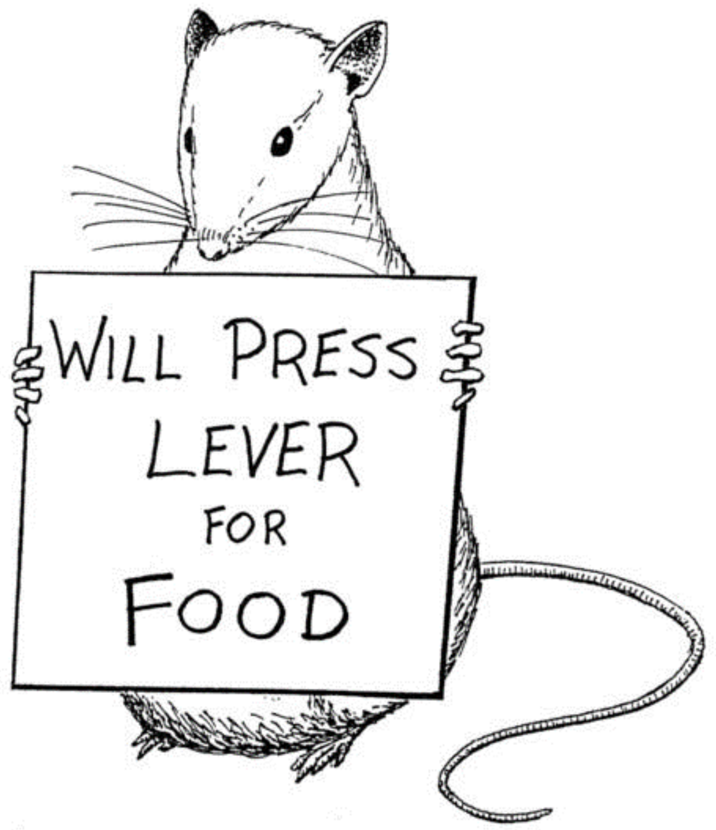
CS 277: Control and Reinforcement Learning

Winter 2021

Lecture 2: Imitation Learning

Roy Fox
Department of Computer Science
Bren School of Information and Computer Sciences
University of California, Irvine



What is imitation learning?

- How can we teach an agent to perform a task?
- Often there is an "expert" that already knows how to perform the task
  - A human operator who controls a robot
  - A black-box artificial agent that we can observe but not copy
  - An agent with different embodiment

- The expert can demonstrate the task to create a training dataset $\mathcal{D} = \{\xi_i\}_i$
  - Each demonstration is a trajectory $\xi = s_0, a_0, s_1, a_1, \ldots$

Today's lecture

Behavior Cloning

Advanced IL methods

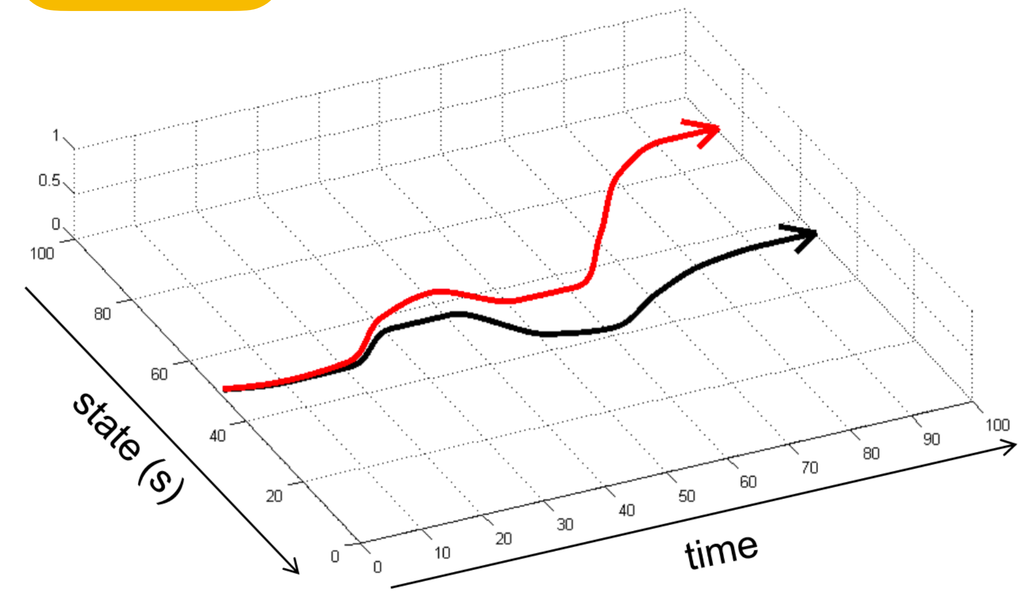
Hierarchical IL

Behavior Cloning (BC)

- The simplest IL is just supervised learning:
  - Break trajectories into examples (s_t, a_t)
  - Learn a function $\pi : s \mapsto a$, or a distribution $\pi(a \mid s)$
- One possible loss: negative log-likelihood $\mathscr{L} = -\sum_{(s,a)\in\mathscr{D}} \log \pi(a \mid s)$

The impact of inaccurate dynamics

$$\cdots \to s_{t-1} \to s_t \to s_{t+1} \to \cdots$$



- Errors in learning are unavoidable

- What impact do they have on sequential behavior?

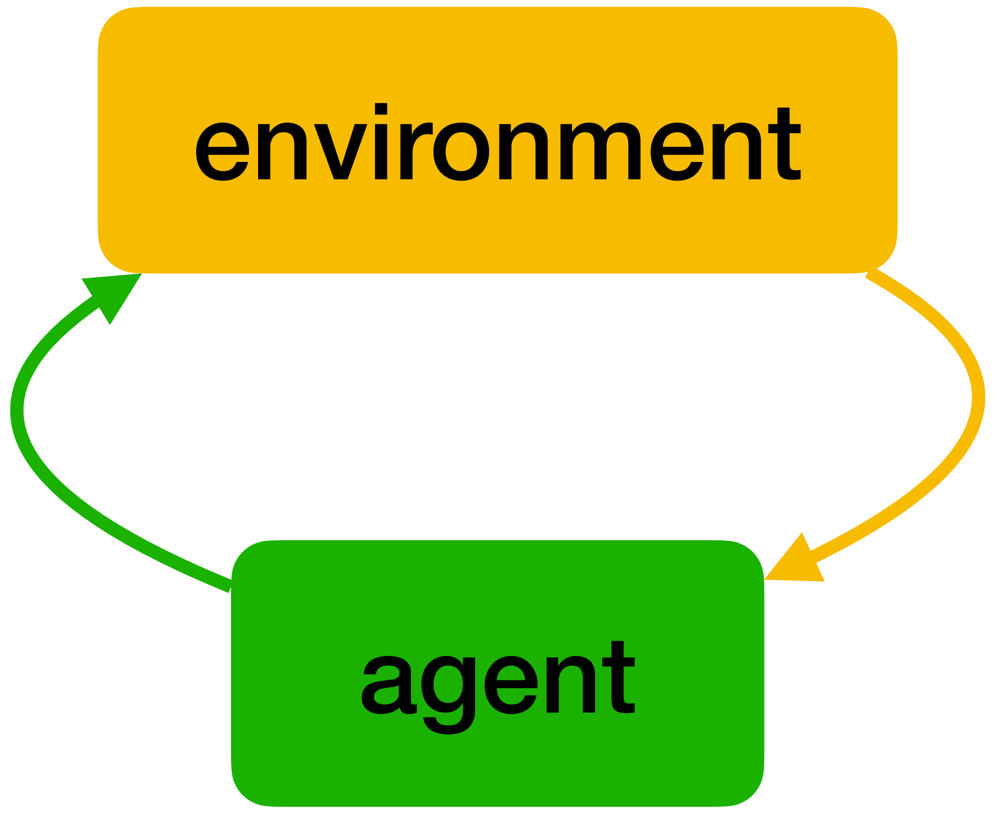
- Bounded one-step error in a dynamical model $\sum_{s'} \left| p_1(s' \mid s) - p_2(s' \mid s) \right| \leq \epsilon$

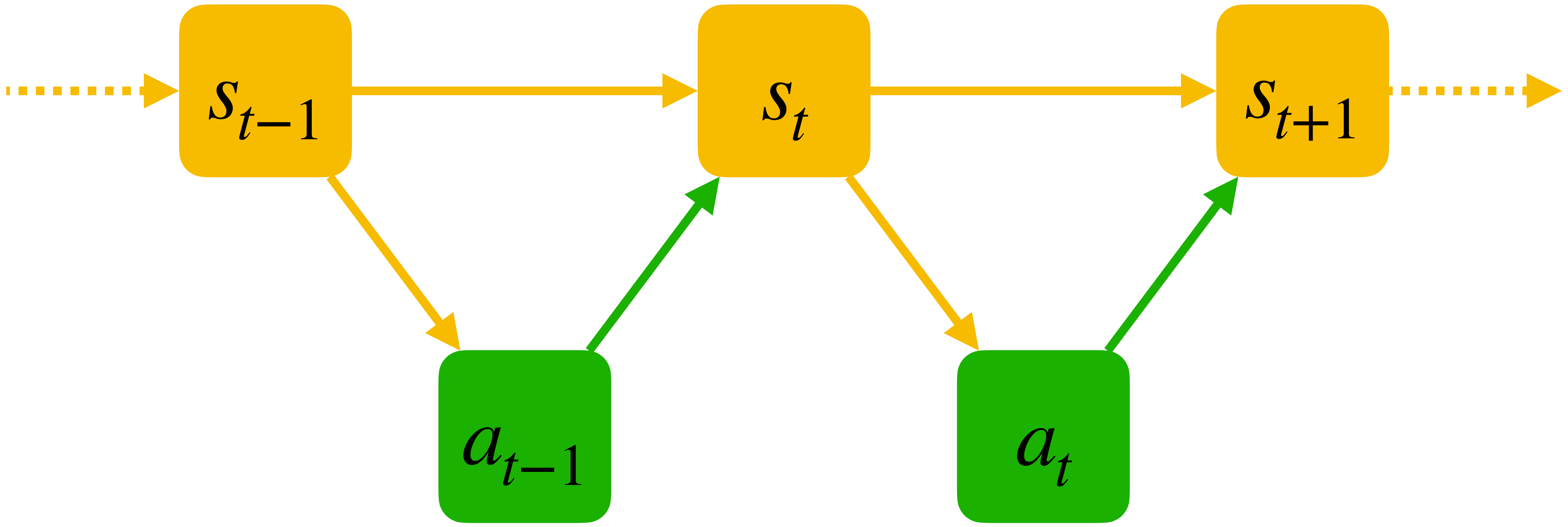
  - Can lead to growing error over time $\sum_{s_t} \left| p_1(s_t) - p_2(s_t) \right| \leq \epsilon t$

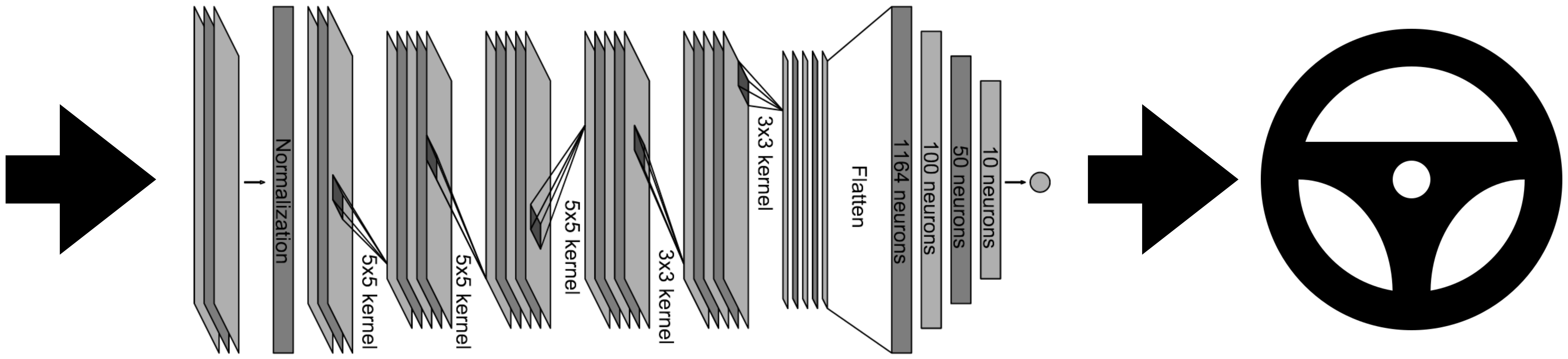
- The same holds for inaccurate learned π, compared to the teacher π^*

Image: Sergey Levine

A policy is a (stochastic) function



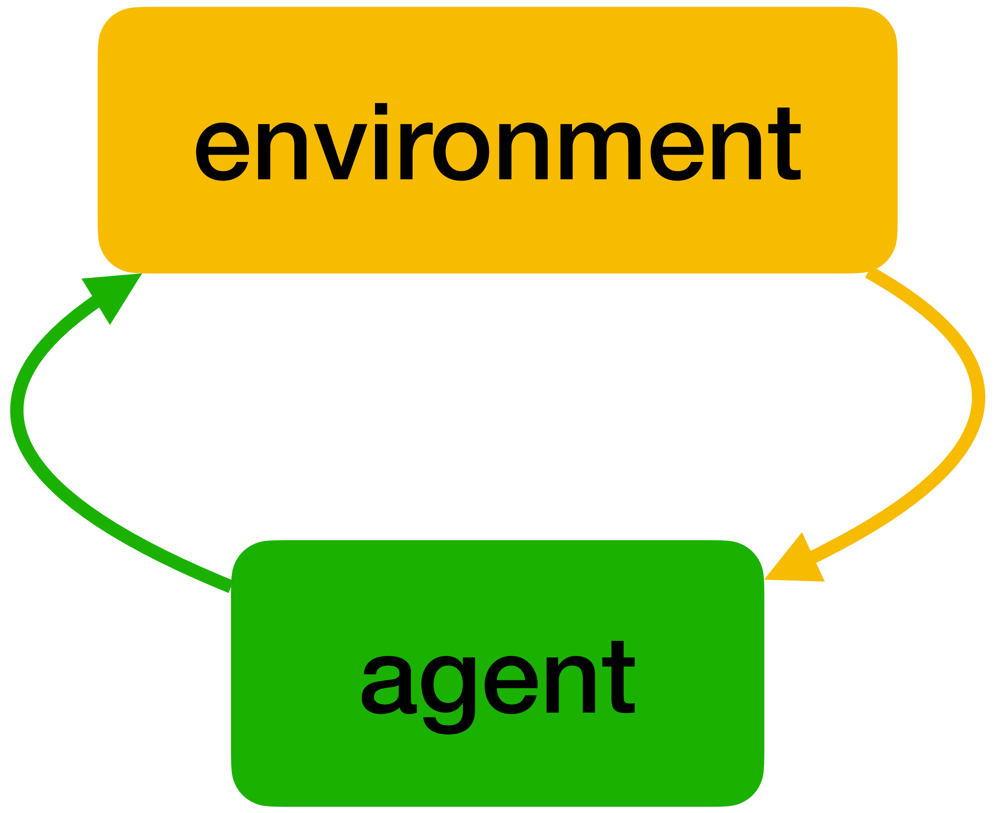


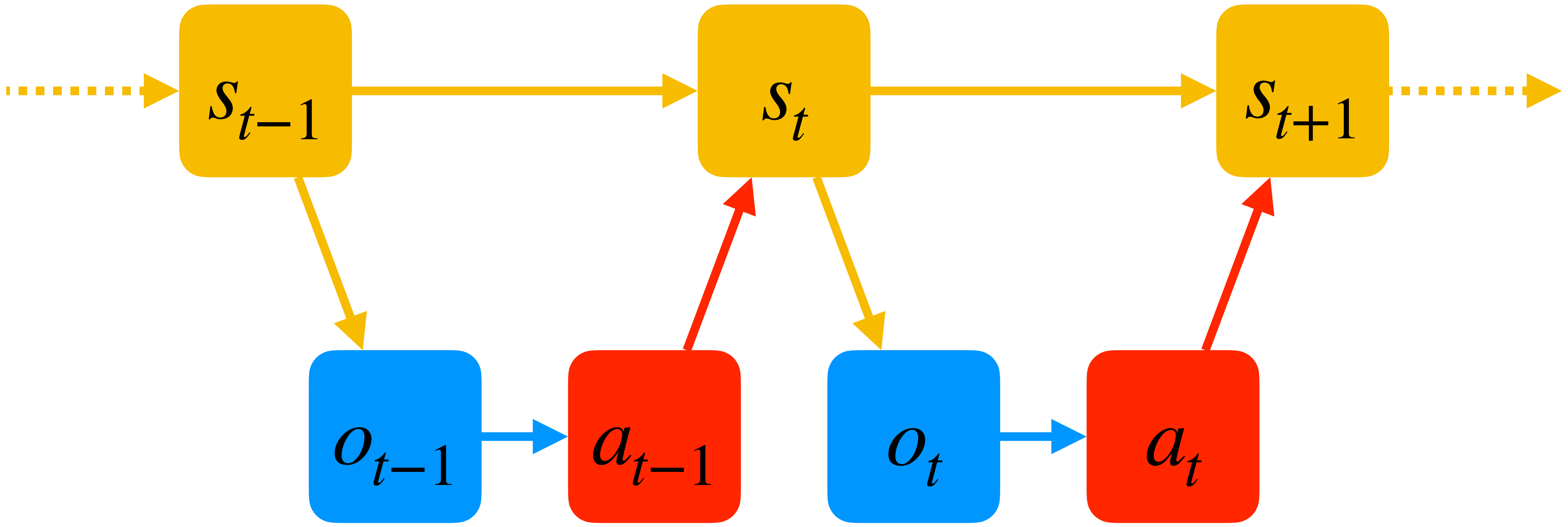


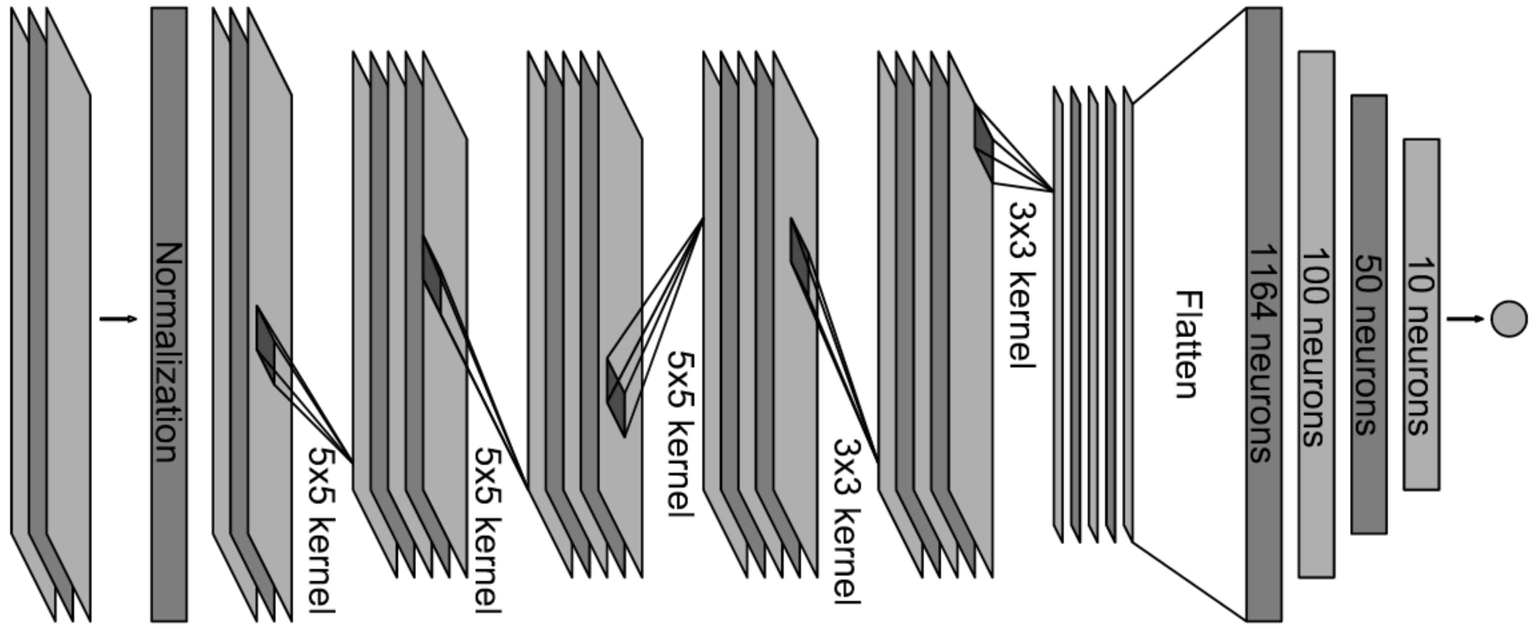
$\pi(a_t \mid s_t)$

Image: Bojarski et al. 2016

A policy is a (stochastic) function







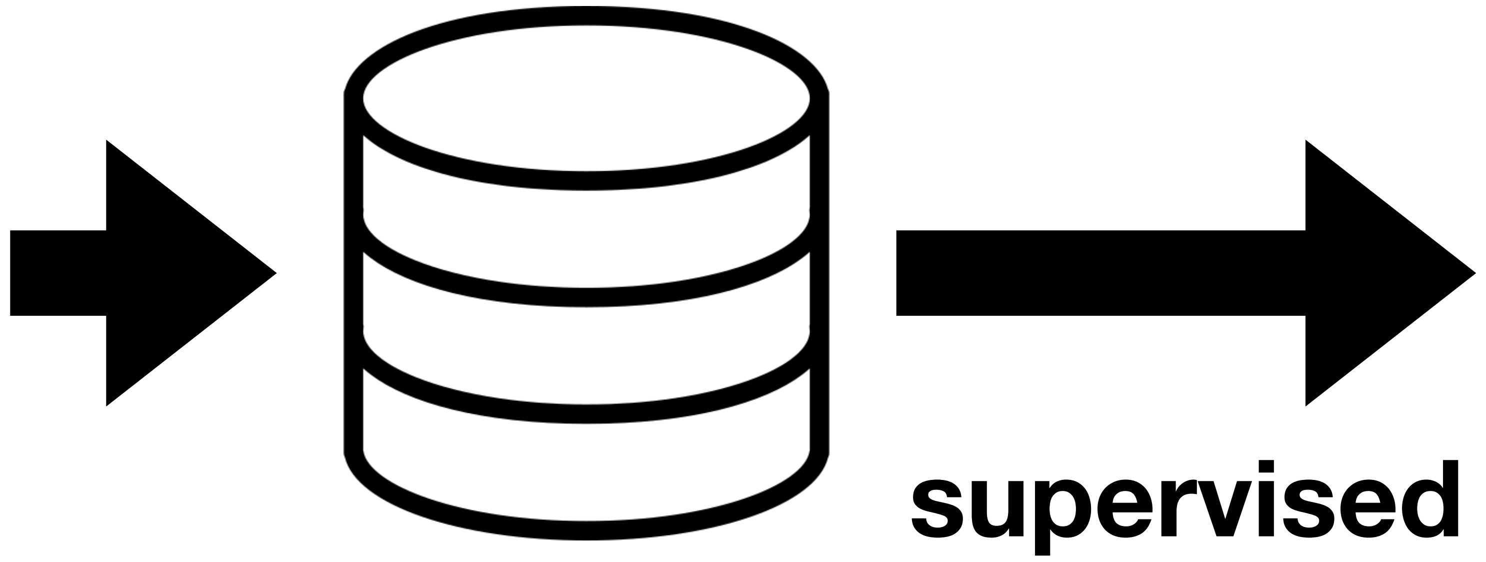
$\pi(a_t \mid o_t)$

observation

action

Inaccuracy in BC

observations
+
actions

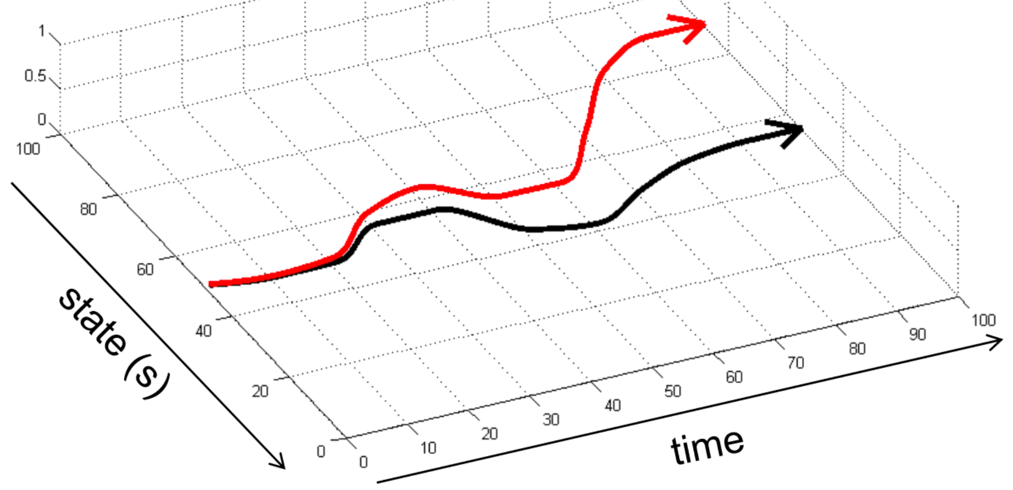


training
data

supervised
learning

$\pi_\theta(a_t | o_t)$

- The state transition distribution is linear in the policy
$$p_\pi(s_{t+1} | s_t) = \sum_{o_t, a_t} p(o_t | s_t)\pi(a_t | o_t)p(s_{t+1} | s_t, a_t)$$

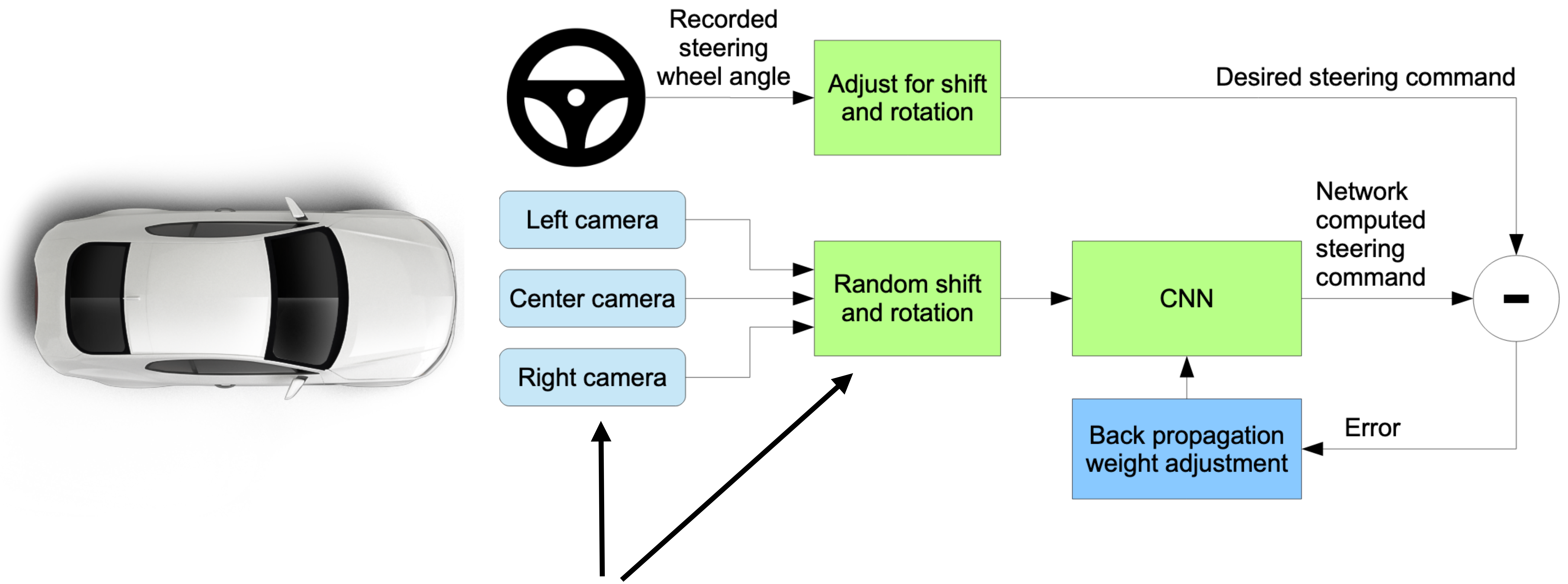


- If the policy approximates the teacher $\pi_\theta(a_t | o_t) \approx \pi^*(a_t | o_t)$
    - The dynamics will also approximate teacher behavior $p_{\pi_\theta}(s_{t+1} | s_t) \approx p_{\pi^*}(s_{t+1} | s_t)$
- But errors do accumulate over time
    - May reach states not seen in the training dataset

But wait...

Video: NVIDIA

How did they do it?



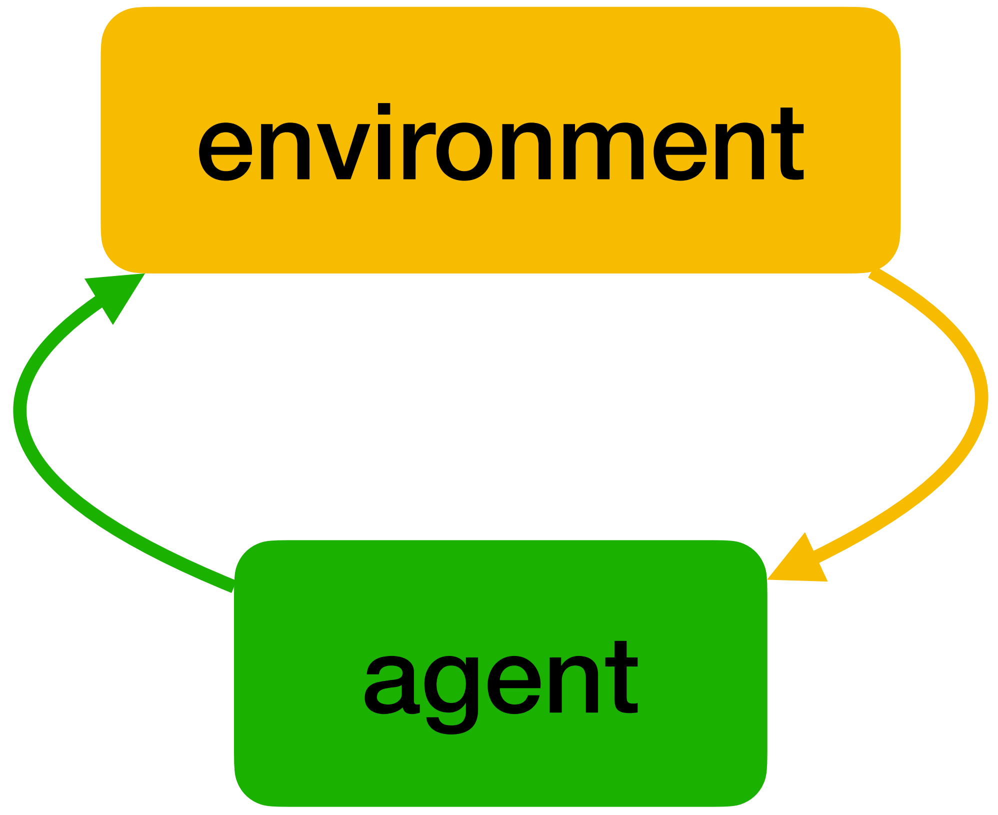
augmented data to better cover test distribution

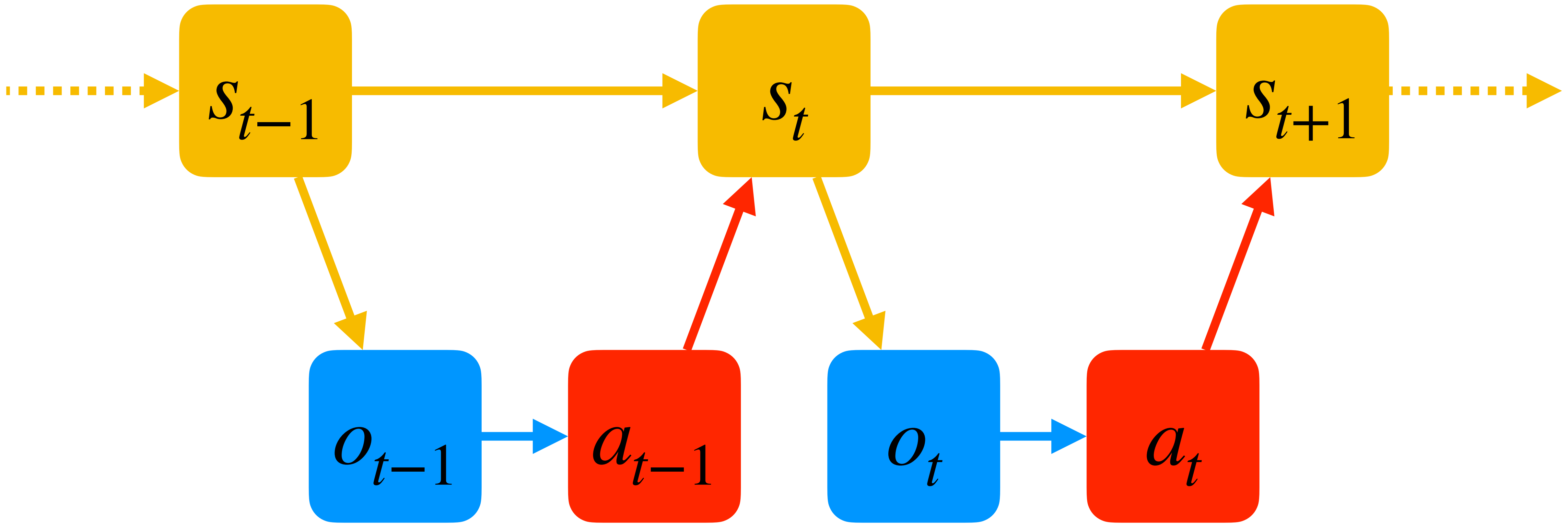
Image: Bojarski et al. 2016

IL challenges: modeling other agents is hard

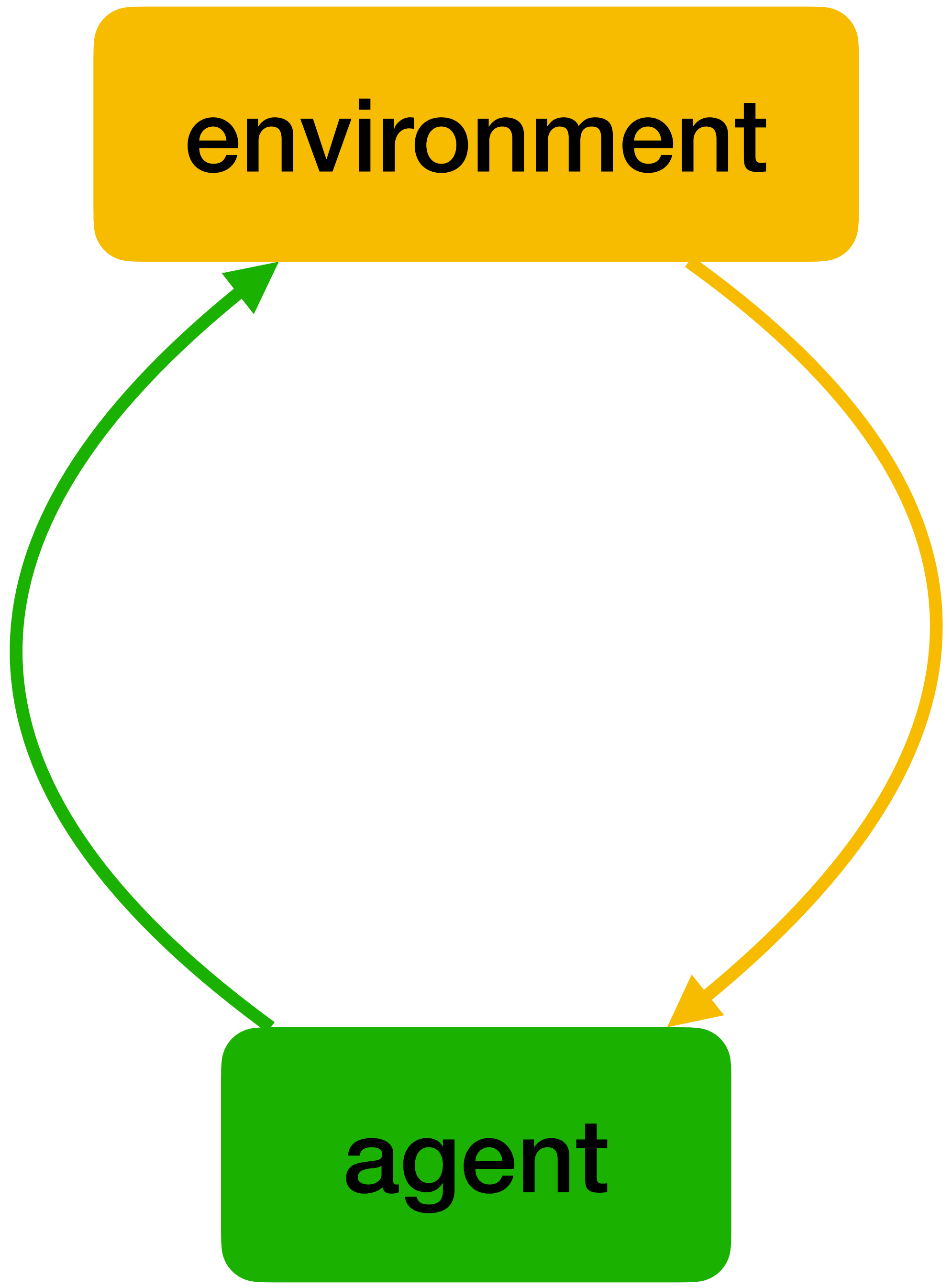
- Are the agent and human observations different ($o_t \neq o_t^{\mathrm{H}}$)?
- Is the state partially observable ($o_t \neq s_t$)?
  - $p(o_{t+1} | o_t, a_t) \neq p(o_{t+1} | o_0, a_0, \ldots, o_t, a_t)$, generally requiring $\pi_\theta(a_t | o_0, a_0, \ldots, o_t)$
  - Can use RNNs $f_\theta : (h_{t-1}, a_{t-1}, o_t) \mapsto h_t$, or other memory models
    - But memory state is latent in demonstrations
  - Modeling memory is hard → prior structure may help
- Is there sufficient data? Demonstrating is a burden!
- Are demonstrations consistent? Humans are fallible + some supervision is hard

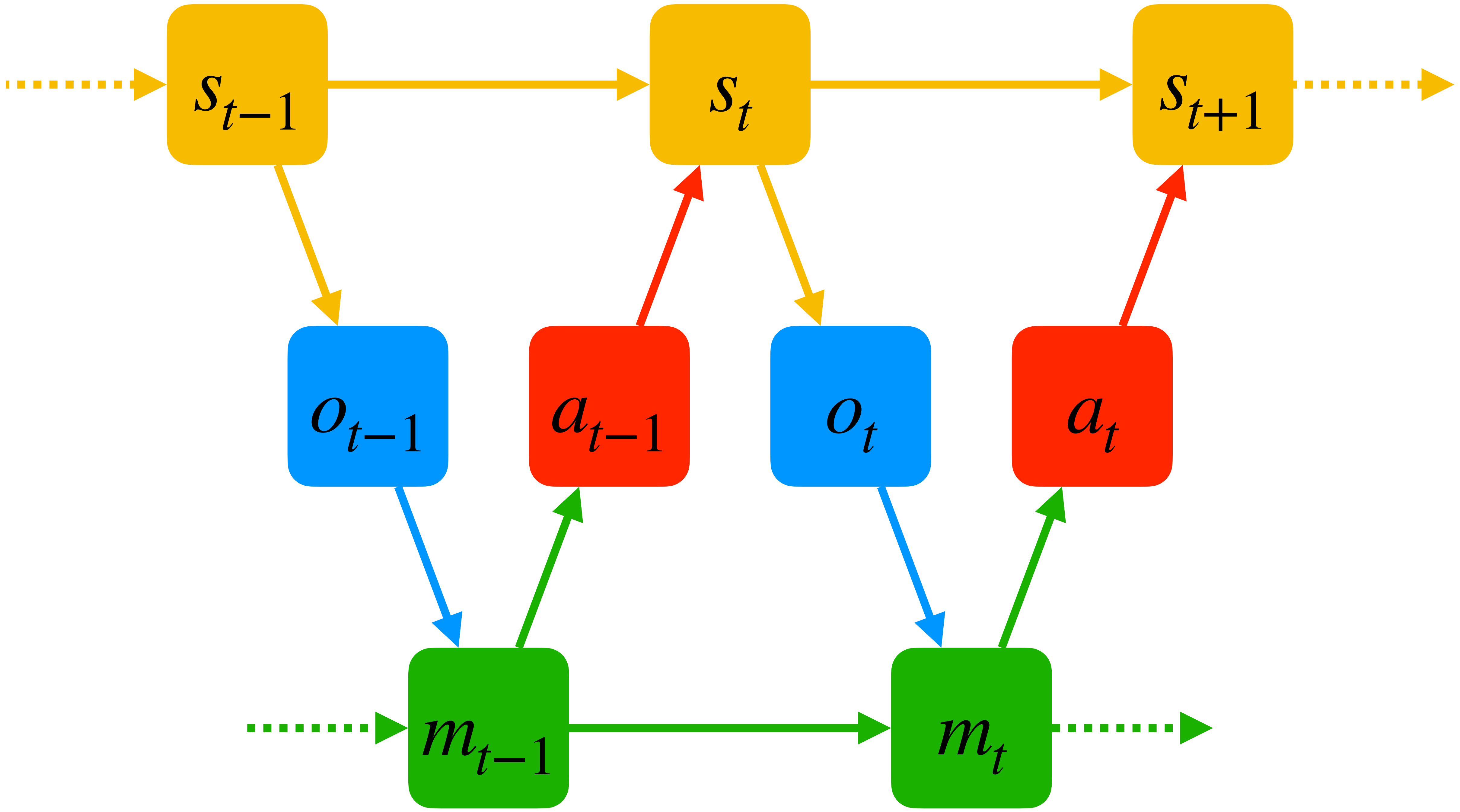
Modeling memory





Modeling memory





$$\pi_\theta(m_t, a_t \mid m_{t-1}, o_t)$$

Today's lecture

Behavior Cloning

Advanced IL methods

Hierarchical IL

DAgger: Dataset Aggregation

- Can we collect demonstration data for $p_{\pi_\theta}(o_t)$?

Algorithm 1 DAgger

Collect dataset $\mathcal{D}$ of teacher demonstrations
$(o_0, a_0^*, o_1, a_1^*, \ldots) \sim p_{\pi^*}$
Train π_θ on $\mathcal{D}$
Execute π_θ to get $(o_0, a_0, \ldots) \sim p_{\pi_\theta}$
Ask teacher to label $a_t^* | o_t \sim \pi^*$
Aggregate $(o_0, a_0^*, o_1, a_1^*, \ldots)$ into $\mathcal{D}$
Repeat!

DAgger demo

Video: Stéphane Ross

DAgger: Dataset Aggregation

- Can we collect demonstration data for $p_{\pi_\theta}(o_t)$?

Algorithm 1 DAgger

Collect dataset $\mathcal{D}$ of teacher demonstrations

$$(o_0, a_0^*, o_1, a_1^*, \dots) \sim p_{\pi^*}$$

Train π_θ on $\mathcal{D}$

Execute π_θ to get $(o_0, a_0, \dots) \sim p_{\pi_\theta}$

$\boxed{\text{Ask teacher to label } a_t^* | o_t \sim \pi^*}$ **but how? challenging...**

Aggregate $(o_0, a_0^*, o_1, a_1^*, \dots)$ into $\mathcal{D}$

Repeat!

- DAgger can reduce the imitation loss from $O(\epsilon T^2)$ to $O(\epsilon T)$

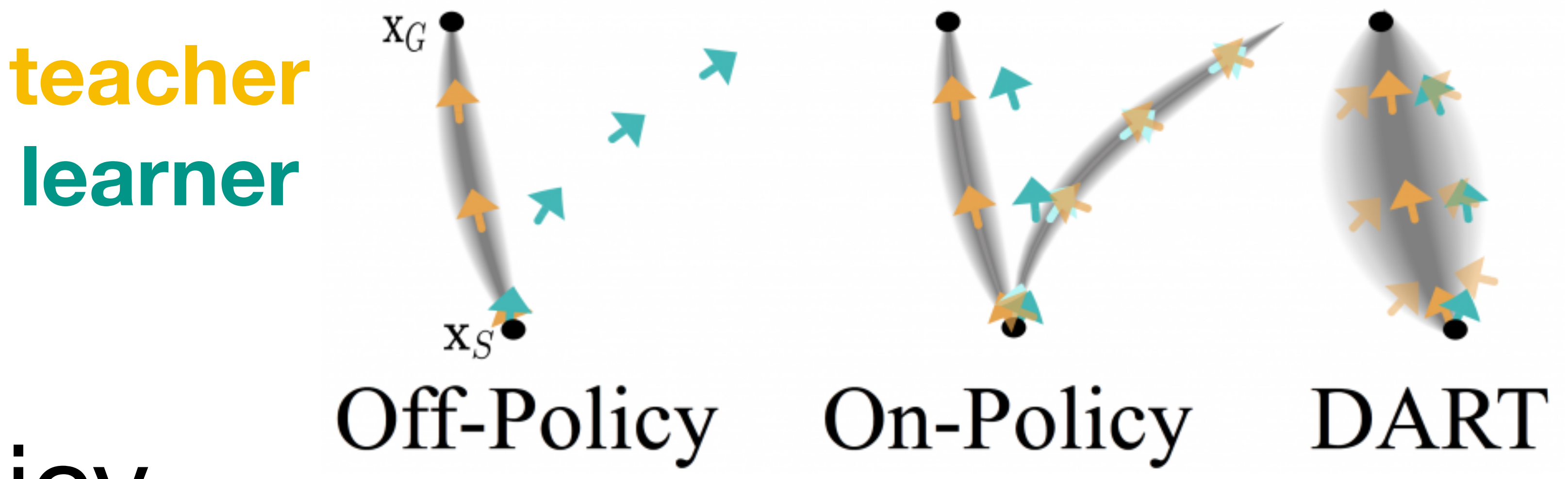
Goal-conditioned Behavior Cloning

- Can we train one policy to reach multiple goals? $\pi_\theta(a_t | s_t, g)$
  - Assume goal = state that the agent should reach
- How can we know the goal in demonstrations $\xi = s_0, a_0, s_1, a_1, \ldots$?
  - Require manual labeling?
- Hindsight: take each s_t as the goal of the trajectory leading to it

$$s_0, a_0, \ldots, s_{t-1}, a_{t-1}, s_t = g$$

  - Supervised learning of $\pi(a | s, g)$ from data points $(s_t, a_t, s_{t'})$ for $t' > t$

DART: Disturbances Augmenting Robot Training

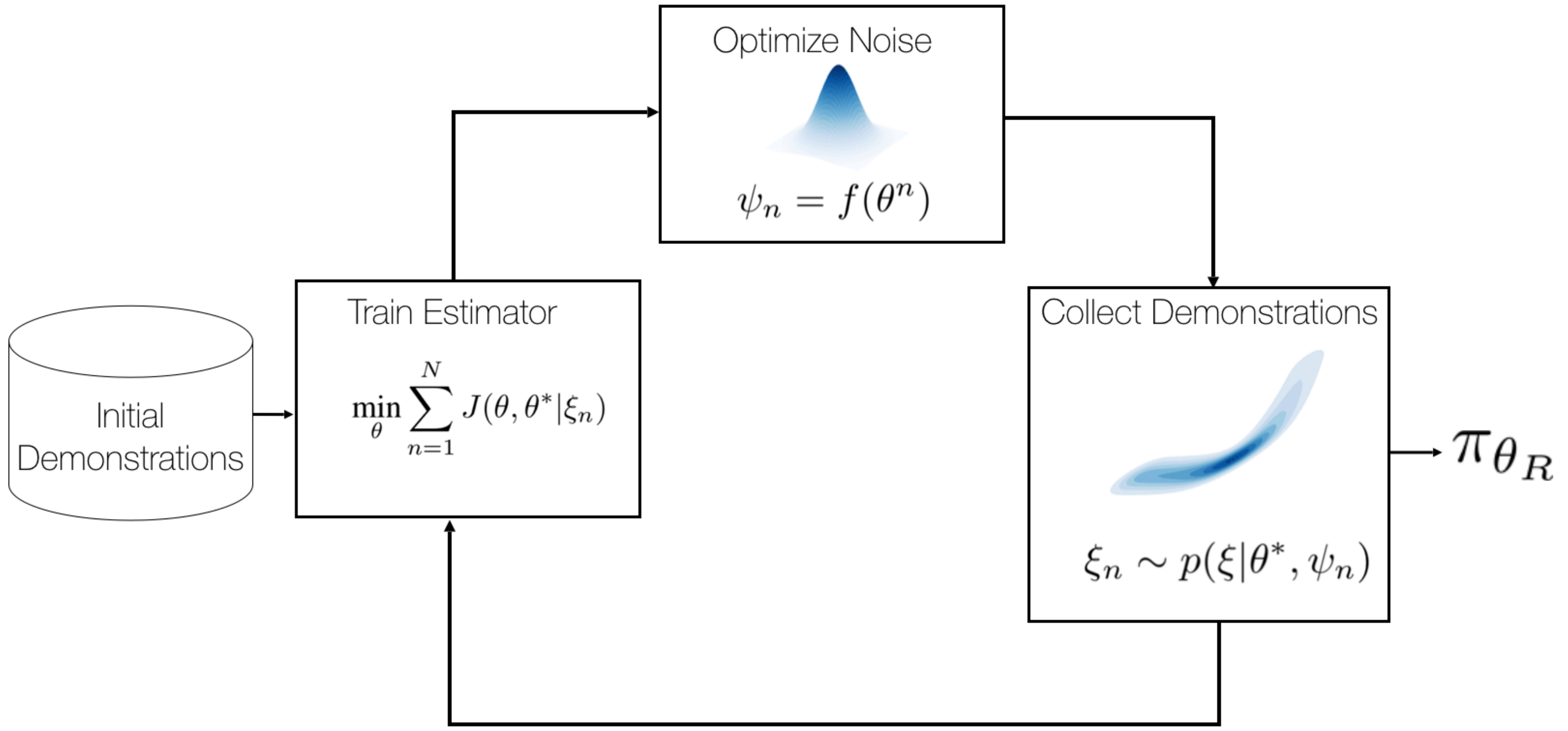


- Off-policy vs. on-policy
  - On-policy = data comes from the learner's current policy
  - Off-policy = data comes from another policy (another agent or past learner)
- In off-policy IL (e.g. BC) learner may go off the teacher's support
- In on-policy IL (e.g. DAgger) learner initially goes off, until corrected
- DART: increase the data support by injecting noise during demonstrations
  - Force teacher into slight-error states, to see how they are fixed

Image: Laskey et al. 2017

DART

- Noise = perturbation of actions
  - $$\tilde{p}(s' \mid s, a) = \sum_{\tilde{a}} q(\tilde{a} \mid a) p(s' \mid s, \tilde{a})$$
  - In continuous actions: $\tilde{a} = a + \epsilon; \quad \epsilon \sim \mathcal{N}(0, \Sigma)$
- Repeat:



  - Collect teacher demonstrations
  - Train agent with BC
  - Optimize noise to force teacher towards agent distribution

Image: Michael Laskey

Grasping task

Behavior Cloning

DART

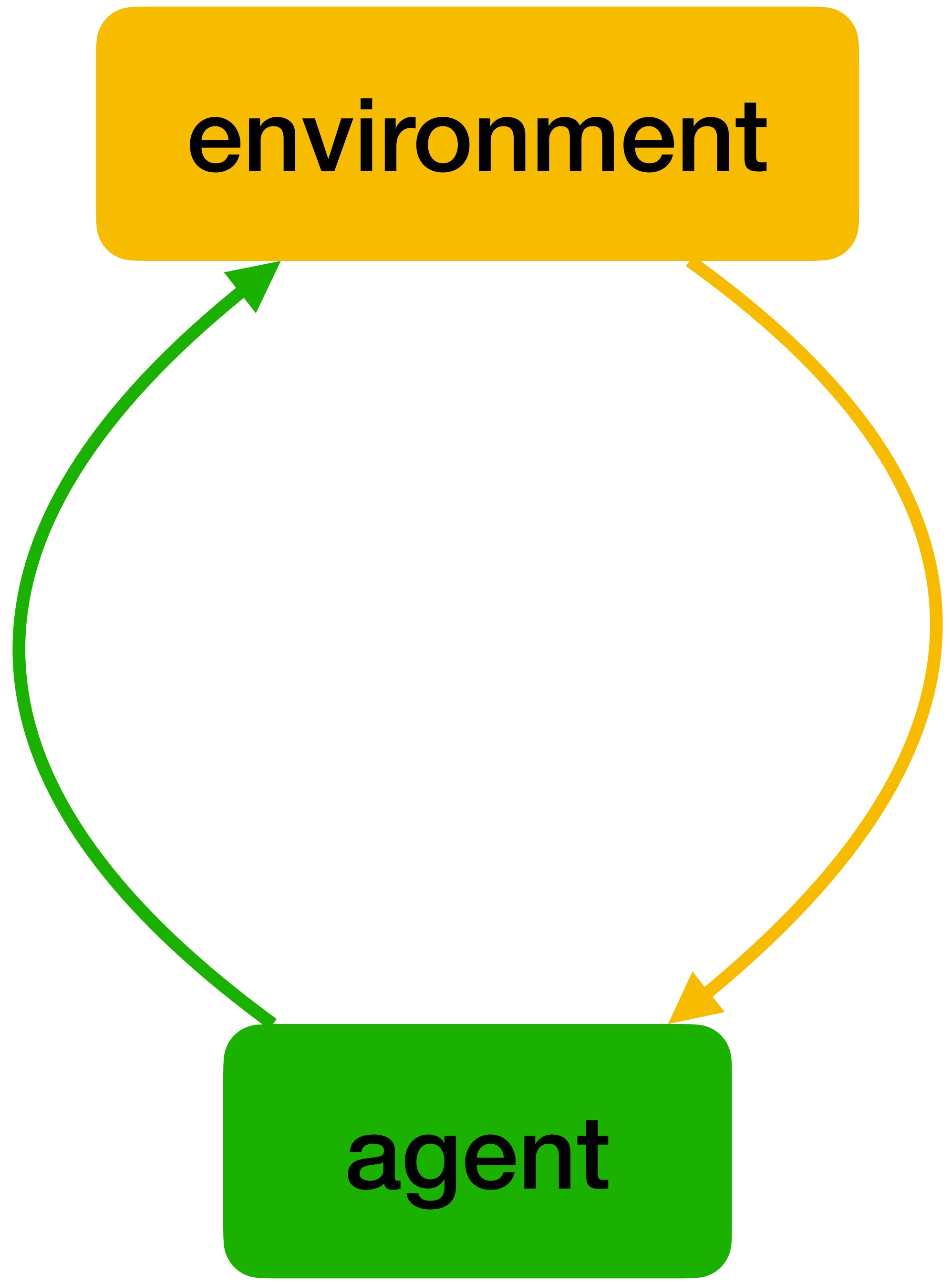
Today's lecture

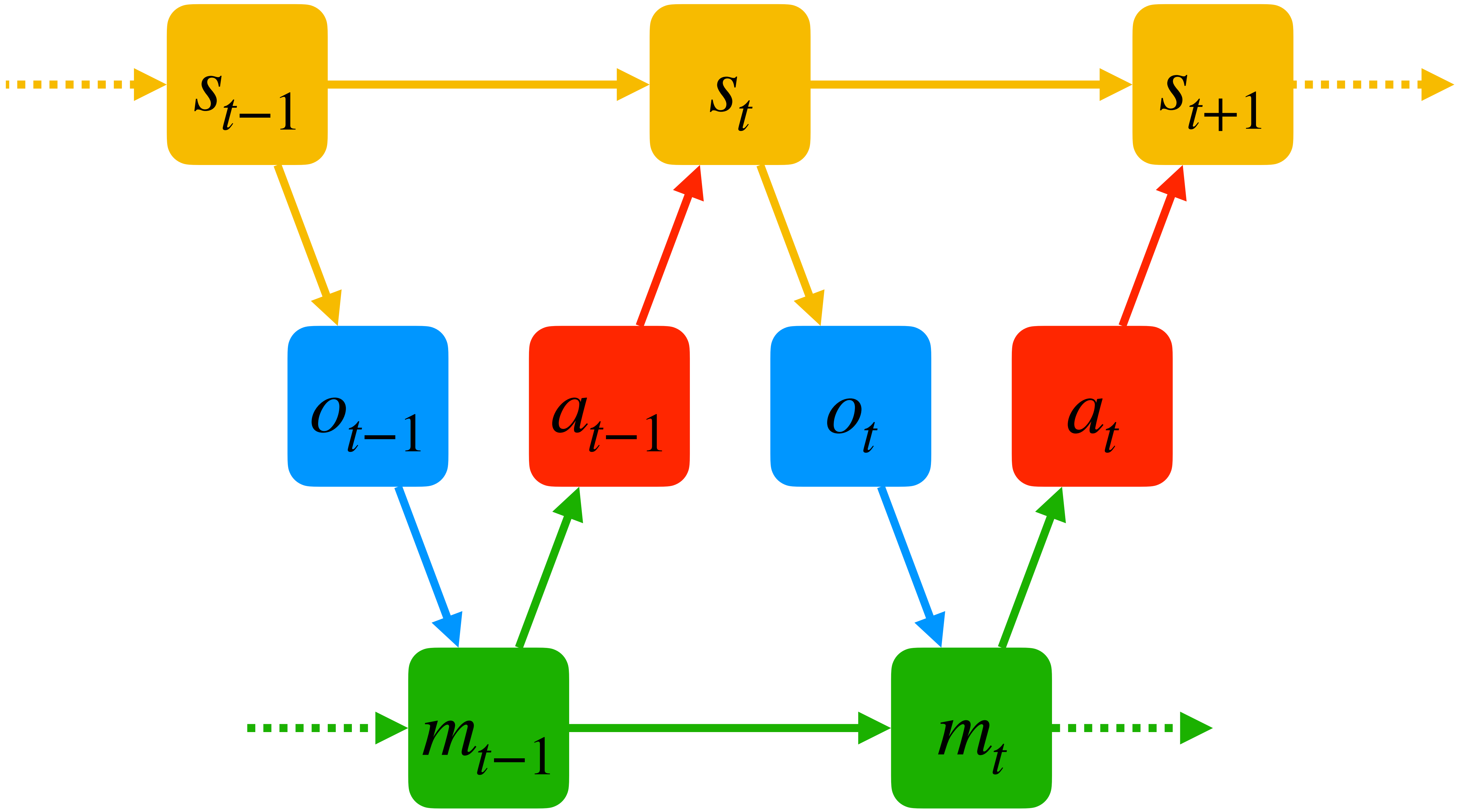
Behavior Cloning

Advanced IL methods

Hierarchical IL

Modeling memory

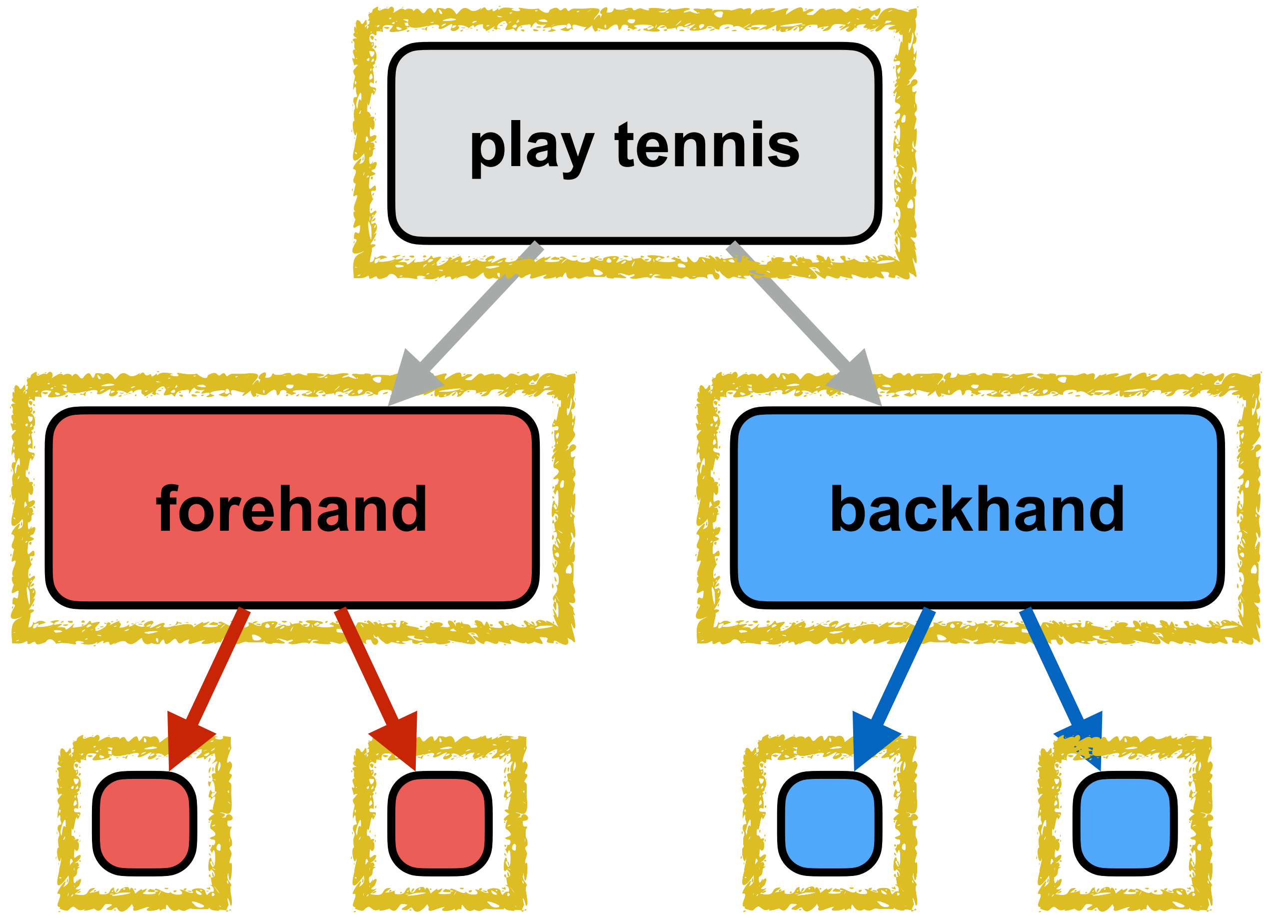


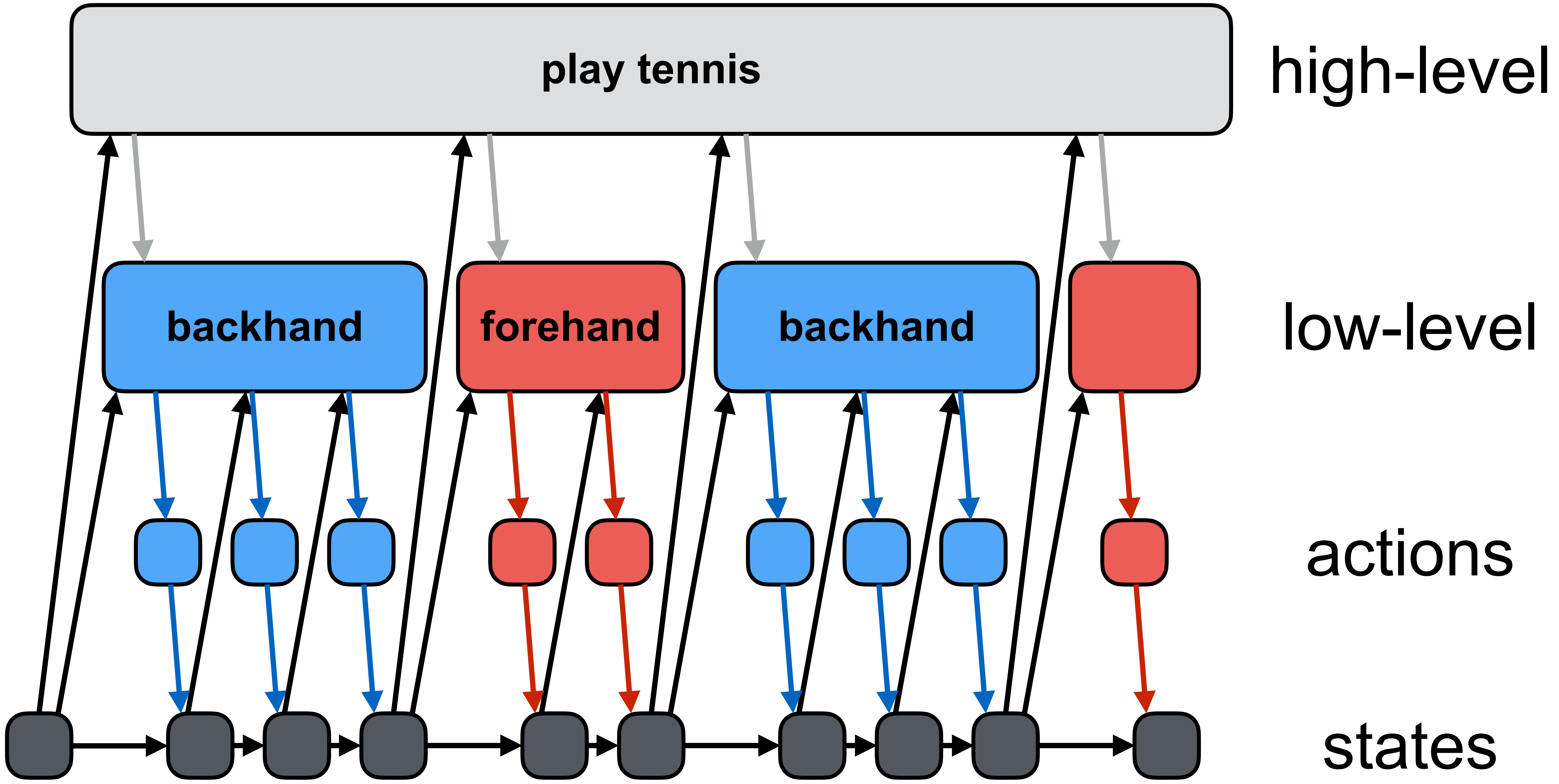


$$\pi_\theta(m_t, a_t \,|\, m_{t-1}, o_t)$$

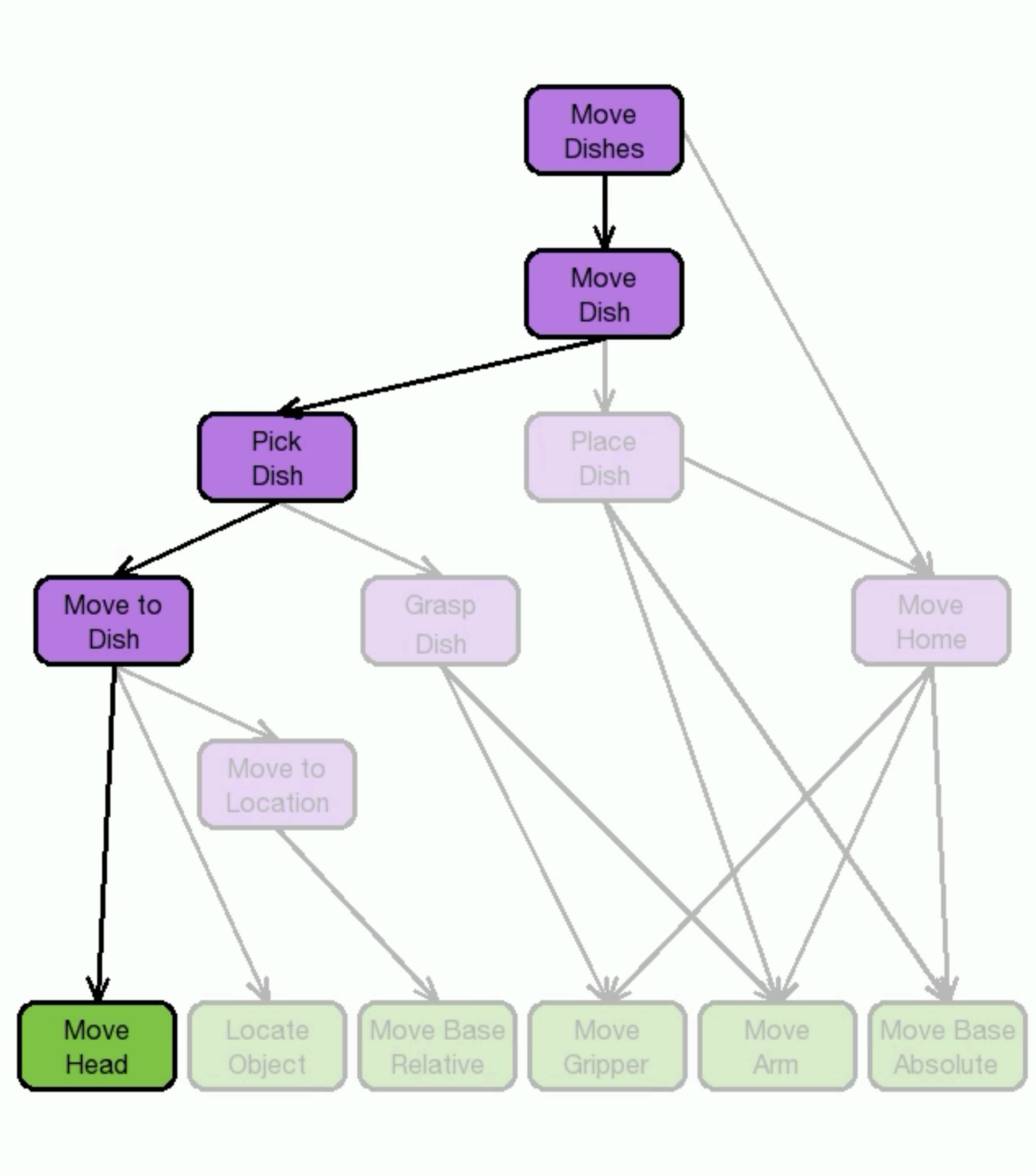
- What is a good structure for memory?

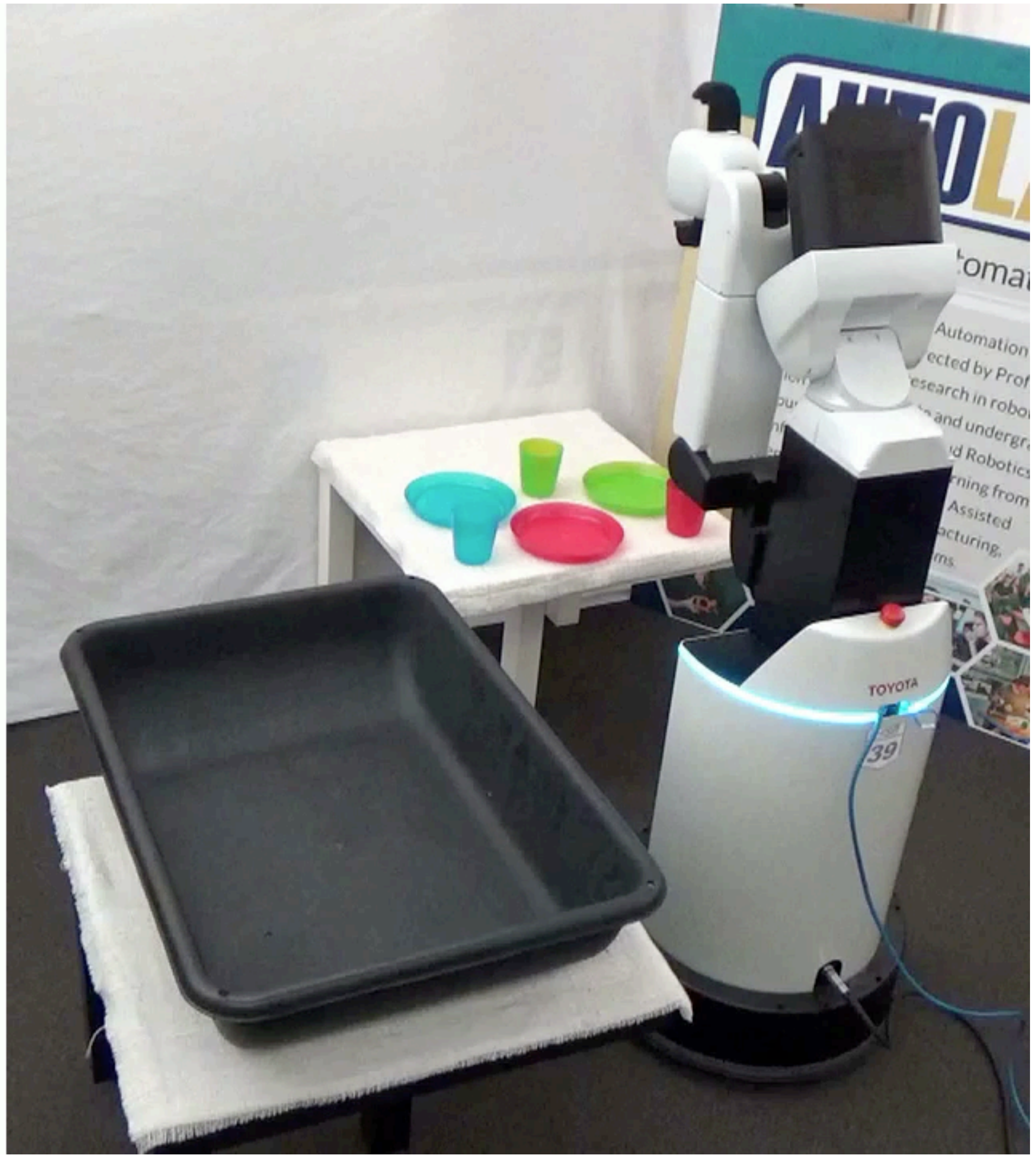
HVIL: Hierarchical Variational Imitation Learning





HVIL: Hierarchical Variational Imitation Learning





Video: Fox et al. 2019

Imitation Learning as inference

- Behavior Cloning with cross-entropy loss maximizes

$$\log p_{\pi_\theta}(\mathscr{D}) = \sum_i \log \pi_\theta(a_i \mid o_i) + \text{const} = \log \pi_\theta(a \mid o) + \text{const}$$

- With latent execution structure m we have $\log \pi_\theta(a \mid o) = \log \sum_m \pi_\theta(m, a \mid o)$

- Evidence Lower Bound (ELBO):

$$\log \pi_\theta(a \mid o) \geq \mathbb{E}_{m \mid o, a \sim q_\phi}[\log \pi_\theta(m, a \mid o) - \log q_\phi(m \mid a, o)]$$

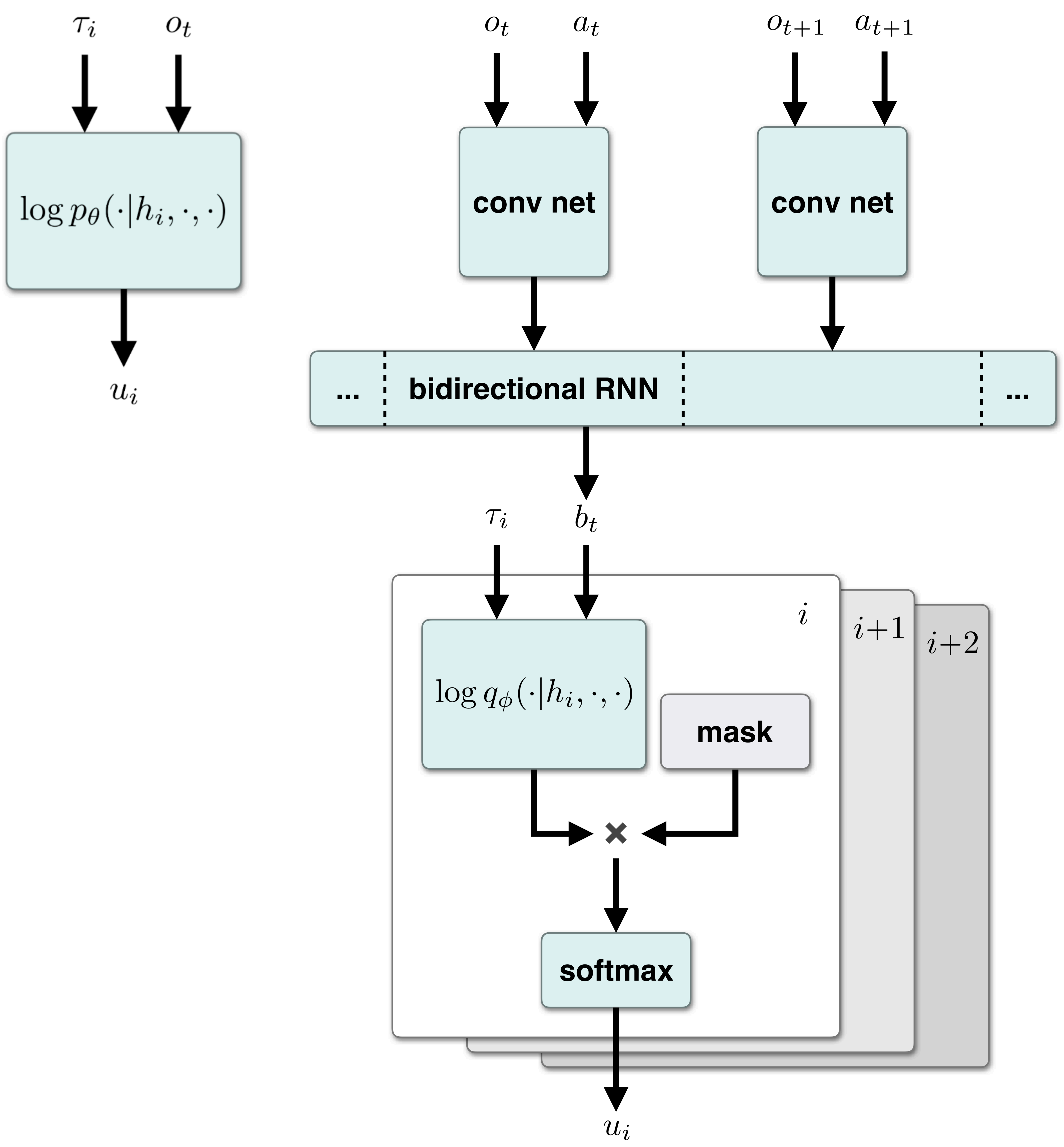
- Inference network $q_\phi(m \mid a, o)$ samples execution structure m
  - which guides training of the agent $\pi_\theta(m, a \mid o)$

Hierarchical Variational Imitation Learning (HVIL)

- Inference network decomposes as

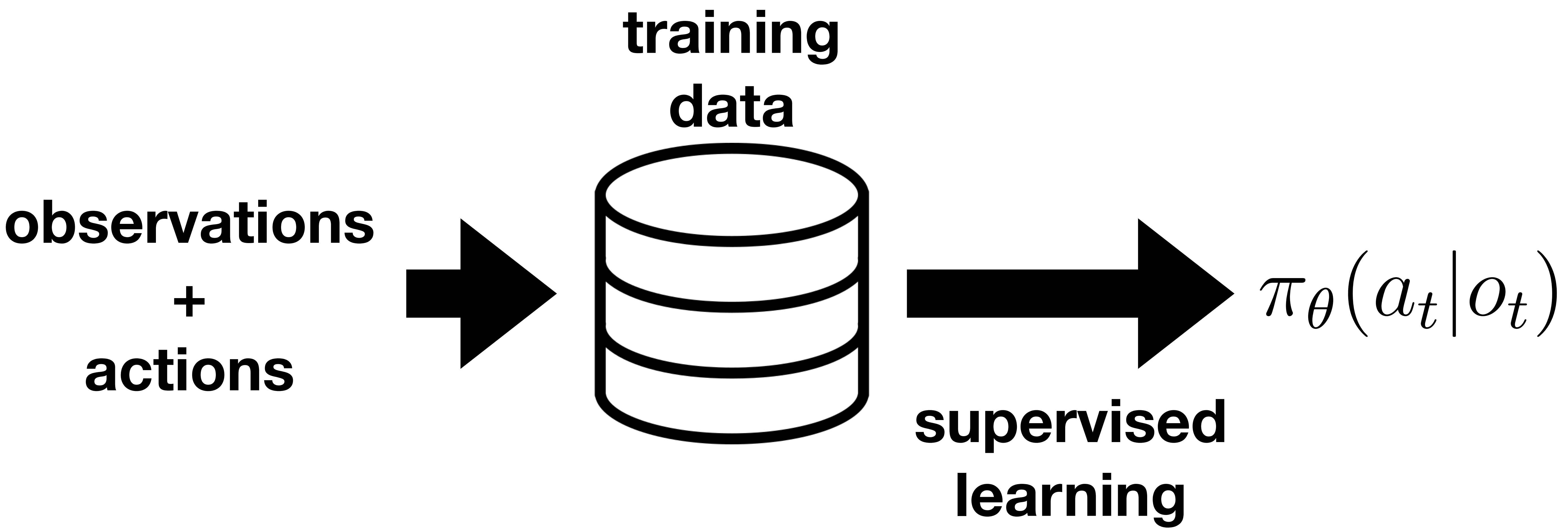
$$q_\phi(m \,|\, a, o) = \prod_i q_\phi(\text{procedure step } i \,|\, a, o)$$

- Bidirectional RNN summarizes demonstration
- into posterior context [Fraccaro et al., NeurIPS 2016]
- Output masked to ensure consistent steps

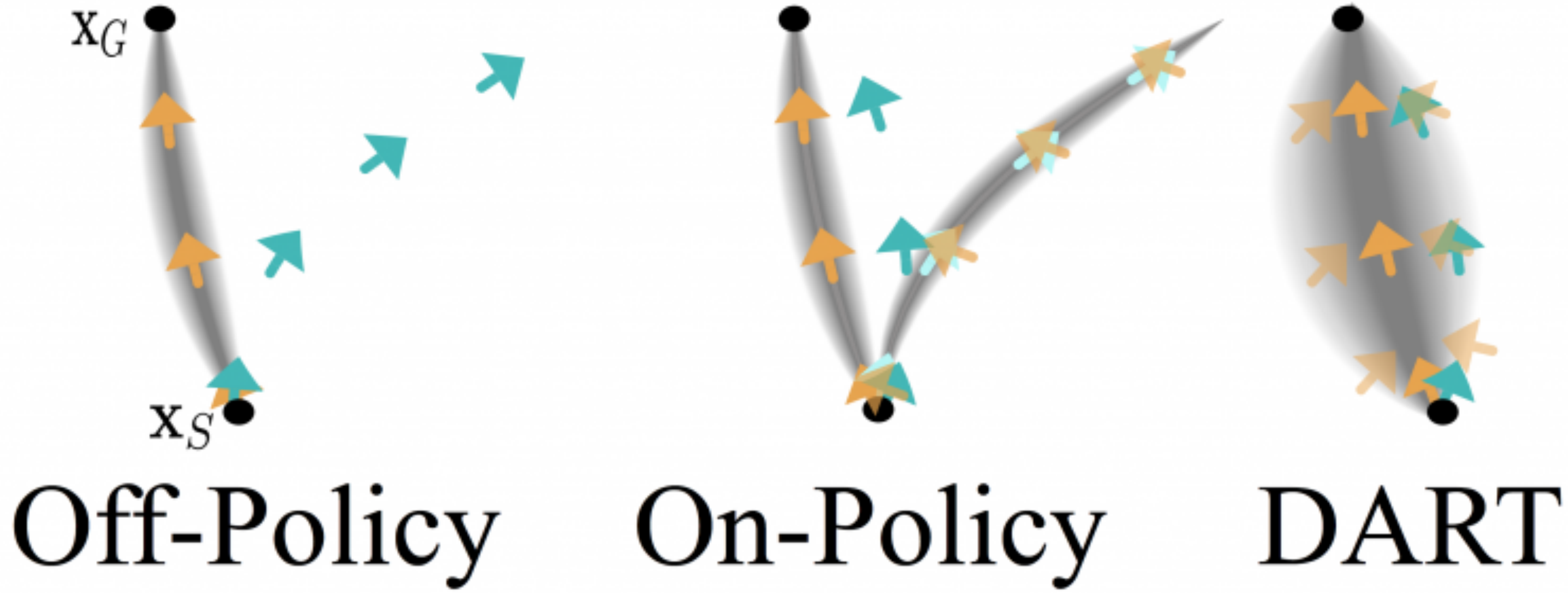


[F., Shin, Paul, Zou, Song, Goldberg, Abbeel, and Stoica, arXiv 2019]

Recap



training data
observations + actions
supervised learning
$\pi_\theta(a_t|o_t)$



x_G
x_S
Off-Policy
On-Policy
DART