Review

Prelim

- In-class prelim, 75 minutes
- Format
  - Multiple choice questions (similar to quizzes)
  - Written questions (similar to written assignments A1, A3)
- Scope: Everything until last lecture (actor critic)

Today's plan

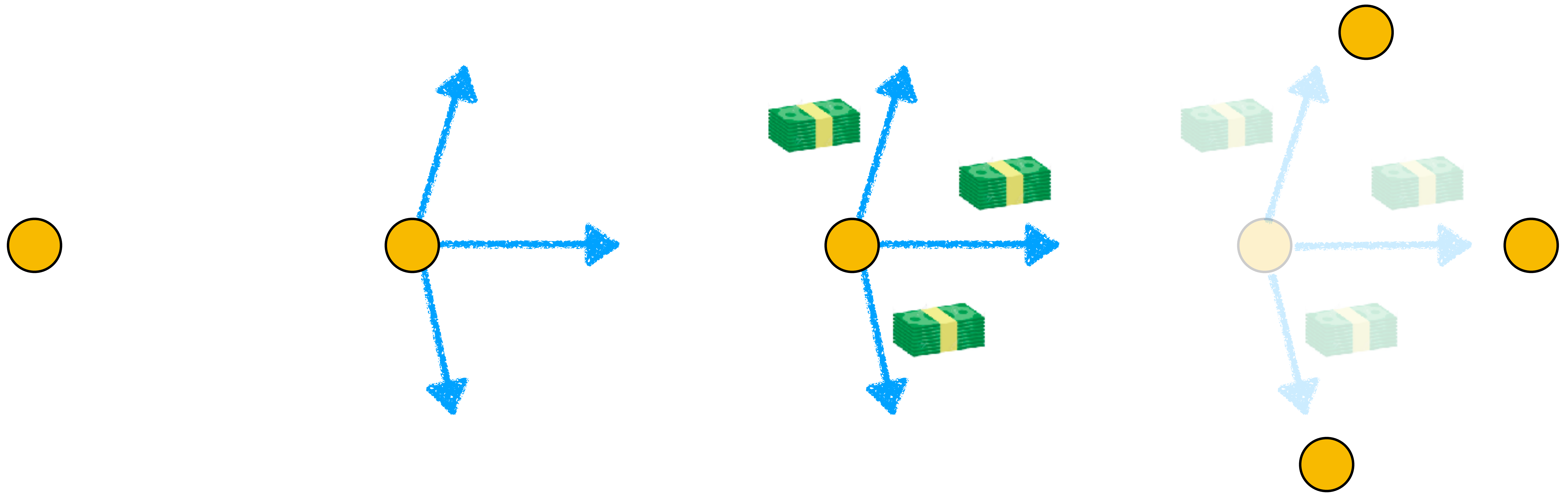
- Go through the greatest hits
- Answer questions YOU have
- Today we will spend more time on MDP, RL and less time on imitation learning

Fundamentals: MDP

Markov Decision Process

A mathematical framework for modeling sequential decision making

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



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

$$\theta_t$$

$$\dot{\theta}_t$$

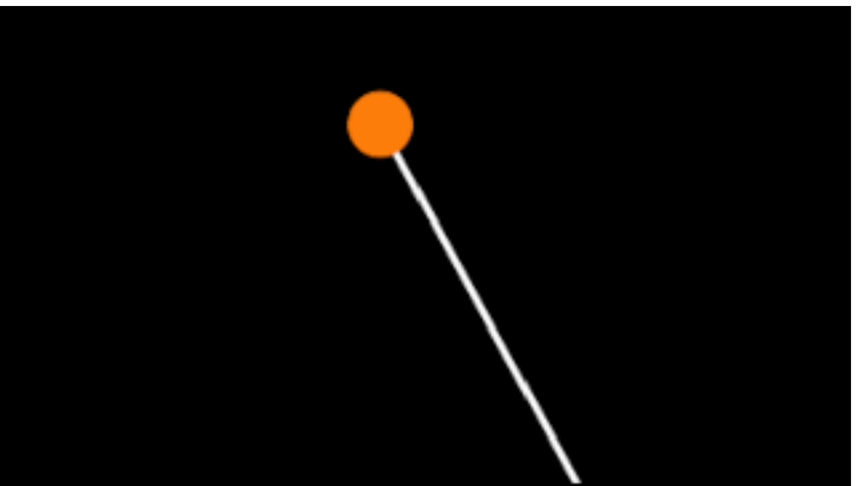
$$\tau$$

$$\frac{1}{2}\theta^2 + \frac{1}{2}\dot{\theta}^2 + \frac{1}{2}\tau^2$$

$$\theta_{t+1} = \theta_t + \dot{\theta}_t \Delta_t$$

$$\dot{\theta}_{t+1} = \dot{\theta}_t + \ddot{\theta}_t \Delta_t$$

$$I\ddot{\theta}_t = mgl\sin(\theta) + \tau$$



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

$\theta_t \in \mathbb{R}^{12}$

(All joints)

$\dot{\theta}_t \in \mathbb{R}^{12}$

(All joint vel)

x, y, ψ

(2d pos, heading)

c_1, c_2, c_3, c_4

(Contact state of feet)

$\tau \in \mathbb{R}^{12}$

(12 torque)

Move at desired vel

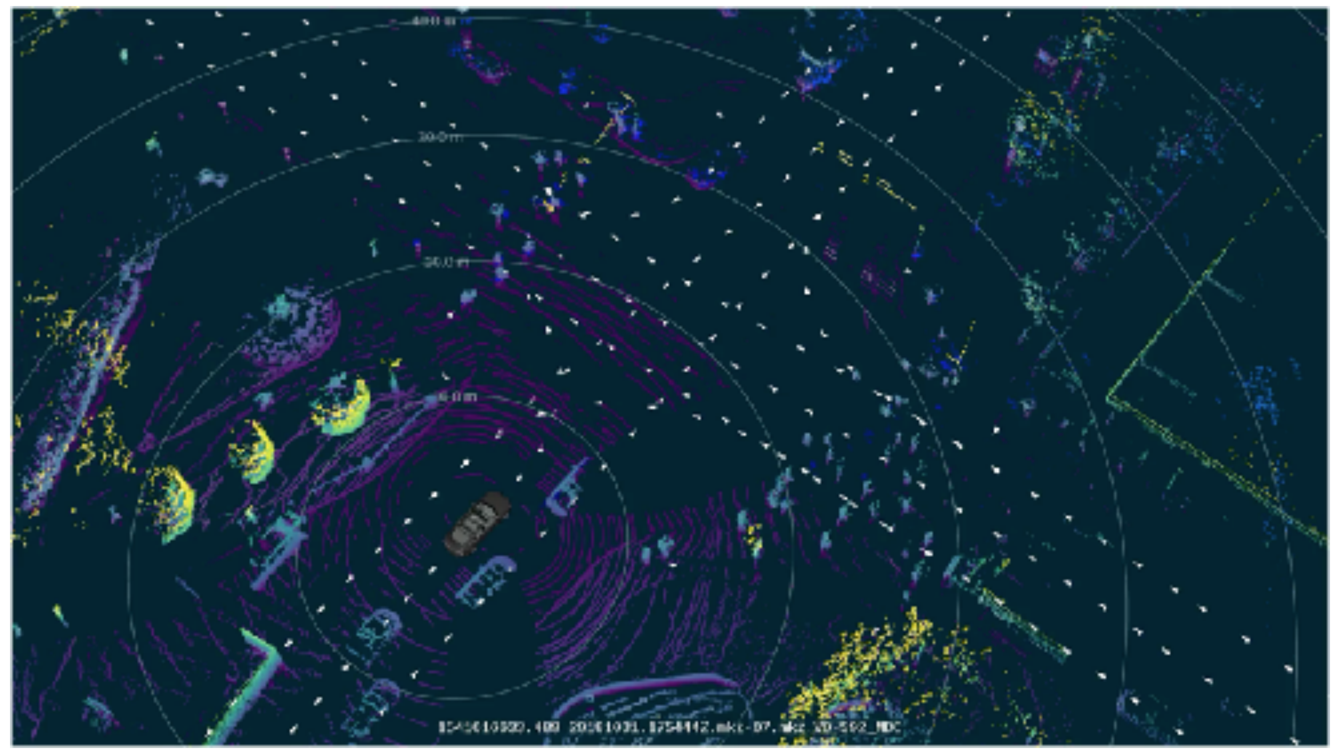
$+$

Minimize torque

Newton-Euler Equation

But need to know ground terrain (Which is typically unknown)

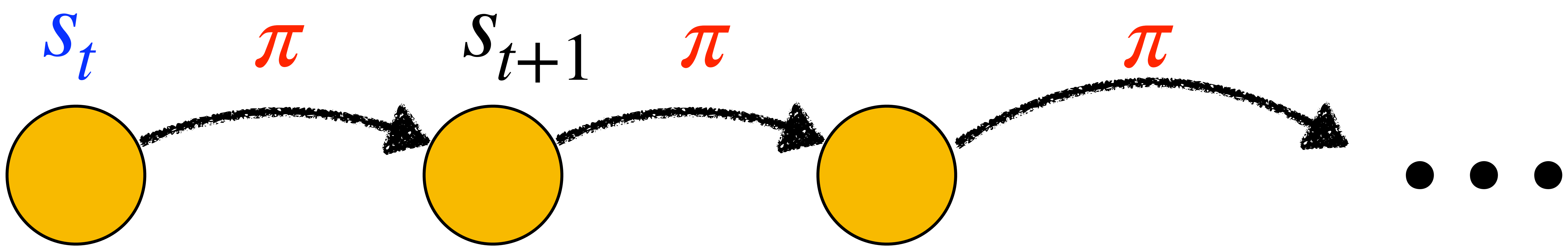
$$S, A, C, \mathcal{T}$$

S	A	C	$\mathcal{T}$
State of car	Steering Gas	Penalty for not reaching goal	Dynamics of car (Known)
State of all other agents		Penalty for violating constraints (Safety, rules)	Dynamics/intent of other agents (Unknown)
State of traffic lights		Penalty for high control effort	Transition of traffic light (Hidden variable)

The "Value" Function

$$V^{\pi}(s_t)$$

Read this as: Value of a policy at a given state and time



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

The Bellman Equation

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

Value of current state

Cost

Value of future state

Optimal policy

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

Bellman Equation for the Optimal Policy

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

Optimal Value

Cost

Optimal Value of Next State

We use V^* to denote optimal value

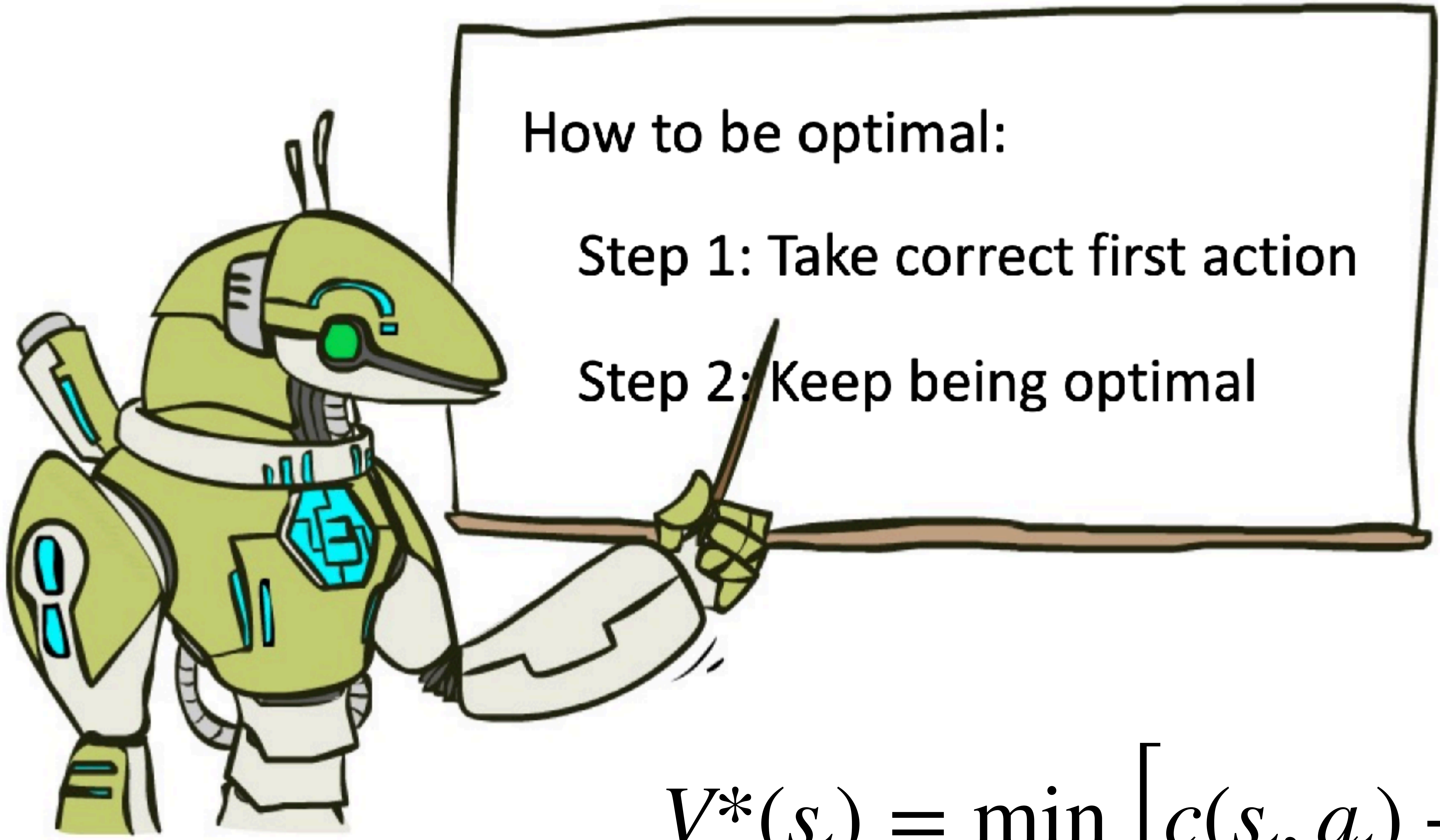
$$V^*(s_t) = \min_{a_t} \left[c(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}} V^*(s_{t+1})) \right]$$

Optimal Value

Cost

Optimal Value of Next State

The Bellman Equation

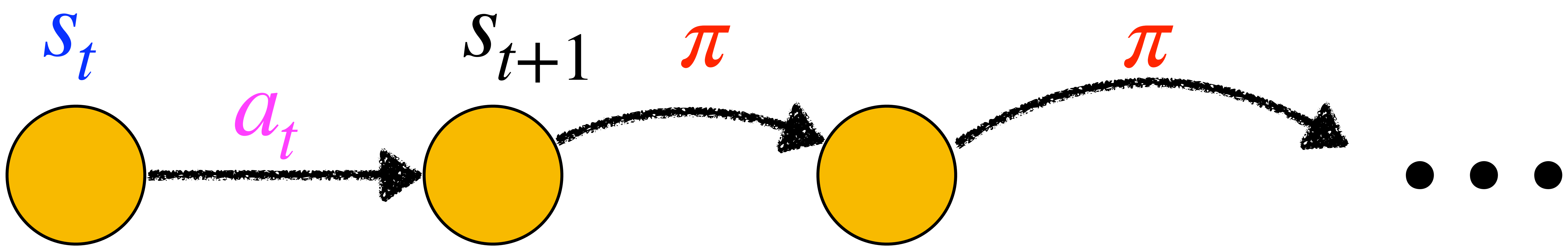


$$V^*(s_t) = \min_{a_t} \left[c(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}} V^*(s_{t+1})) \right]$$

Image courtesy Dan Klein

The "Action Value" Function

$$Q^{\pi}(s_t, a_t)$$



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

Quiz: Express V in terms of Q?

$$V^{\pi}(s_t) = \mathbb{E}_{a_t \sim \pi(s_t)} Q^{\pi}(s_t, a_t)$$

Express Q in terms of V?

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

The Bellman Equation

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

Value of current state

Cost

Value of future state

We use Q^* to denote optimal value

$$Q^*(s_t, a_t) = c(s_t, a_t) + \min_{a_{t+1}} \left[\gamma \mathbb{E}_{s_{t+1}} Q^*(s_{t+1}, a_{t+1})) \right]$$

Optimal Value

Cost

Optimal Value of Next State

Everything you can do with V, you can do with Q!

Value Iteration, Policy Iteration, Approximate Value Iteration, Approximate Policy Iteration, ...

You can also translate cost to reward

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

$$\Downarrow$$

$$V^*(s_t) = \max_{a_t} \left[r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}} V^*(s_{t+1}) \right]$$

The Advantage Function

$$A^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t)$$

How much better is it to take action a_t vs action $\pi(s_t)$?

(given you roll-out with π from there on)

The Advantage Function

$$A^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) - Q^{\pi}(s_t, \pi(s_t))$$

90   100   10

$$\begin{array}{r|c|}
\hline
 & 0 \\ \hline
a_t & 100 \\ \hline
 & 10 \\ \hline
 & 1 \\ \hline
 & 0 \\ \hline
\end{array}
\qquad\qquad
\begin{array}{|c|l}
\hline
0 & \\ \hline
100 & \\ \hline
10 & \pi(s_t) \\ \hline
1 & \\ \hline
0 & \\ \hline
\end{array}$$

$Q^{\pi}(s_t, .)$ $Q^{\pi}(s_t, .)$

Questions?

Questions

1. Express V as Q? Express Q in terms of V?

2. If a genie offered you V* or Q*, which one would you take? Why?

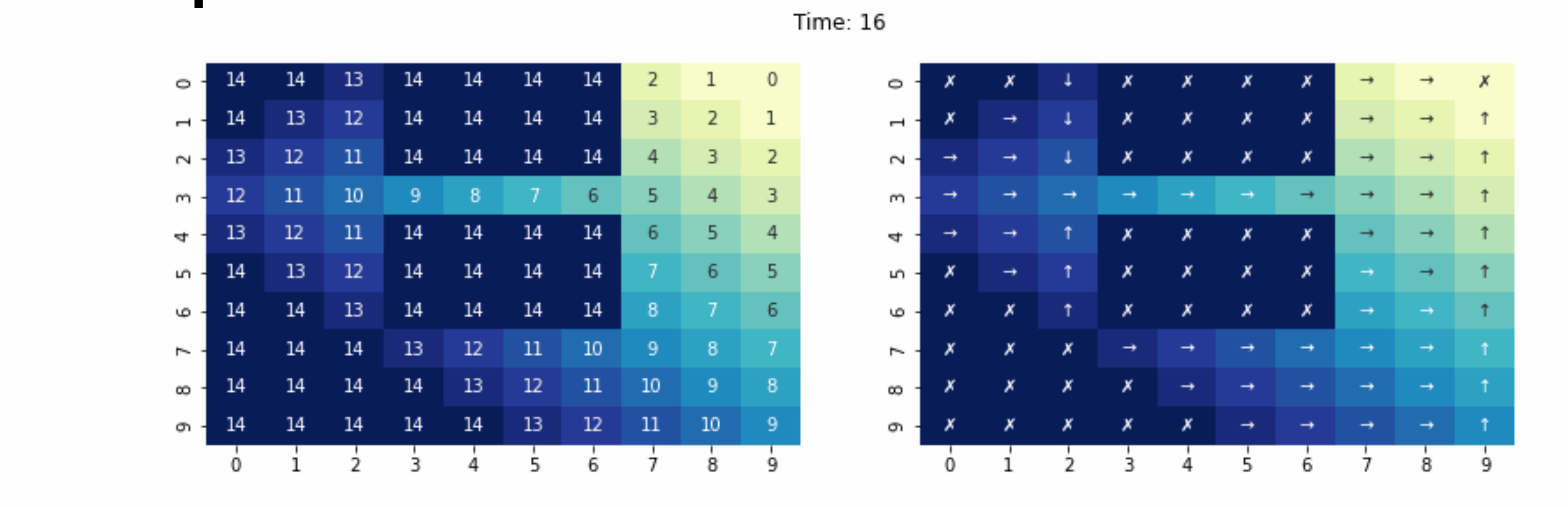
3. What is Bellman Equation over infinite horizon?

Solving Known MDP (Planning)

Value Iteration (Finite Horizon)

Initialize value function at last time-step

$$V^*(s, T-1) = \min_a c(s, a)$$



for $t = T-2, \ldots, 0$

Compute value function at time-step t

$$V^*(s, t) = \min_a \left[c(s, a) + \gamma \sum_{s'} \mathcal{T}(s' | s, a) V^*(s', t+1) \right]$$

Infinite Horizon Value Iteration

Initialize with any value function $V^*(s)$

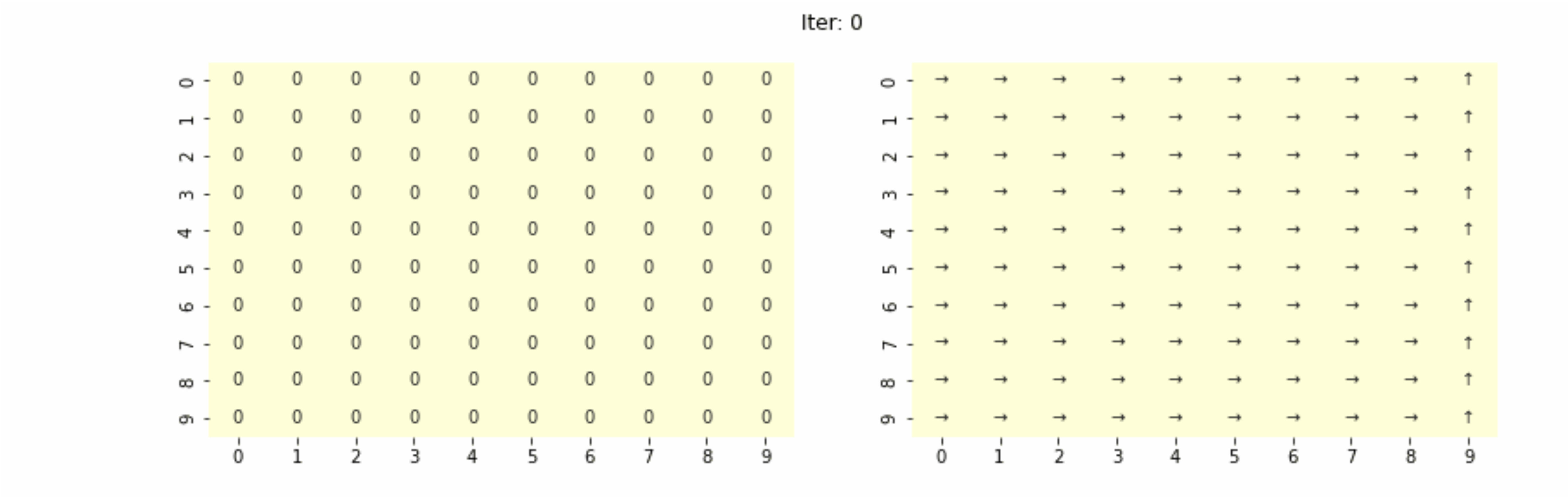
Repeat until convergence

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

Sometimes, it's faster to iterate over policies than values

Policy Iteration (Infinite horizon)

Init with some policy π



Repeat forever

Evaluate policy

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

Improve policy

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

You can translate from V to Q!

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

Value iteration

$$Q^*(s,a) = c(s,a) + \gamma \sum_{s'} \mathcal{T}(s' \mid s,a) \min_{a'} Q^*(s',a')$$

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

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

Policy iteration

$$Q^\pi(s,a) = c(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(s,a)} Q^\pi(s',\pi(s'))]$$

$$\pi^+(s) = \arg\min_a Q^\pi(s,a)$$

Linear Quadratic Regulator (LQR)

$$V^*(s,t) = \min_a \left[c(s,a) + \gamma \sum_{s'} \mathcal{T}(s' \mid s,a) V^*(s', t+1) \right]$$

(Quadratic) (Quadratic) (Linear) (Quadratic)

$\frac{1}{2} x_t^T V_t x_t$ $x_t^T Q x_t + u_t^T R u_t$ $x_{t+1} = A_t x_t + B_t u_t$ $\frac{1}{2} x_{t+1}^T V_{t+1} x_{t+1}$

How can we *analytically* do value iteration?

The LQR Algorithm

Initialize $V_T = Q$

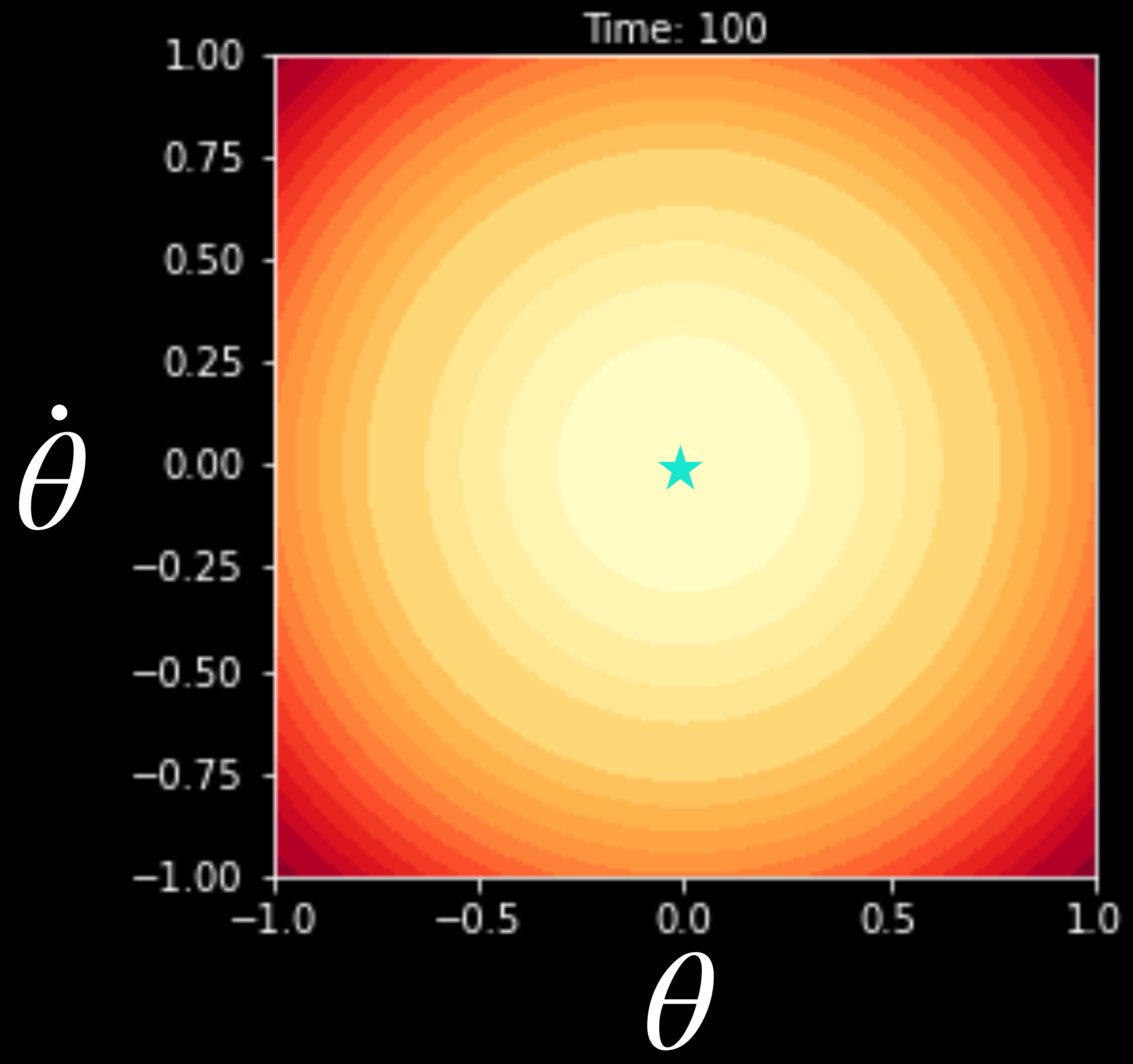
For t = T-1, ..., 1

Compute gain matrix

$$K_t = (R + B^T V_{t+1} B)^{-1} B^T V_{t+1} A$$

Update value

$$V_t = Q + K_t^T R K_t + (A + BK_t)^T V_{t+1} (A + BK_t)$$



LQR Converges

Q is positive semi-definite

$$x^T Q x \geq 0$$

State costs are always non-negative

R is positive definite

$$u^T R u > 0$$

for $u \neq 0$

Control cost are always positive

Questions?

Questions

1. Why might we prefer policy iteration over value iteration?

2. How can I apply LQR if my MDP is not linear and quadratic?

Unknown MDP
(Reinforcement Learning)

Why is it hard to solve unknown MDP?

Just run Value iteration?

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s'} \mathcal{T}(s' | s,a) \max_{a'} Q^*(s',a') \quad \forall (s,a)$$

Don't know,
Need to collect data!

Solution:
1. Collect a batch of data 2. Fit a function approximator to Q

Recap: Fitted Q-Iteration

Receive some dataset $\mathcal{D} = \{(s, a, r, s')\}$

Initialize $\hat{Q}_0 \in \mathcal{F}_Q$, $t \leftarrow 0$

for $t \in 1, \ldots, T$

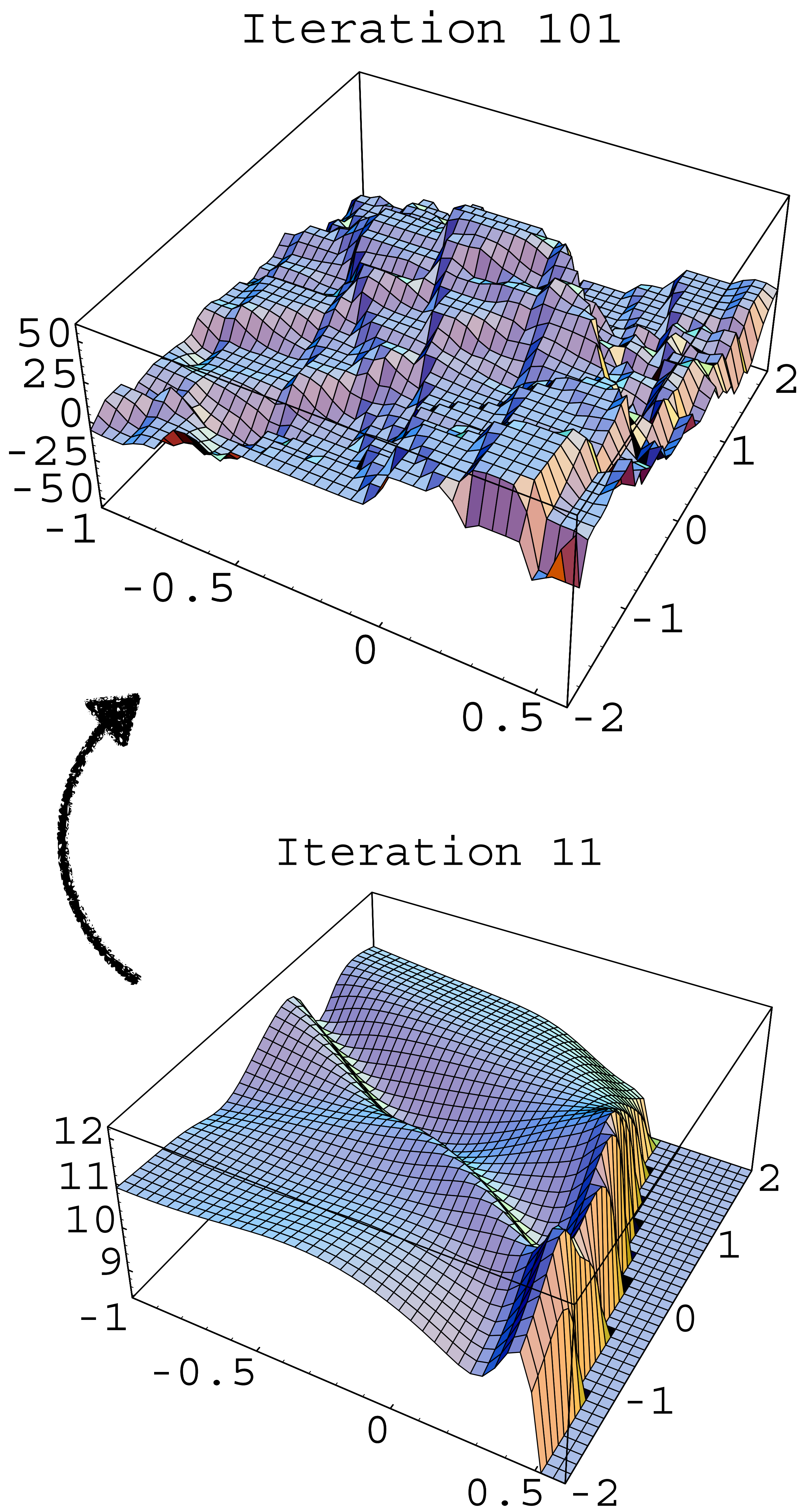
$$\hat{Q}_{t+1} \leftarrow \arg\min_{Q \in \mathcal{F}_Q} \mathbb{E}_{\mathcal{D}}[(Q(s, a) - (r + \max_{a' \in \mathcal{A}} \hat{Q}_t(s', a')))^2]$$

Return π_T

The problem of Function Approximation!

Errors in approximation are amplified! Why?

$$\hat{Q}_{t+1} \leftarrow \arg\min_{Q \in \mathcal{F}_Q} \mathbb{E}_{\mathcal{D}}[(Q(s,a) - (r + \max_{a' \in \mathcal{A}} \hat{Q}_t(s', a')))^2]$$



Let's work out an example

Recap: Approximate Policy Iteration

Initialize with arbitrary π_0, $t = 0$

for $t \in 1, \ldots, T$

Sample $\mathcal{D}_t = \{(s_h, a_h, \hat{Q} = \sum_{\tau=h}^{H} r(s_\tau, a_\tau)) \sim \pi_t\}$

Fit $\hat{Q}_t \leftarrow \arg\min_{Q \in \mathcal{F}_Q} \mathbb{E}_{\mathcal{D}_t}[(Q(s,a) - \hat{Q})^2]$

$\pi_{t+1}(s) = \arg\max_{a \in \mathcal{A}} \hat{Q}_t(s,a)$

if $\pi_{t+1} = \pi_t$: **break;**

Return π_T

Performance Difference Lemma (PDL)

$$V^{\pi^+}(s_0) - V^{\pi}(s_0) = \sum_{t=0}^{T-1} \mathbb{E}_{s_t \sim d_t^{\pi^+}} A^{\pi}(s_t, \pi^+)$$

Problem with Approximate Policy Iteration

$$V^{\pi^+}(s_0) - V^{\pi}(s_0) = \sum_{t=0}^{T-1} \mathbb{E}_{s_t \sim d_t^{\pi^+}} A^{\pi}(s_t, \pi^+)$$

PDL requires accurate Q_θ^π on states that π^+ will visit! $(d_t^{\pi^+})$

But we only have states that π visits (d_t^π)

If π^+ changes drastically from π, then $|d_t^{\pi^+} - d_t^\pi|$ is big!

Policy Gradients

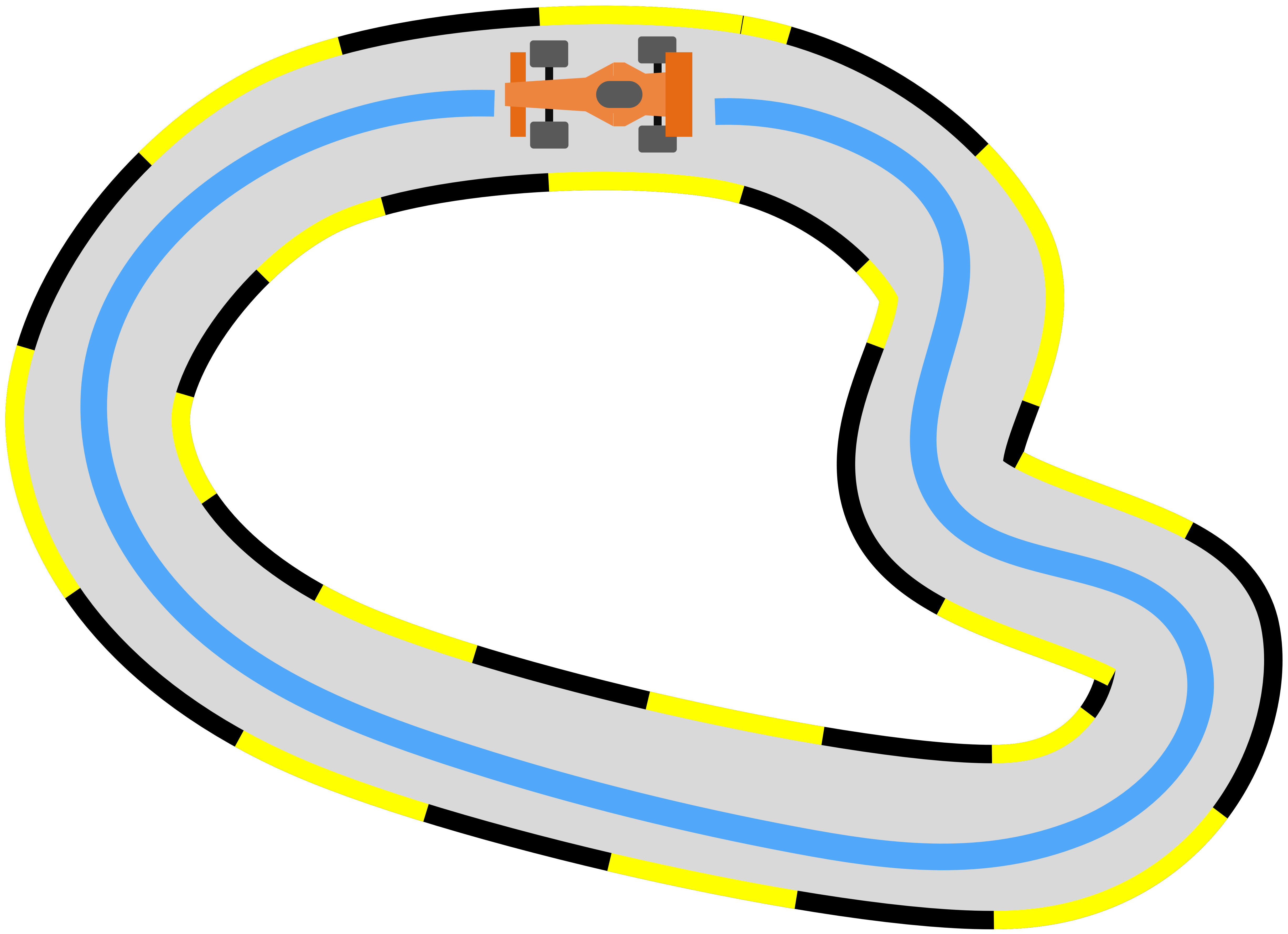
$$\nabla_\theta J = E_{s \sim d^{\pi_\theta}(s), a \sim \pi_\theta(a|s)} \left[\nabla_\theta \log \pi_\theta(a|s) Q^{\pi_\theta}(s,a) \right]$$

$$\nabla_\theta J = E_{d^{\pi_\theta}(s)} E_{\pi_\theta(a|s)} \left[\nabla_\theta \log(\pi_\theta(a|s) A^{\pi_\theta}(s,a) \right]$$

Questions?

Unknown MDP
(Imitation Learning)

Behavior Cloning

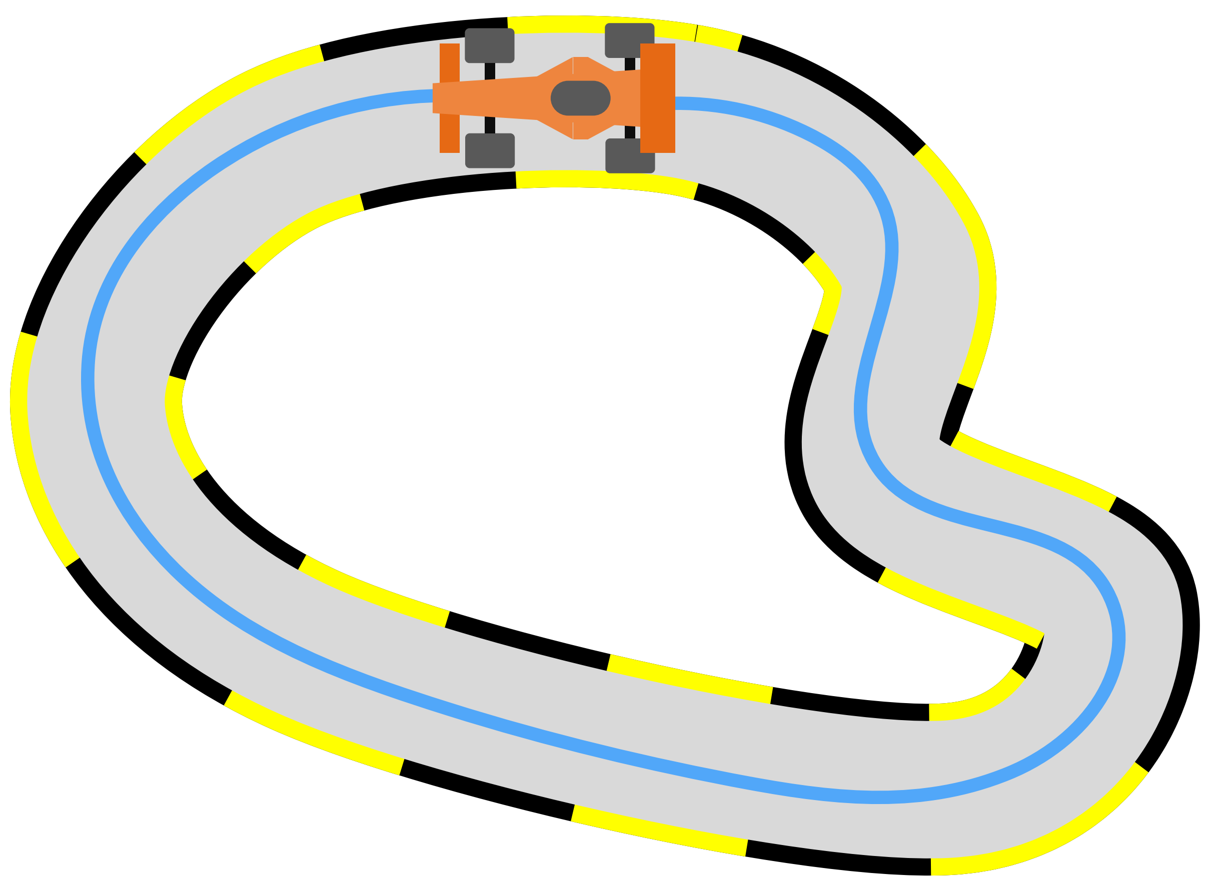


Expert runs away after demonstrations

The Big Problem with BC

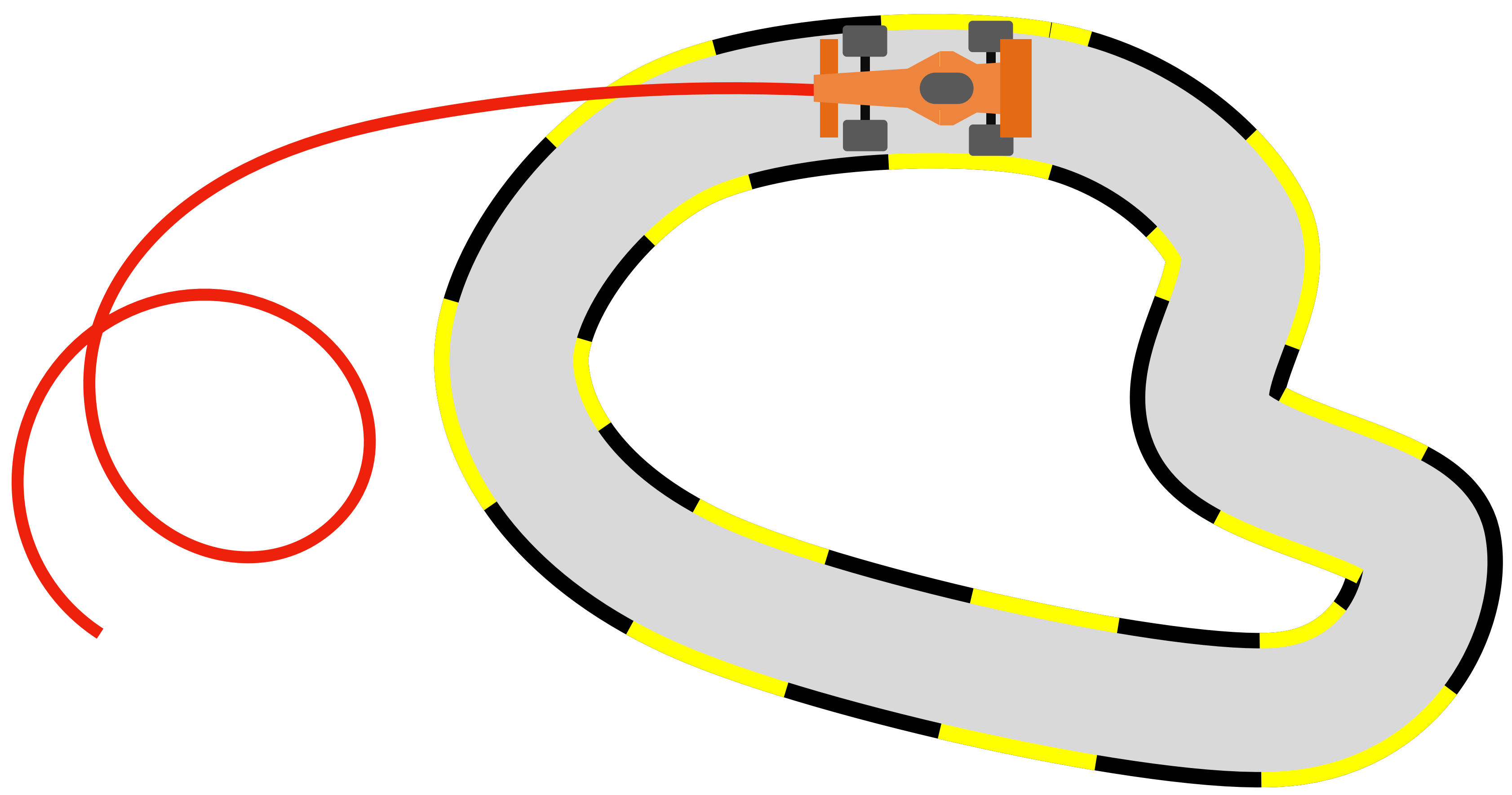
Train

$$\sum_{t=0}^{T-1} \mathbb{E}_{s_t \sim d_t^{\pi^\star}}[\ell(s_t, \pi(s_t))]$$



Test

$$\sum_{t=0}^{T-1} \mathbb{E}_{s_t \sim d_t^{\pi}}[\ell(s_t, \pi(s_t))]$$



The Goal

$$\sum_{t=0}^{T-1} \mathbb{E}_{s_t \sim d_t^{\pi}}[\ell(s_t, \pi(s_t))]$$

Can we bound this to $O(\epsilon T)$?

DAgger (Dataset Aggregation)

Initialize with a random policy π_1 # Can be BC

Initialize empty data buffer $\mathscr{D} \leftarrow \{\}$

For $i = 1, \ldots, N$

Execute policy π_i in the real world and collect data

$$\mathscr{D}_i = \{s_0, a_0, s_1, a_1, \ldots\}$$ # Also called a rollout

Query the expert for the optimal action on learner states

$$\mathscr{D}_i = \{s_0, \pi^\star(s_0), s_1, \pi^\star(s_1), \ldots\}$$

Aggregate data $\mathscr{D} \leftarrow \mathscr{D} \cup \mathscr{D}_i$

Train a new learner on this dataset $\pi_{i+1} \leftarrow \text{Train}(\mathscr{D})$

Select the best policy in $\pi_{1:N+1}$