \cdots$$

Expected discounted sum of cost from starting at a state, executing action and following a policy from then on

$$Q^{\pi}(s_t, a_t) = c(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim \mathcal{T}(s_t, a_t)} V^{\pi}(s_{t+1})$$

Dynamic Programming all the way!



$$V^*(s_t) = \min_a [c(s_t, a) + V^*(s_{t+1})] \qquad \pi^*(s_t) = \arg\min_a [c(s_t), a) + V^*(s_{t+1})]$$