# Generative Adversarial Imitation Learning

Alex Zuzow

10/28/2021

# Motivation

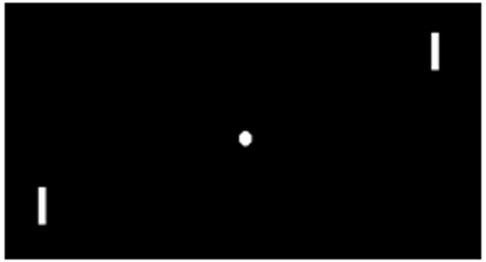
**Mimic human behavior in a given task.**

- Sometimes it is easier for an expert to demonstrate a task instead of specifying a reward function.
- Facilitates teaching complex tasks without the need for explicitly designing a reward function
- Applicable to large, high dimensional environments

# Imitation

**Input: expert behavior generated by $\pi_E$**

$$\{(s_0^i, a_0^i, s_1^i, a_1^i, \ldots)\}_{i=1}^n \sim \pi_E$$



**Goal: learn *cost function (reward) or policy***

(Ng and Russell, 2000), (Abbeel and Ng, 2004; Syed and Schapire, 2007), (Ratliff et al., 2006), (Ziebart et al., 2008), (Kolter et al., 2008), (Finn et al., 2016), etc.

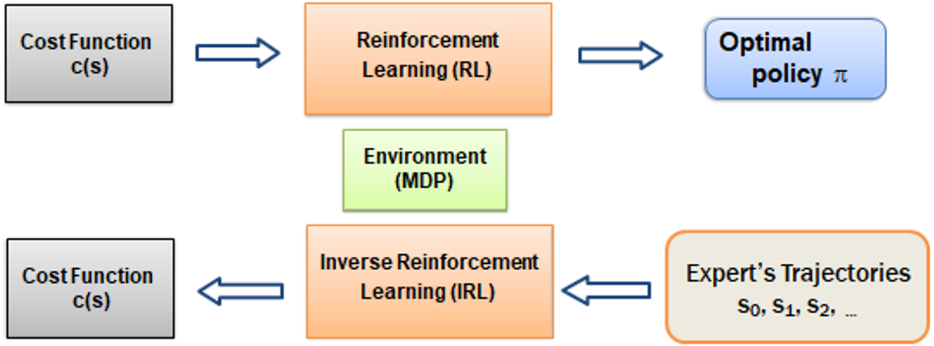
# Inverse RL

- An approach to imitation
- Learns a cost c such that

$$\pi_E = \arg\max_{\pi} \mathbb{E}_{\pi}[c(s, a)]$$

# Problem setup

$$\mathrm{RL}(c) = \arg\min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_\pi[c(s,a)]$$



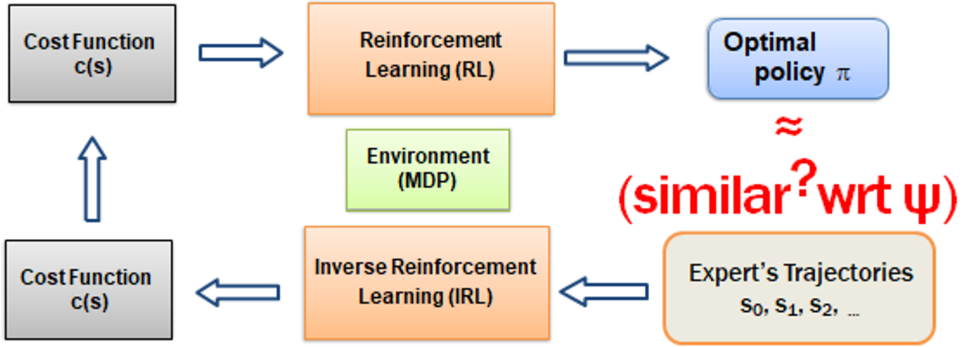

$$\underset{c \in \mathcal{C}}{\text{maximize}} \left( \min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_\pi[c(s,a)] \right) - \mathbb{E}_{\pi_E}[c(s,a)]$$

(Ziebart et al., 2010; Rust 1987)

Everything else has high cost

Expert has small cost

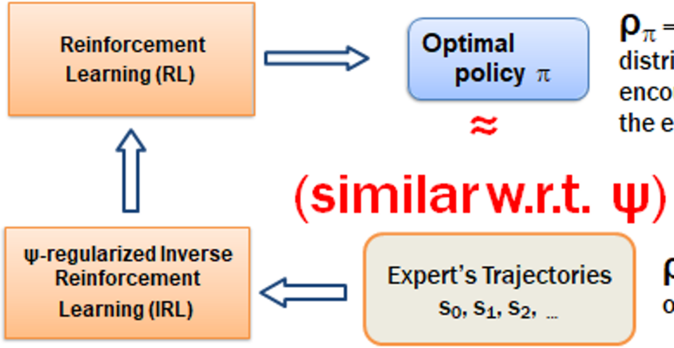
# Problem setup




$$\mathrm{IRL}_\psi(\pi_E) = \underset{c \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}}{\arg\max} \boxed{-\psi(c)} + \left(\min_{\pi\in\Pi} -H(\pi) + \mathbb{E}_\pi[c(s,a)]\right) - \mathbb{E}_{\pi_E}[c(s,a)]$$

Convex cost regularizer

# Combining RL∘IRL




$\rho_\pi$ = occupancy measure = distribution of state-action pairs encountered when navigating the environment with the policy

$\rho_{\pi E}$ = Expert's occupancy measure

**Theorem: ψ-regularized inverse reinforcement learning, implicitly, seeks a policy whose occupancy measure is close to the expert's, as measured by ψ* (convex conjugate of ψ)**

$$\mathrm{RL} \circ \mathrm{IRL}_\psi(\pi_E) = \arg\min_{\pi \in \Pi} -H(\pi) + \psi^*(\rho_\pi - \rho_{\pi_E})$$

# Takeaway

Theorem: ψ-regularized inverse reinforcement learning, implicitly, seeks a policy whose occupancy measure is close to the expert's, as measured by $\psi^*$

- Typical IRL definition: finding a cost function c such that the expert policy is uniquely optimal w.r.t. c

- Alternative view: IRL as a procedure that tries to induce a policy that matches the expert's occupancy measure (generative model)

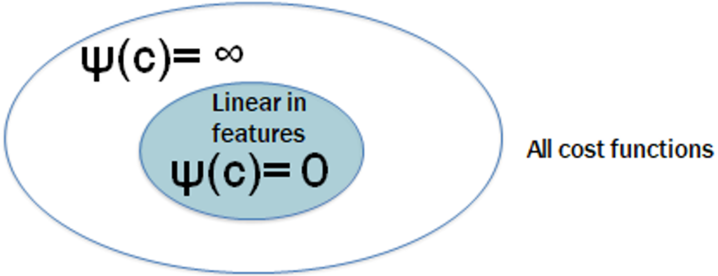
# Related Work

## Apprenticeship learning

- Solution: use **features** $f_{s,a}$
- Cost $c(s,a) = \theta \cdot f_{s,a}$

$$\mathrm{IRL}_\psi(\pi_E) = \underset{c \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}}{\arg\max}\; -\psi(c) + \left( \min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_\pi[c(s,a)] \right) - \mathbb{E}_{\pi_E}[c(s,a)]$$

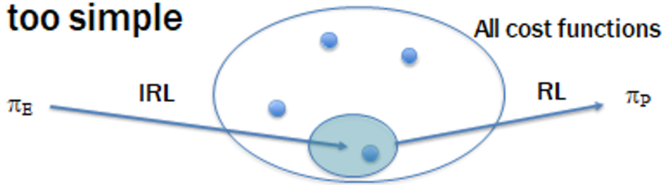
Only these "simple" cost functions are allowed



# Related Work

## Issues with Apprenticeship learning

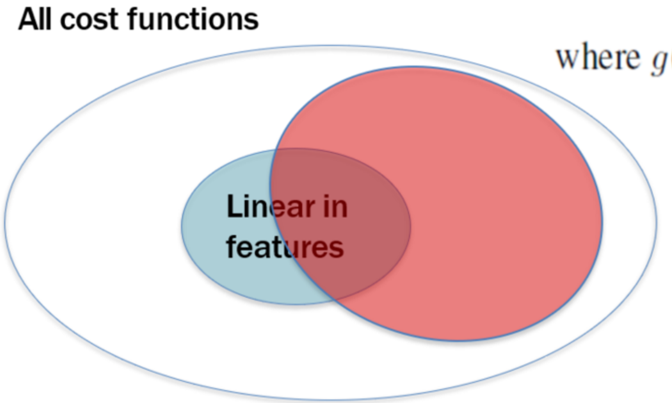
- Need to craft features very carefully
  - unless the true expert cost function (assuming it exists) lies in C, there is no guarantee that AL will recover the expert policy
- $RL \circ IRL_{\psi}(\pi_E)$ is "encoding" the expert behavior as a cost function in C.
  - it might not be possible to decode it back if C is too simple



# Generative Adversarial Imitation Learning

$$\psi_{\mathrm{GA}}(c) \triangleq \begin{cases} \mathbb{E}_{\pi_E}[g(c(s,a))] & \text{if } c < 0 \\ +\infty & \text{otherwise} \end{cases}$$

$$\text{where } g(x) = \begin{cases} -x - \log(1 - e^x) & \text{if } x < 0 \\ +\infty & \text{otherwise} \end{cases}$$



# Related Work

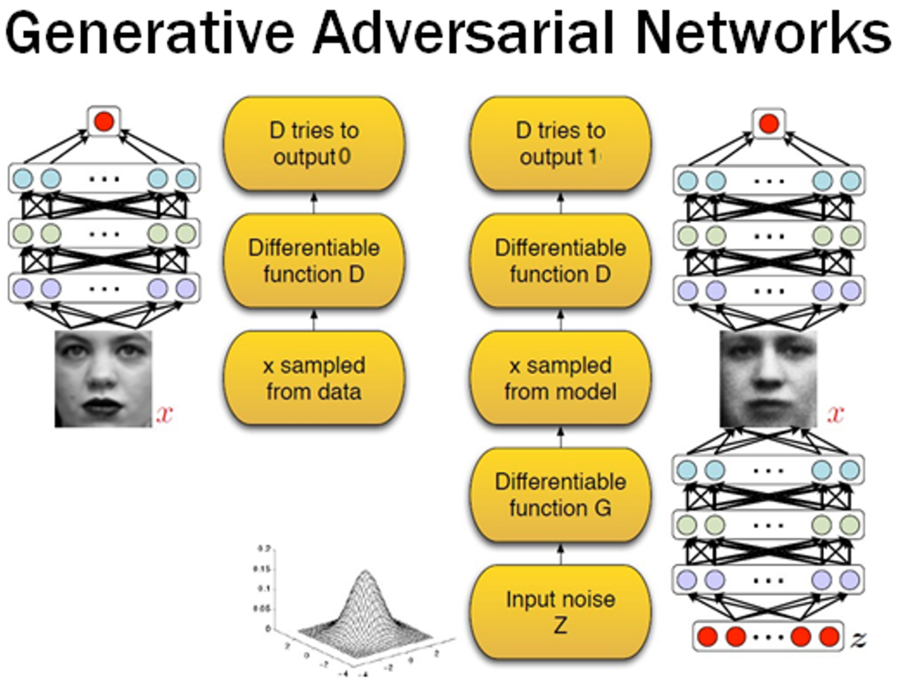
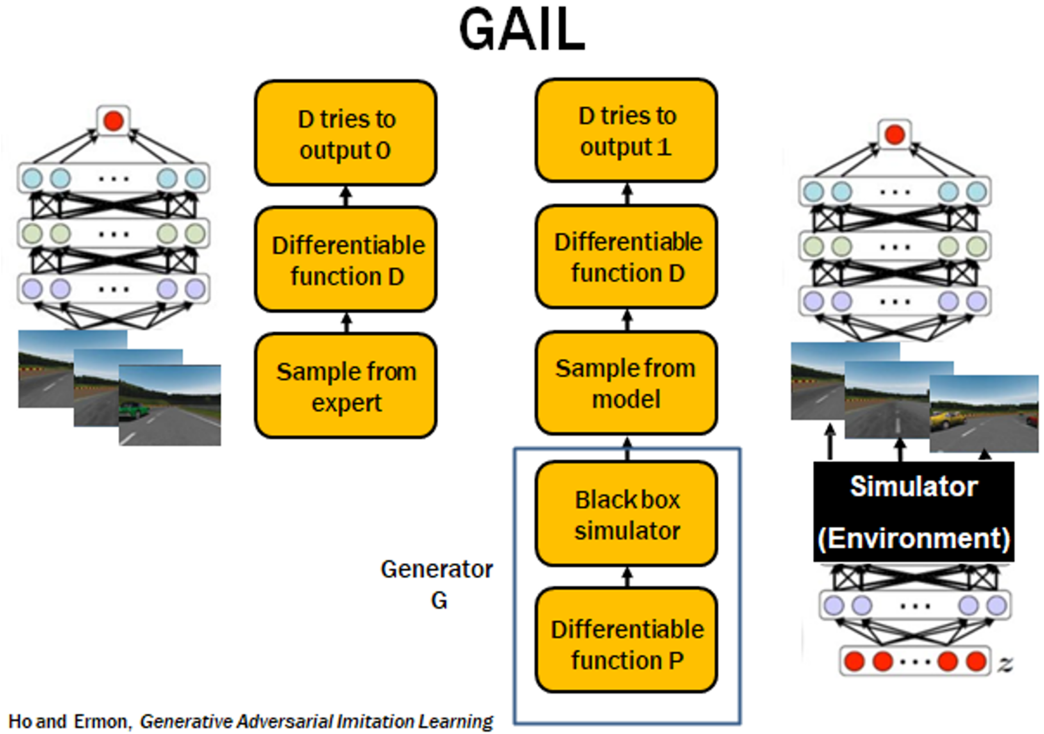



Figure from Goodfellow et al, 2014

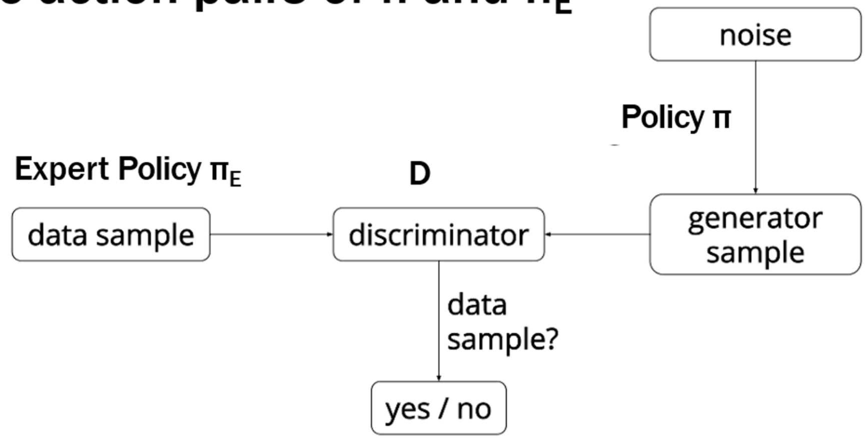
# Method




Ho and Ermon, *Generative Adversarial Imitation Learning*

# Generative Adversarial Imitation Learning

- $\psi^*$ = optimal negative log-loss of the binary classification problem of distinguishing between state-action pairs of $\pi$ and $\pi_E$




$$\psi^*_{\mathrm{GA}}(\rho_\pi - \rho_{\pi_E}) = \sup_{D \in (0,1)^{\mathcal{S}\times\mathcal{A}}} \mathbb{E}_\pi[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))]$$

# Experimental Setup

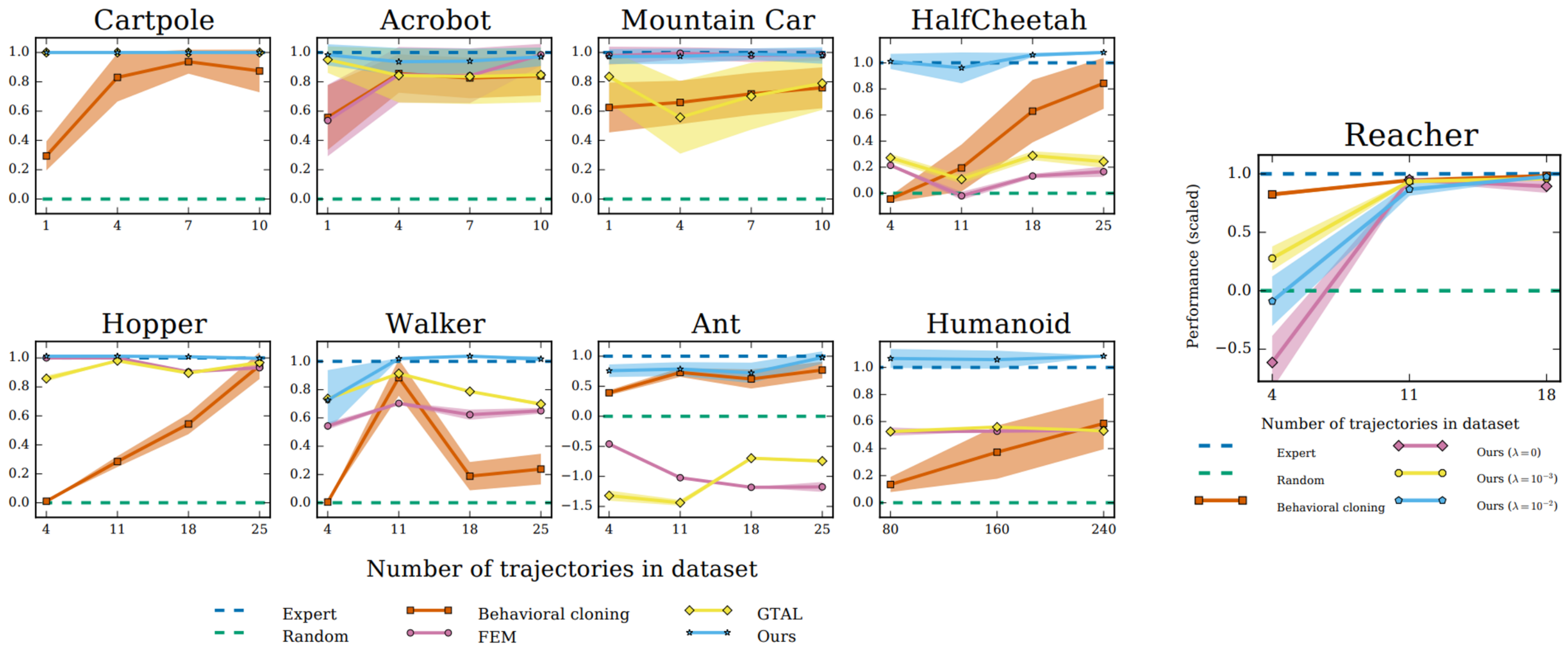
- Tested on 9 physics-based control tasks such as a 3D humanoid locomotion.
- Each task comes with a true cost function, defined in the OpenAI Gym.
- Expert trajectories generated for these tasks by running Trust Region Policy Optimization (TRPO).

# Experimental Setup

**Baselines**

- Behavioral cloning.
- Feature expectation matching (FEM).
- Game-theoretic apprenticeship learning (GTAL).

# Experimental Results



# Discussion of Results

- GAIL is generally quite sample efficient in terms of expert data.
- Inefficient in terms of environment interaction during training.
- GAIL always produced policies performing better than behavioral cloning, FEM, and GTAL.

# Critique / Limitations

- Requires a large number of environment interactions during training.
- Infeasible to train using real robots

# Future Work

- Precompute policy weights by applying behavioral cloning
- combine with a method like DAgger to allow GAIL to query expert when uncertain

# Extended Readings

- InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations
  https://arxiv.org/pdf/1703.08840.pdf
- Agail: Learning Robust Rewards with Adversarial Inverse Reinforcement.
  https://arxiv.org/pdf/1710.11248.pdf
- TextGAIL: Generative Adversarial Imitation Learning for Text Generation
  https://arxiv.org/pdf/2004.13796.pdf
- Triple-GAIL: A Multi-Modal Imitation Learning Framework with Generative Adversarial Nets
  https://arxiv.org/pdf/2005.10622.pdf
- MAGAIL: Multi-Agent Generative Adversarial Imitation Learning
  https://arxiv.org/pdf/1807.09936.pdf

# Summary

- **Problem**: learning complex behaviors in high dimensional environments.
  - Easier for an expert to demonstrate behaviors than specify a reward function.
- **Previous limitations**: imitation learning methods.
  - Inverse Reinforcement learning: required lots of expert demonstrations and computationally expensive.
  - Behavioral Cloning: small errors compound over time, poor generalization.
- **key insights:** directly learns a policy without an explicit reward function.
  - Similar to Inverse Reinforcement Learning although discriminator acts as reward function.