CS885 Reinforcement Learning Module 3: July 5, 2020

Imitation Learning

Torabi, F., Warnell, G., & Stone, P. (2018). Behavioral cloning from observation. In *IJCAI* (pp. 4950-4957).

Ho, J., & Ermon, S. (2016). Generative adversarial imitation learning. In *NeurIPS* (pp. 4565-4573).

Imitation Learning

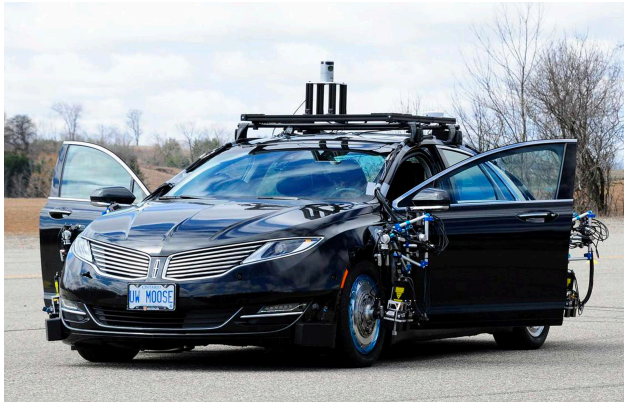
- Behavioural cloning (supervised learning)
- Generative adversarial imitation learning (GAIL)
- Imitation learning from observations
- Inverse reinforcement learning

Motivation

- Learn from expert demonstrations
  - No reward function needed
  - Faster learning

robotics

chatbots



autonomous driving

Behavioural Cloning

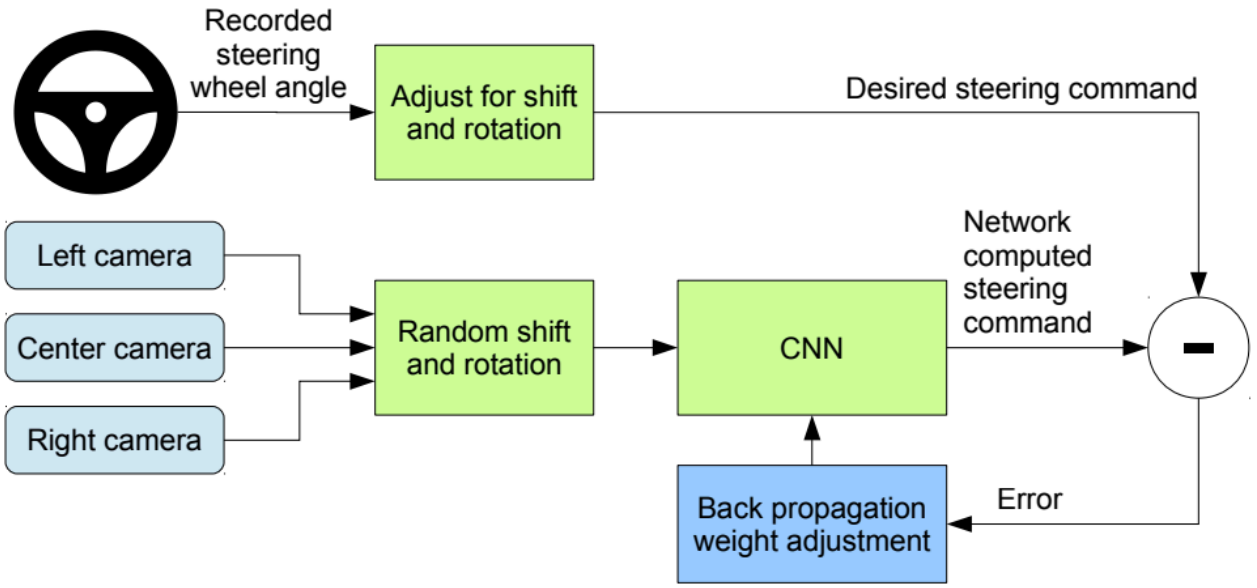
- Simplest form of imitation learning
- Assumption: state-action pairs observable

Imitation learning

- Observe trajectories: $(s_1, a_1), (s_2, a_2), (s_3, a_3), \ldots, (s_n, a_n)$
- Create training set: $S \rightarrow A$
- Train by supervised learning
  - Classification or regression

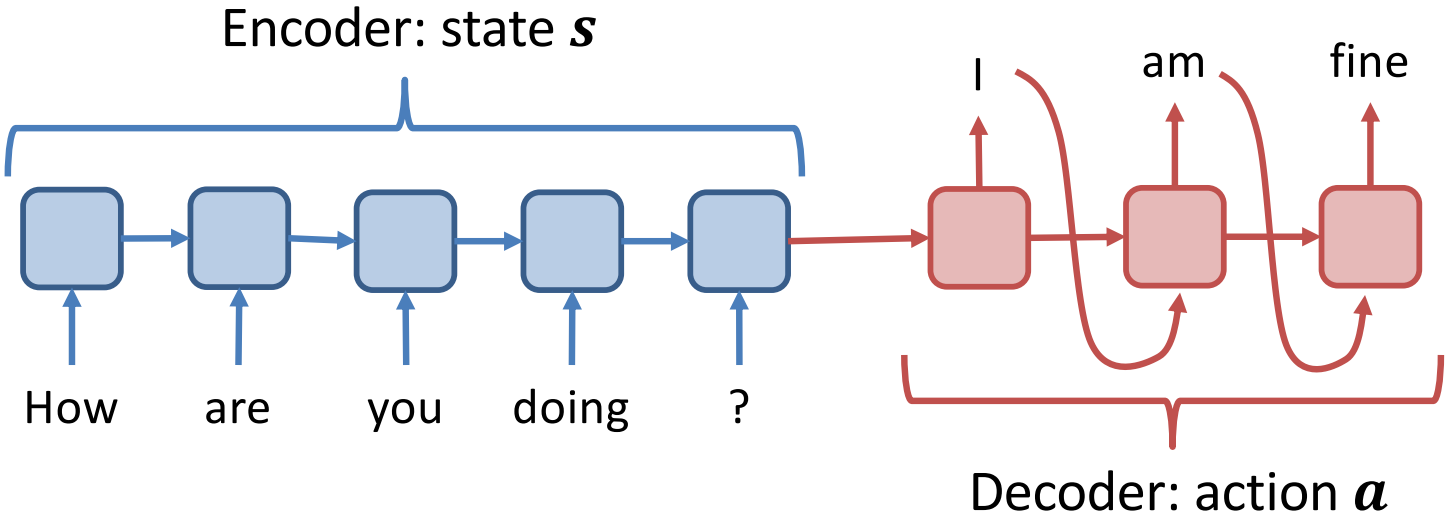
Case Study I: Autonomous driving

- Bojarski et al. (2016) End-to-end learning for self-driving cars



- On road tests:
  - Holmdel to Atlantic Highlands (NJ): autonomous ~98% of the time
  - Garden State Parkway (10 miles): no human intervention

Case study II: conversational agents



$$\text{Objective: } \max_{\mathbf{a}} \Pr(\boldsymbol{a}|\boldsymbol{s}) = \prod_i \Pr(a_i|a_{i-1}, \ldots, a_1, \boldsymbol{s})$$

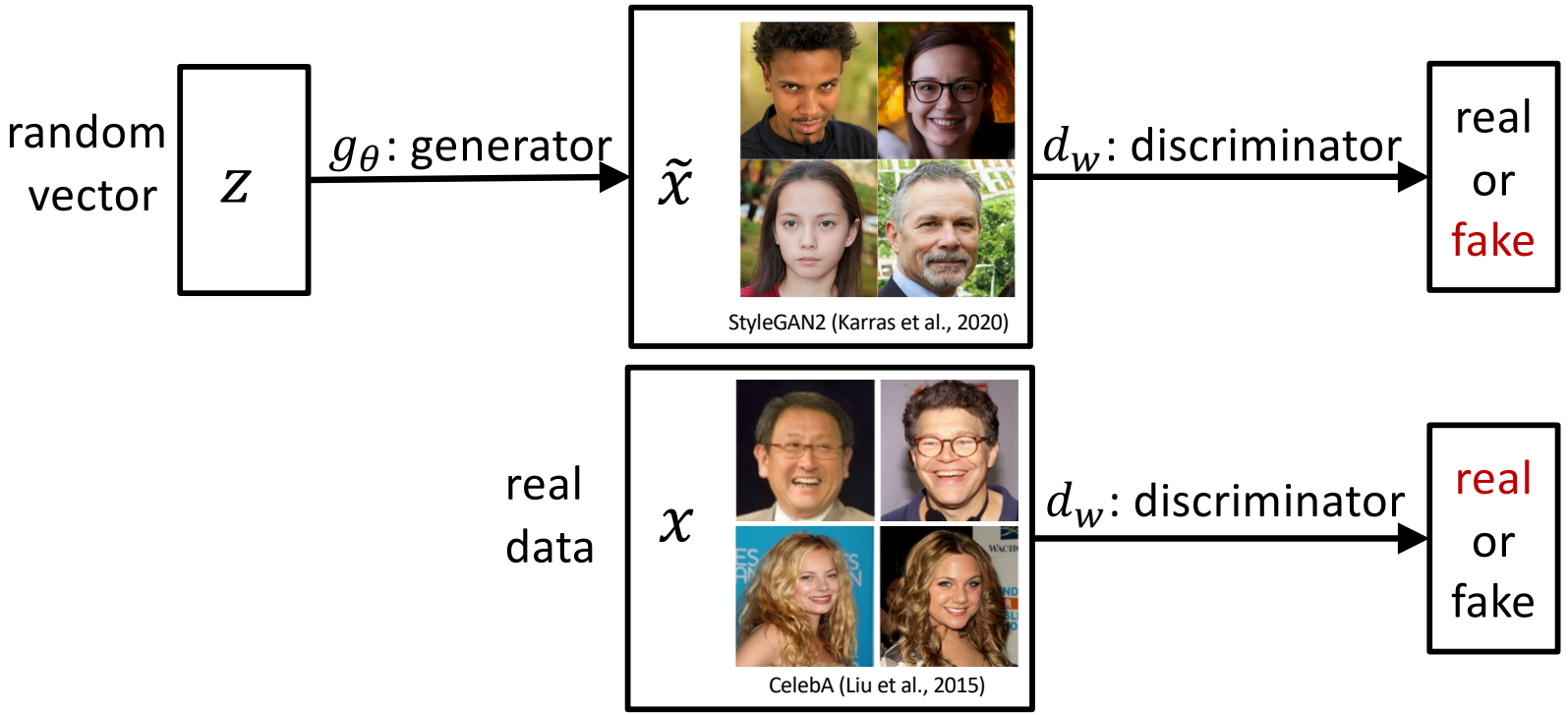
Sordoni et al., 2015

#	Context c	Message m	Generated Response r
1	noone can help this time its just i need a support system	well im here to support you. whenever you need me.	thank you, i appreciate it thank you
2	nothin much, and how's the book?!	its good but i'm only like halfway through cuz i don't feel like reading. i'm so bored ...	that's good! i have the book but i'm bored too.
3	i know! what have you been up to?	nothing really. just enjoying summer and working. what about you?	that's good! nothing much just listening to music and you?

Generative adversarial imitation learning (GAIL)

- Common approach: training generator to maximize likelihood of expert actions
- Alternative: train generator to fool a discriminator in believing that the generated actions are from expert
  - Leverage GANs (Generative adversarial networks)
  - Ho & Ermon, 2016

Generative adversarial networks (GANs)



$$\min_{\theta}\max_{w}\sum_{n}\log \Pr(x_n \; is \; real; w) + \log(\Pr(g_\theta(z_n) \; is \; fake; w)$$

$$= \min_{\theta}\max_{w}\sum_{n}\log d_w(x_n) + \log\left(1 - d_w\left(g_\theta(z_n)\right)\right)$$

GAIL Pseudocode

Input: expert trajectories $\tau_e \sim \pi_{expert}$ where $\tau_e = (s_1, a_1, s_2, a_2, \dots)$
Initialize params θ of policy π_θ and params w of discriminator d_w
Repeat until stopping criterion

Update discriminator parameters:

$$\delta_w = \sum_{(s,a) \in \tau_e} \nabla_w \log d_w(s,a) + \sum_{s,a \sim \pi_\theta(a|s)} \nabla_w \log(1 - d_w(s,a))$$

$$w \leftarrow w + \alpha_w \delta_w$$

Update policy parameters with TRPO:

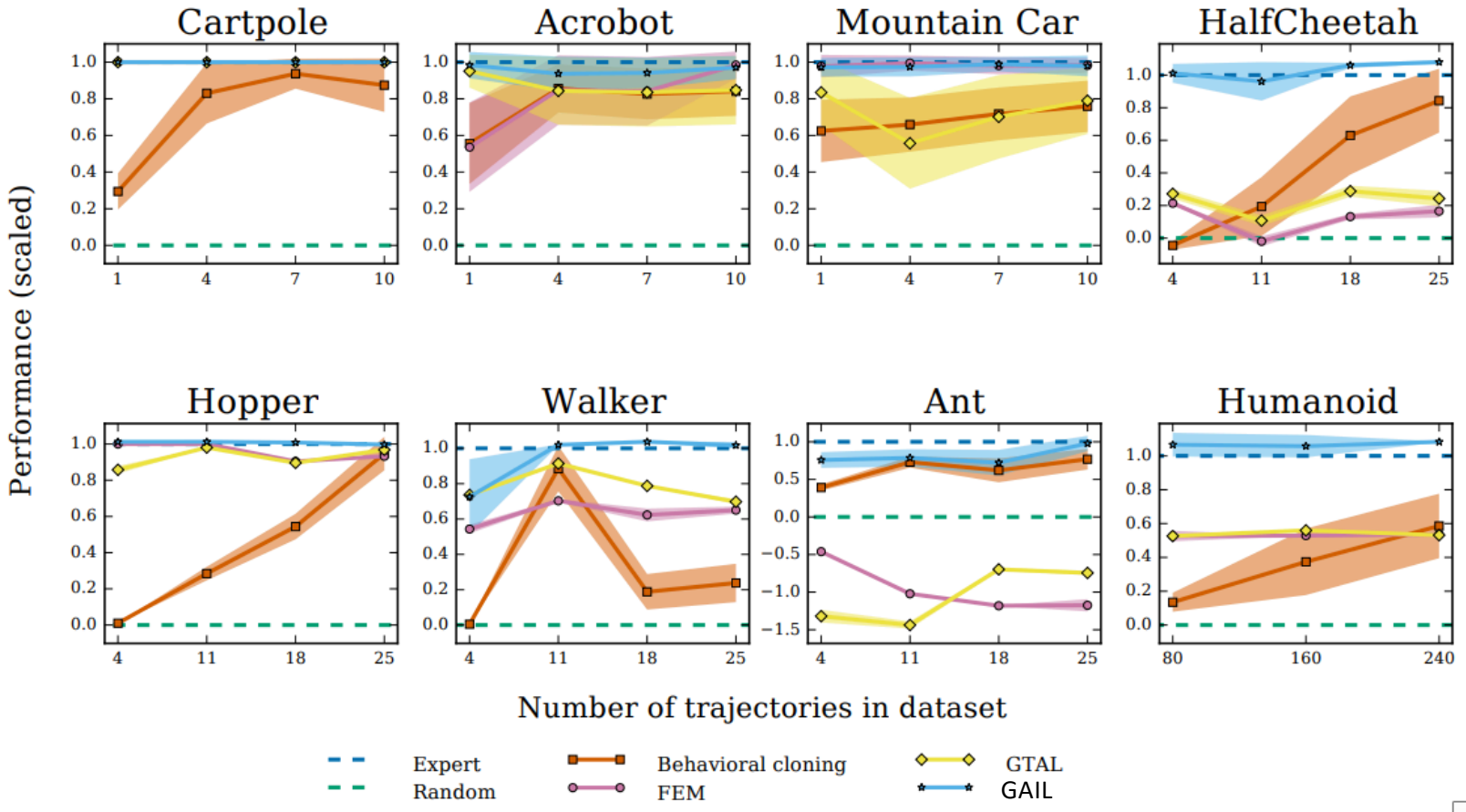
$$Cost(s_0, a_0) = \sum_{s,a|s_0,a_0,\pi_\theta} \log(1 - d_w(s,a))$$

$$\delta_\theta = \left[\sum_{s,a|\pi_\theta} \nabla_\theta \log \pi_\theta(a|s) Cost(s,a)\right] - \lambda \nabla_\theta H(\pi_\theta)$$

$$\theta \leftarrow \theta - \alpha_\theta \delta_\theta$$

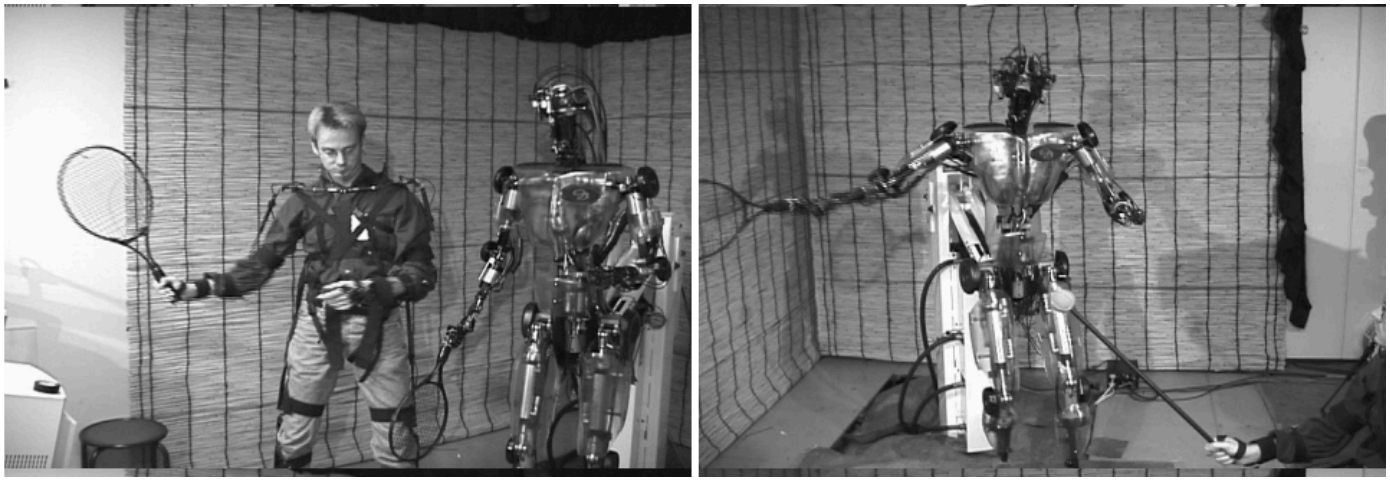
Robotics Experiments

- Robot imitating expert policy (Ho & Ermon, 2016)



Imitation Learning from Observations

- Consider imitation learning from a human expert:



Schaal et al., 2003

- Actions (e.g., forces) unobservable
- Only states/observations (e.g., joint positions) observable
- Problem: infer actions from state/observation sequences

Inverse Dynamics

Two steps:

1. Learn inverse dynamics
   - Learn $\Pr(a|s, s')$ by supervised learning
   - From (s, a, s') samples obtained by executing random actions

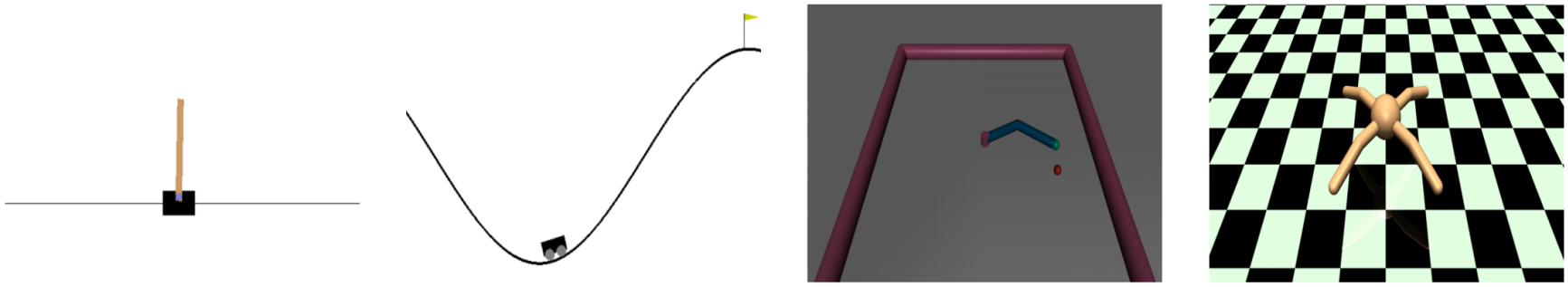
2. Behavioural cloning
   - Learn $\pi(\hat{a}|s)$ by supervised learning
   - From (s, s') samples from expert trajectories and from $\hat{a} \sim \Pr(a|s, s')$ sampled by inverse dynamics

Pseudocode:
Imitation Learning from Observations

Input: expert trajectories $\tau_e \sim \pi_{expert}$ where $\tau_e = (s_1, s_2, s_3, \dots)$
Initialize agent policy π_θ at random
Repeat
    Learn inverse dynamics model with parameters w:
        Sample $\left(s_t^{(\pi_\theta)}, a_t^{(\pi_\theta)}, s_{t+1}^{(\pi_\theta)}\right)$ by executing π_θ
        $w \leftarrow argmax_w \sum_t \log \Pr_w(a_t^{(\pi_\theta)} | s_t^{(\pi_\theta)}, s_{t+1}^{(\pi_\theta)})$
    Learn policy parameters θ:
        For each $\left(s_t^{(\tau_e)}, s_{t+1}^{(\tau_e)}\right)$ from expert trajectories τ_e do:
            $\hat{a}_t^{(\tau_e)} \sim \Pr(a_t^{(\tau_e)} | s_t^{(\tau_e)}, s_{t+1}^{(\tau_e)})$
        $\theta \leftarrow argmax_\theta \sum_t \log \pi_\theta(\hat{a}_t^{(\tau_e)} | s_t^{(\tau_e)})$

Robotics Experiments

Torabi et al., 2018



(a) CartPole (b) MountainCar (c) Reacher (d) Ant