

CS 277: Control and Reinforcement Learning

Winter 2026

Lecture 13: Inverse RL

Roy Fox

Department of Computer Science

School of Information and Computer Sciences

University of California, Irvine



Logistics

participation

- If you participated in class
 - I appreciate it!
 - Have a ducky
 - Please PM me on the forum so I have your name
- Still time to participate in class / forum / office hours
- Don't forget course evaluations



State of the Course

- Model-Free RL
 - Imitation Learning
 - Temporal Difference
 - Policy Gradient
 - Actor–Critic
- Model-Based RL (and friends)
 - Exploration
 - Partial Observability
 - Planning
 - World Models
- Other settings
 - Inverse RL
 - Bounded RL
 - Multi-task RL
 - Offline RL
 - Non-reward feedback
 - Multi-Agent RL



Today's lecture

Sparse rewards

Inverse RL

MaxEnt IRL

GAIL

Relation between RL and IL

- What makes RL harder than IL?
 - IL: expert policy $\pi^*(a | s)$ indicates a good action to take in s
 - RL: $r(s, a)$ does not indicate a globally good action; $Q^*(s, a)$ does, but it's nonlocal
- But didn't we see an equivalence between RL and IL?
 - NLL loss in BC: $-\nabla_{\theta} \mathbb{E}[\log \pi_{\theta}(a | s)]$
 - s and a sampled from teacher distribution
 - PG loss: $-\nabla_{\theta} \mathbb{E}[R \log \pi_{\theta}(a | s)]$
 - s and a sampled from learner distribution

Informational quantities: refresher

- Entropy: $\mathbb{H}[p(a)] = -\mathbb{E}_{a \sim p}[\log p(a)] = -\sum_a p(a) \log p(a)$
- Conditional entropy: $\mathbb{H}[\pi | s] = -\mathbb{E}_{a \sim \pi}[\log \pi(a | s)]$
- Expected conditional entropy: $\mathbb{H}[\pi] = \mathbb{E}_{s \sim p_\pi}[\mathbb{H}[\pi | s]] = -\mathbb{E}_{s, a \sim p_\pi}[\log \pi(a | s)]$
- Expected relative entropy: $\mathbb{D}[\pi || \pi'] = \mathbb{E}_{s, a \sim p_\pi} \left[\log \frac{\pi(a | s)}{\pi'(a | s)} \right]$
- Expected cross entropy (aka NLL): $-\mathbb{E}_{s, a \sim p_\pi}[\log \pi'(a | s)]$
 - $\mathbb{D}[\pi || \pi'] = \text{NLL} - \mathbb{H}[\pi]$

IL as sparse-reward RL

- **NLL BC**: maximize $\mathbb{E}_{s,a \sim p^*}[\log \pi_\theta(a | s)] = -\mathbb{D}[\pi^* || \pi_\theta] - \mathbb{H}[\pi^*]$
 - ▶ Experience from **teacher distribution** p^*
 - RL: experience from **learner distribution** p_θ
 - ▶ “Return” $R = 1_{\text{success}}$ for **successful trajectory**
 - RL: $r_t = r(s_t, a_t)$ in **every step**
- **Sparse reward** = most rewards are 0 \implies rare learning signal
 - ▶ $R = 1$ on success = very sparse; but doesn't IL provide **dense learning signal**?

constant in θ

IL as dense-reward RL

- What if instead we minimize the **other relative entropy**?

$$\mathbb{D}[\pi_\theta \parallel \pi^*] = - \mathbb{E}_{s, a \sim p_\theta} [\log \pi^*(a | s)] - \mathbb{H}[\pi_\theta]$$

teacher labeling of learner states/actions
as in DAgger

- ▶ This is exactly an **RL objective** with $\tilde{r}(s, a) = \log \pi^*(a | s)$ and entropy **regularizer**
 - ▶ Now $\tilde{r}(s, a)$ does give **global** information on optimal action
 - ▶ In fact, with **deterministic** teacher, $\tilde{r}(s, a) = -\infty$ for any suboptimal action
- The same return can be viewed as **dense** reward or **sum of sparse** rewards
 - ▶ Can we do the same in proper RL?

Reward shaping

- "Ideal" reward: $r(s, a) = -\infty$ for suboptimal action \implies as hard to provide as π^*
 - We need supervision signal that's sufficiently easy to program \implies generate more data
 - We also need it to be easy to learn from, not as brittle as "optimal or die"
- Sparse often easier to provide, harder to use: goal states, safety violations, etc.
- Reward shaping: art of adjusting the reward function for easier RL; some tips:
 - Reward "bottleneck states": subgoals that are likely to lead to bigger goals
 - Break down long sequences of coordinated actions \implies better exploration
 - E.g. reward beacons on long narrow paths, for exploration to stumble upon

Today's lecture

Sparse rewards

Inverse RL

MaxEnt IRL

GAIL

Learning rewards from demonstrations

- RL: rewards \rightarrow policy; IL: demonstrations \rightarrow policy
- Inverse Reinforcement Learning (IRL): demonstrations \rightarrow reward function
 - Better understand agents (humans, animals, users, markets)
 - Preference elicitation, teleology (the “what for” of actions), theory of mind, language
 - First step toward Apprenticeship Learning: demos \rightarrow rewards \rightarrow policy
 - Infer the teacher's goals and learn to achieve them; overcome suboptimal demos
 - Partly model-based (learn r but not p); may be easier to learn, generalize, transfer
 - Teacher and learner may have different observation + action spaces (human \rightarrow robot)

Inverse Reinforcement Learning (IRL)

- Given a dataset of **demonstration trajectories** $\mathcal{D} = \{\xi_i\}$
- Find teacher's **reward function** $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$
 - ▶ **Principle**: demonstrated actions should achieve high expected return
- IRL is **ill-defined**
 - ▶ How low is the reward for states and actions **not in** \mathcal{D} ?
 - ▶ How is the reward **distributed** along the trajectory?
 - Sparse rewards = identify “**subgoal**” states; dense = score **each step**, as hard as IL
 - ▶ Demonstrator can be **fallible** = take suboptimal actions; how much?

Feature matching

- Assume linear reward $r_\theta(s) = \theta^\top f_s$ in given state features $f_s \in \mathbb{R}^d$
 - ▶ Value = $J_\theta^\pi = \sum_t \gamma^t \mathbb{E}_{s_t \sim p_\pi} [\theta^\top f_{s_t}] = \mathbb{E}_{s \sim p_\pi} [\theta^\top f_s]$, with $p_\pi(s) = (1 - \gamma) \sum_t \gamma^t p_\pi(s_t)$ 
- Teacher optimality: expert value $J_\theta^{\pi^*}$ higher than any other policy's value J_θ^π
 - ▶ Find θ that maximizes the gap $J_\theta^{\pi^*} - J_\theta^\pi$; but for which π ?
 - ▶ Apprenticeship Learning: find π that maximizes J_θ^π ; but for which θ ?
- Solve: $\max_\theta \min_\pi \{J_\theta^{\pi^*} - J_\theta^\pi\} = \max_\theta \min_\pi \{ \mathbb{E}_{s \sim p^*} [\theta^\top f_s] - \mathbb{E}_{s \sim p_\pi} [\theta^\top f_s] \}$
 - ▶ Approximate $s \sim p^*$ with $s \sim \mathcal{D}$

Feature matching

- Solving $\max_{\theta} \min_{\pi} \{ \mathbb{E}_{s \sim p^*} [\theta^\top f_s] - \mathbb{E}_{s \sim p_\pi} [\theta^\top f_s] \}$

Algorithm Feature Matching

Initialize policy set $\Pi = \{\pi_0\}$

repeat

Solve Quadratic Program: $\max_{\eta, \|\theta\|_2 \leq 1} \eta$ ← θ must be bounded, or solution is ∞

s.t. $\mathbb{E}_{s \sim \mathcal{D}} [\theta^\top f_s] \geq \mathbb{E}_{s \sim p_\pi} [\theta^\top f_s] + \eta \quad \forall \pi \in \Pi$

$\pi \leftarrow$ optimal policy for $r_\theta(s) = \theta^\top f_s$

Add π to Π

- **On convergence:** π optimal for θ (no gap), while no θ increases the gap feature matching

▶ $\Rightarrow \mathbb{E}_{s \sim \mathcal{D}} [\theta^\top f_s] \approx \mathbb{E}_{s \sim p_\pi} [\theta^\top f_s]$ for all $\theta \Rightarrow \mathbb{E}_{s \sim \mathcal{D}} [f_s] \approx \mathbb{E}_{s \sim p_\pi} [f_s]$ ← feature matching

Today's lecture

Sparse rewards

Inverse RL

MaxEnt IRL

GAIL

Modeling bounded teachers

- An **expert** teacher maximizes the value $J_{\theta}^{\pi_{\beta}} = \sum_t \gamma^t \mathbb{E}_{s_t \sim p_{\beta}} [\theta^{\top} f_{s_t}] = \mathbb{E}_{\xi \sim p_{\beta}} [\theta^{\top} f_{\xi}]$
 - ▶ With trajectory-summed features $f_{\xi} = \sum_t \gamma^t f_{s_t}$
- Assume teacher has unintentional / uninformed **prior policy** π_0
 - ▶ **Bounded rationality**: cost to intentionally diverge $\mathbb{D}[\pi_{\beta} \parallel \pi_0]$ (with π_0 uniform: $\mathbb{H}[\pi_{\beta}]$)
 - ▶ Total cost: $\sum_t \mathbb{E}_{(s_t, a_t) \sim p_{\beta}} \left[\log \frac{\pi_{\beta}(a_t | s_t)}{\pi_0(a_t | s_t)} \right] = \mathbb{E}_{\xi \sim p_{\beta}} \left[\log \frac{p_{\beta}(\xi)}{p_0(\xi)} \right] = \mathbb{D}[p_{\beta}(\xi) \parallel p_0(\xi)]$

for simplicity, for now assume $\tau = 1$
- **Bounded optimality**: $\max_{\pi_{\beta}} \mathbb{E}_{\xi \sim p_{\beta}} [\theta^{\top} f_{\xi}] - \tau \mathbb{D}[p_{\beta} \parallel p_0]$

Bounded optimality: naïve solution

- Bounded optimality: $\max_{\mathcal{P}_\beta} \mathbb{E}_{\xi \sim p_\beta} [\theta^\top f_\xi] - \mathbb{D}[p_\beta \| p_0]$
 $\mathbb{E}_{\xi \sim p_\beta} [\log p_\beta(\xi) - \log p_0(\xi)]$
- ▶ Naïve solution: allow **any** distribution p_β over trajectories
- ▶ No need to be consistent with **dynamics** $p(s' | s, a) \Rightarrow p_\beta$ may be **unachievable**

- Add the **constraint** $\sum_{\xi} p_\beta(\xi) = 1$ with Lagrange multiplier λ

- **Differentiate** by $p_\beta(\xi)$ and $= 0$ to optimize

$$\theta^\top f_\xi - \log p_\beta(\xi) + \log p_0(\xi) - 1 + \lambda = 0 \implies p_\beta(\xi) = \frac{p_0(\xi) \exp(\theta^\top f_\xi)}{\sum_{\bar{\xi}} p_0(\bar{\xi}) \exp(\theta^\top f_{\bar{\xi}})}$$

IRL with bounded teacher

- Assume that **demonstrations** are distributed $p_{\theta}(\xi) = \frac{1}{Z_{\theta}} p_0(\xi) \exp(\theta^{\top} f_{\xi})$
 - ▶ With **partition function** $Z_{\theta} = \mathbb{E}_{\bar{\xi} \sim p_0} [\exp(\theta^{\top} f_{\bar{\xi}})]$
- Find θ that **minimizes NLL** of demonstrations

$$\begin{aligned} \nabla_{\theta} \log p_{\theta}(\xi) &= \nabla_{\theta} (\theta^{\top} f_{\xi} - \log Z_{\theta}) = f_{\xi} - \frac{1}{Z_{\theta}} \nabla_{\theta} Z_{\theta} \\ &= f_{\xi} - \frac{1}{Z_{\theta}} \mathbb{E}_{\bar{\xi} \sim p_0} [\exp(\theta^{\top} f_{\bar{\xi}}) f_{\bar{\xi}}] = f_{\xi} - \mathbb{E}_{\bar{\xi} \sim p_{\theta}} [f_{\bar{\xi}}] \end{aligned}$$

- ▶ To compute gradient, we need p_{θ} , but how to **compute** Z_{θ} ?

Computing Z_θ : backward recursion

- Partition function: $Z_\theta = \mathbb{E}_{\xi \sim p_0}[\exp(\theta^\top f_\xi)]$
← product of step-wise terms $\exp(\theta^\top f_{s_t})$
- Compute Z_θ recursively **backward**: like a value function, but + becomes \cdot .

$$Z_\theta(s_t, a_t) = \mathbb{E}_{p_0}[\exp(\theta^\top f_{\xi_{\geq t}}) \mid s_t, a_t] = \exp(\theta^\top f_{s_t}) \mathbb{E}_{(s_{t+1} \mid s_t, a_t) \sim p}[Z_\theta(s_{t+1})]$$

$$Z_\theta(s_t) = \mathbb{E}_{p_0}[\exp(\theta^\top f_{\xi_{\geq t}}) \mid s_t] = \mathbb{E}_{(a_t \mid s_t) \sim \pi_0}[Z_\theta(s_t, a_t)]$$

part of the normalizer involving trajectories following (s_t, a_t)

- How to get a policy from Z_θ ?

← everything up to s_t cancels out

$$\text{Marginalize: } \pi_\theta(a_t \mid s_t) = \frac{p_\theta(\xi \mid s_t, a_t)}{p_\theta(\xi \mid s_t)} = \frac{p_0(\xi_{\geq t} \mid s_t, a_t) \exp(\theta^\top f_{\xi_{\geq t}}) \cdot Z_\theta(s_t)}{Z_\theta(s_t, a_t) \cdot p_0(\xi_{\geq t} \mid s_t) \exp(\theta^\top f_{\xi_{\geq t}})} = \pi_0(a_t \mid s_t) \frac{Z_\theta(s_t, a_t)}{Z_\theta(s_t)}$$

consistent π may not even exist

- ▶ This π_θ is not globally **consistent** $p_\theta(\xi) \neq p_{\pi_\theta}(\xi)$, as $p_\theta(\xi)$ ignores the **dynamics**

MaxEnt IRL

- For each sample $\xi \sim \mathcal{D}$:

Limitations:

- ▶ Compute $Z_\theta = \mathbb{E}_{\xi \sim p_0} [\exp(\theta^\top f_\xi)]$ recursively **backward**
 - ▶ Compute $\mathbb{E}_{\bar{\xi} \sim p_{\pi_\theta}} [f_{\bar{\xi}}]$ recursively **forward**
 - ▶ Take a gradient step to **improve** θ : $\nabla_\theta \log p_\theta(\xi) \approx f_\xi - \mathbb{E}_{\bar{\xi} \sim p_{\pi_\theta}} [f_{\bar{\xi}}]$
- Requires dynamics p
 - Assumes $p_\theta = p_{\pi_\theta}$
 - Assumes $\mathcal{D} = p^*$

- At the optimum: **feature matching** $\mathbb{E}_{\xi \sim \mathcal{D}} [f_\xi] = \mathbb{E}_{\bar{\xi} \sim p_{\pi_\theta}} [f_{\bar{\xi}}]$ **equivalent to feature matching with entropy-regularized RL**

- ▶ **MaxEnt IRL** approximates $\max_{\theta} \mathbb{H}[\pi_\theta] \quad \text{s.t.} \quad \mathbb{E}_{\xi \sim \mathcal{D}} [f_\xi] = \mathbb{E}_{\bar{\xi} \sim p_{\pi_\theta}} [f_{\bar{\xi}}]$

Today's lecture

Sparse rewards

Inverse RL

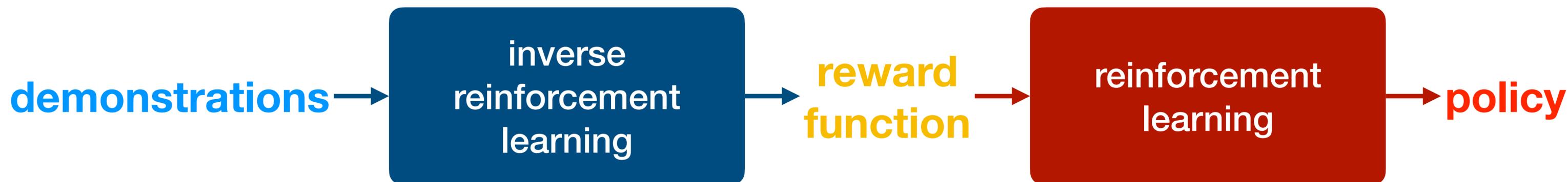
MaxEnt IRL

GAIL

IRL: downstream tasks

- One IRL **motivation**: learn reward function for downstream tasks

...such as **RL**



- $IL = RL \circ IRL$ (**composition** of RL on IRL)
- Our algorithms already **learn** π as part of learning θ for $r : s \mapsto \theta^\top f_s$
 - Let's **directly optimize** IRL for the overall IL task = learn good π

IL as RL \circ IRL

- Entropy-regularized RL: $\max_{\pi \in \Pi} \left\{ \mathbb{E}_{s \sim p_\pi} [r(s)] + \mathbb{H}[\pi] \right\}$
- MaxEnt IRL: $\max_{r \in \mathbb{R}^{\mathcal{S}}} \left\{ \mathbb{E}_{s \sim p_\beta} [r(s)] - \max_{\pi \in \Pi} \left\{ \mathbb{E}_{s \sim p_\pi} [r(s)] + \mathbb{H}[\pi] \right\} \right\} - \psi(r)$

regularization over
reward-function space



- For any π , our objective with respect to r is:

$$\max_{r \in \mathbb{R}^{\mathcal{S}}} \left\{ \overbrace{(p_\beta - p_\pi)}^{\in \mathbb{R}^{\mathcal{S}}} \cdot r - \psi(r) \right\} = \psi^*(p_\beta - p_\pi)$$

* denotes Fenchel conjugate
(not “optimality”)

- ▶ This form of function $\psi^* : \mathbb{R}^{\mathcal{S}} \rightarrow \mathbb{R}$ is called the convex (Fenchel) conjugate of ψ

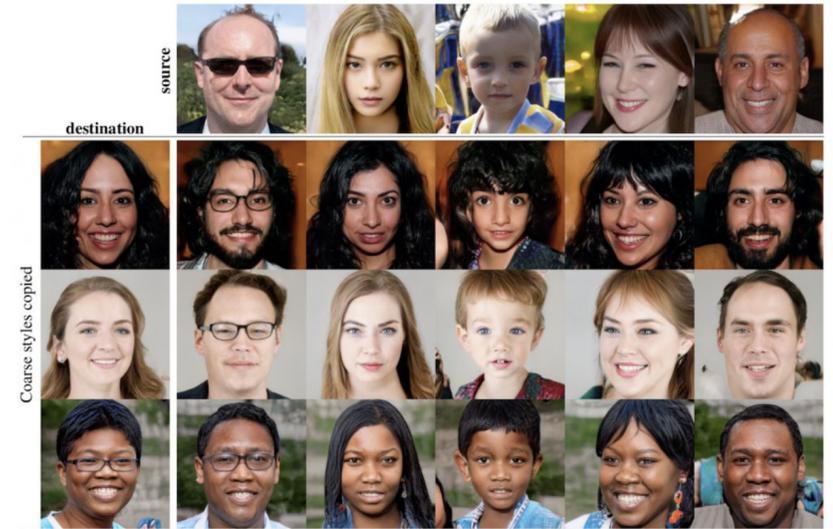
Reward-function regularizers

$$\psi^*(p_\beta - p_\pi) = \max_{r \in \mathbb{R}^{\mathcal{S}}} \{ (p_\beta - p_\pi) \cdot r - \psi(r) \}$$

- **Without regularizer:** $\psi = 0 \implies$ solution only exists when $p_\beta = p_\pi$
 - \implies learner achieves teacher's **state distribution**: perfect solution, but hard to find
- **Hard linearity constraint:** $\psi(r) = \begin{cases} 0 & \text{if } r(s) = \theta^\top f_s \\ \infty & \text{otherwise} \end{cases}$
 - \implies max-entropy feature matching (**MaxEnt IRL**)
 - Great when the reward function really is **linear in f_s** , otherwise no guarantees

Generative Adversarial Networks (GANs)

- Train **generative model** $p_{\theta}(s)$ to generate states / observations
 - Can we focus the training on **failure modes**?
- Also train **discriminator** $D_{\phi}(s) \in [0,1]$ to score instances
 - Kind of like a critic: are generated instances good?

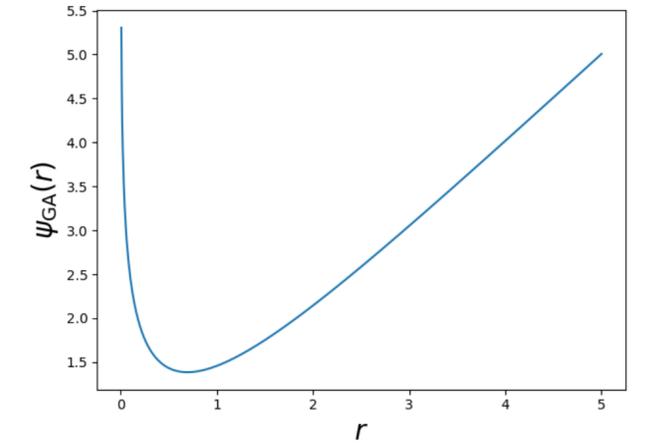


- $D_{\phi}(s)$ **predicts** the probability $p(s \text{ generated by learner} | s) = \frac{p_{\theta}(s)}{p_{\theta}(s) + p_{\beta}(s)}$
 - Trained with cross-entropy loss: $\max_{\phi} \left\{ \mathbb{E}_{s \sim p_{\theta}} [\log D_{\phi}(s)] + \mathbb{E}_{s \sim p_{\beta}} [\log(1 - D_{\phi}(s))] \right\}$
- The generator tries to **fool** the discriminator: $\min_{\theta} \mathbb{E}_{s \sim p_{\theta}} [\log D_{\phi}(s)]$
 \swarrow
 $-\tilde{r}(s)$

Teacher-based reward-function regularizer

- Consider the **regularizer**

$$\psi_{\text{GA}}(\tilde{r}) = \mathbb{E}_{s \sim p_{\beta}} \left[\underbrace{\tilde{r}(s)}_{-\log D(s)} - \underbrace{\log(1 - \exp(-\tilde{r}(s)))}_{D(s)} \right]$$



- It's **convex conjugate** is:

$$\begin{aligned} \psi_{\text{GA}}^*(p_{\beta} - p_{\pi}) &= \max_{\tilde{r} \in \mathbb{R}^{\mathcal{S}}} \left\{ (p_{\beta} - p_{\pi}) \cdot \tilde{r} - \psi_{\text{GA}}(\tilde{r}) \right\} \\ &= \max_{\tilde{r} \in \mathbb{R}^{\mathcal{S}}} [\tilde{r}(s) - \tilde{r}(s) + \log(1 - \exp(-\tilde{r}(s)))] - \mathbb{E}_{s \sim p_{\pi}}[\tilde{r}(s)] \\ &= \mathbb{E}_{s \sim p_{\pi}}[\log D(s)] + \mathbb{E}_{s \sim p_{\beta}}[\log(1 - D(s))] \end{aligned}$$

- \implies GAN: **generator** p_{π} imitating **teacher** p_{β} ; **discriminator** $D(s) = \exp(-\tilde{r}(s))$

Generative Adversarial Imitation Learning (GAIL)

Input: demonstration dataset $\mathcal{D}_T \sim p_T$

repeat

$\mathcal{D}_L \leftarrow$ roll out π_θ

take discriminator gradient ascent step

$$\mathbb{E}_{s \sim \mathcal{D}_L} [\nabla_\phi \log D_\phi(s)] + \mathbb{E}_{s \sim \mathcal{D}_T} [\nabla_\phi \log(1 - D_\phi(s))]$$

take entropy-regularized policy gradient step with reward $r(s) = -\log D_\phi(s)$

- We've already seen one entropy-regularized PG algorithm: **TRPO**
 - More next time

Recap

- To understand behavior: **infer the intentions** of observed agents
- If teacher is **optimal** for a reward function
 - The reward function should make an optimizer **imitate** the teacher
 - State (or state–action) **distribution** of learner should **match** the teacher
- In this view, **Inverse Reinforcement Learning (IRL)** is a game:
 - Reward is optimized to show how much the **teacher is better** than the learner
 - **Learner optimizes** for the reward
 - Reward is like a **discriminator** (high = probably teacher); learner like a **generator**