Generative Adversarial Imitation Learning

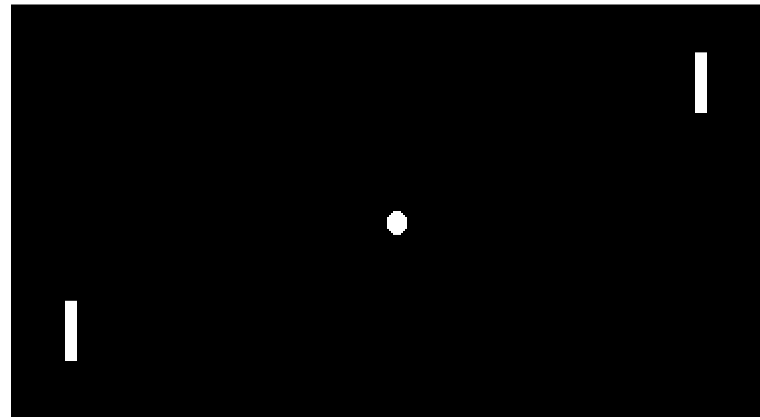
Stefano Ermon

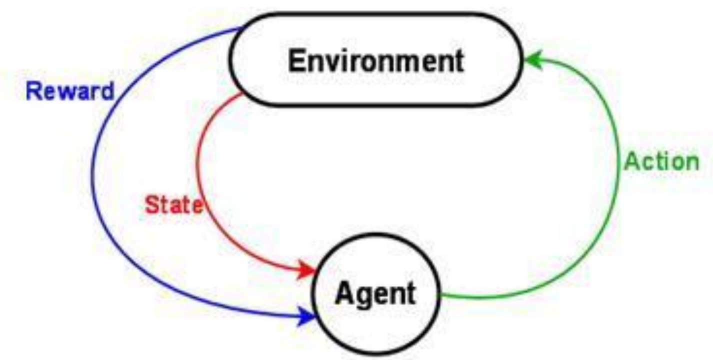
Joint work with Jayesh Gupta, Jonathan Ho, Yunzhu Li, and Jiaming Song

Stanford University

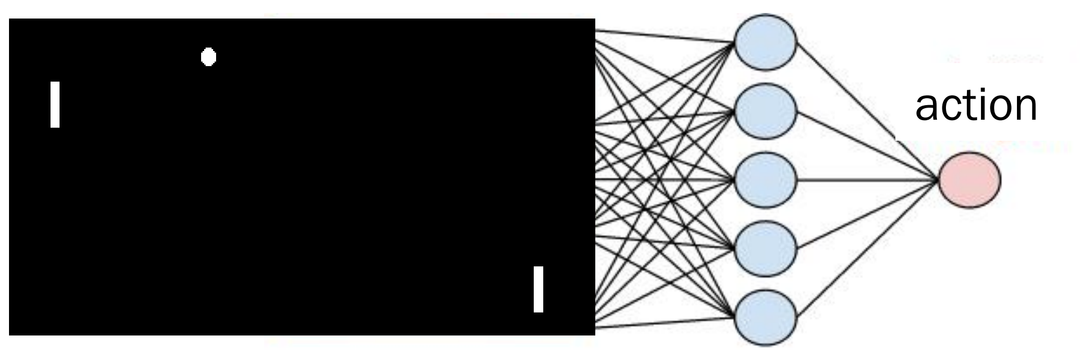
Reinforcement Learning

- Goal: Learn policies
- High-dimensional, raw observations





RL needs cost signal



Reinforcement Learning

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

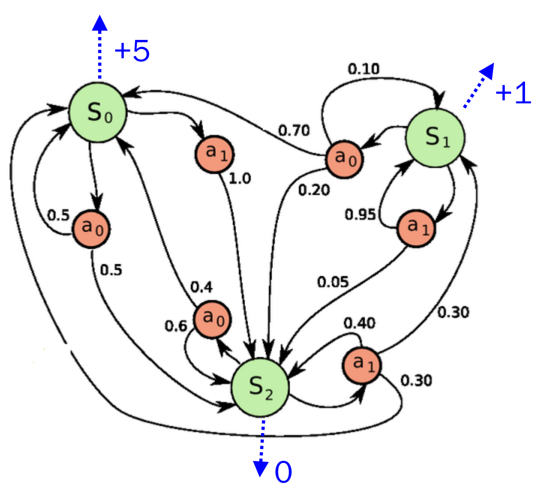
Cost Function c(s) ⇒ **Reinforcement Learning (RL)** ⇒ **Optimal policy π**

Environment (MDP)

$c : S \rightarrow \mathbf{R}$

Cost

Policy: mapping from states to actions
E.g., (S_0->a_1, S_1->a_0, S_2->a_0)

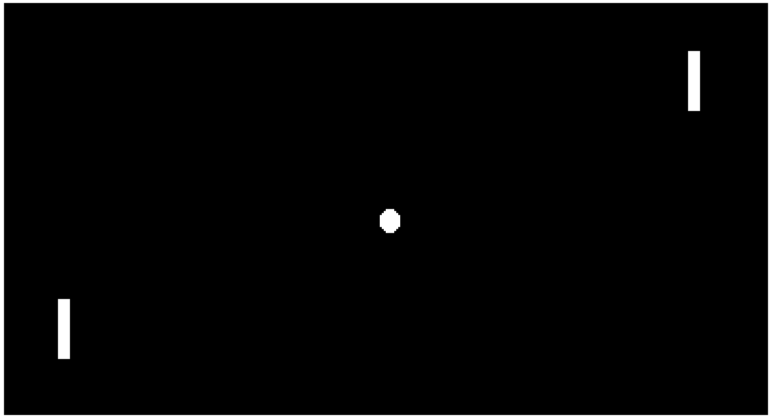


- States S
- Actions A
- Transitions: $P(s' | s, a)$

Imitation

Input: expert behavior generated by π_E

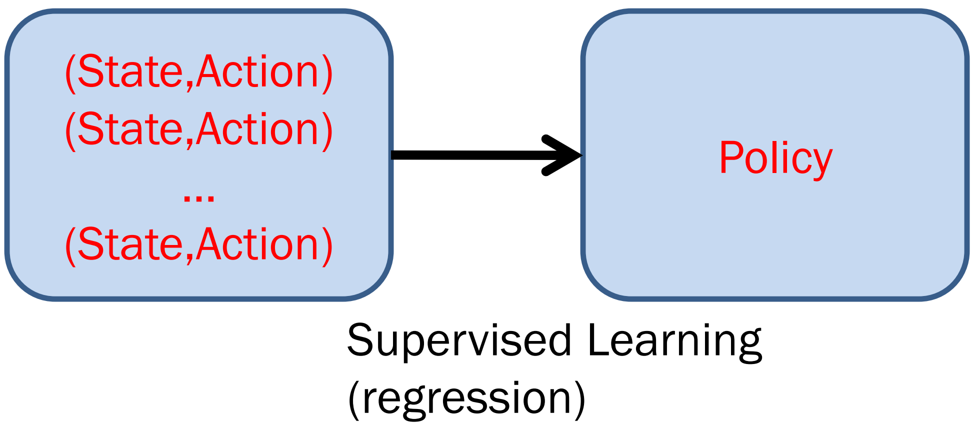
$$\{(s_0^i, a_0^i, s_1^i, a_1^i, \ldots)\}_{i=1}^n \sim \pi_E$$



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.

Behavioral Cloning



- Small errors compound over time *(cascading errors)*
- *Decisions are purposeful (require planning)*

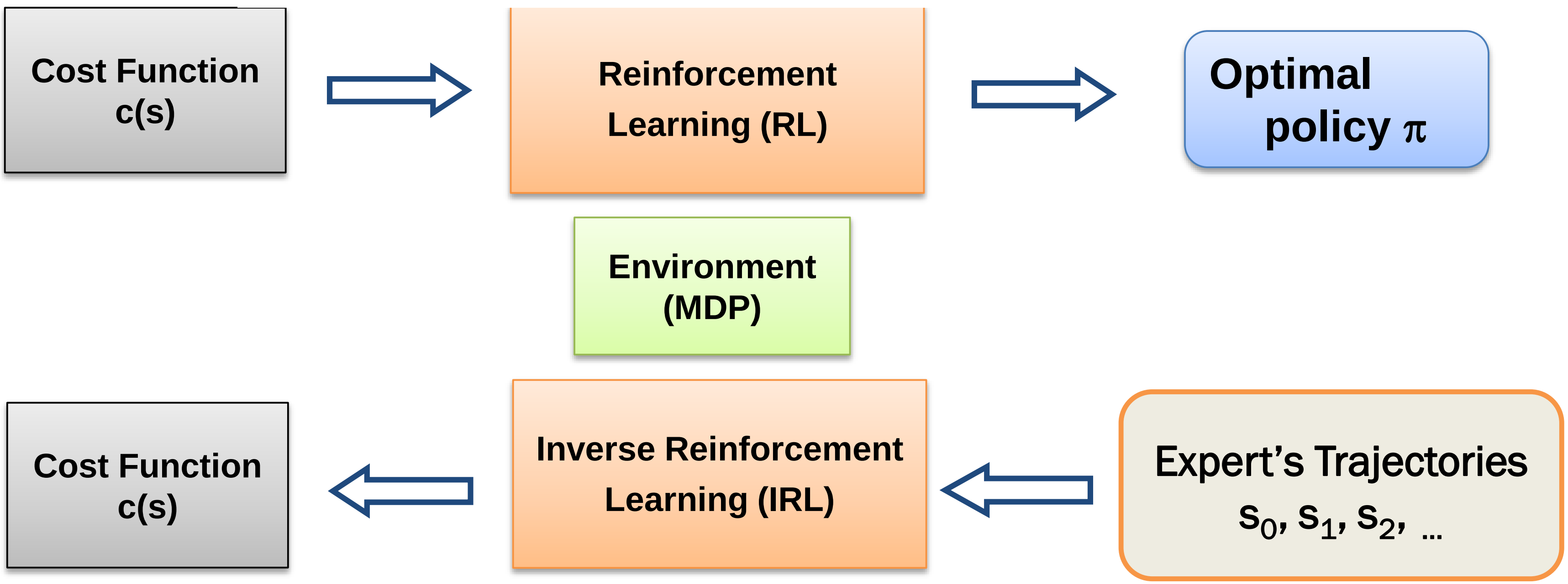
Inverse RL

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

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

Problem setup

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



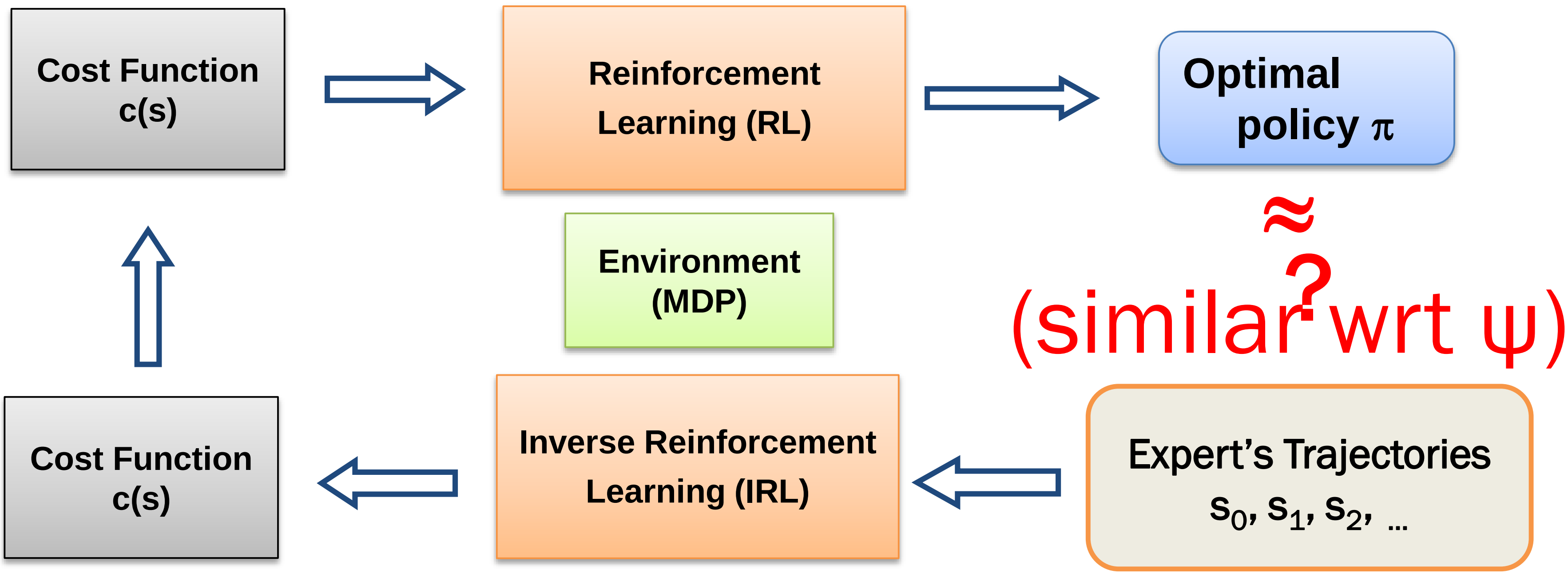
$$\underset{c \in \mathcal{C}}{\mathrm{maximize}} \left(\min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_\pi[c(s,a)] \right) - \mathbb{E}_{\pi_E}[c(s,a)]$$

Everything else has high cost

Expert has small cost

(Ziebart et al., 2010; Rust 1987)

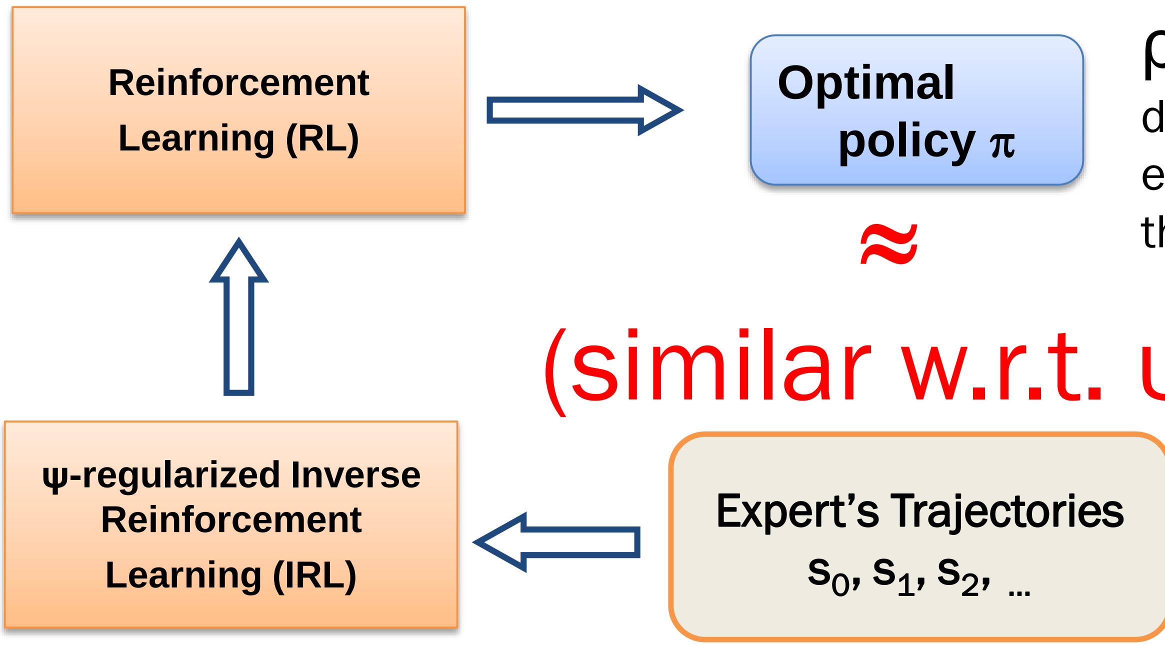
Problem setup



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

Convex cost regularizer

Combining RL∘IRL



ρ_π = 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

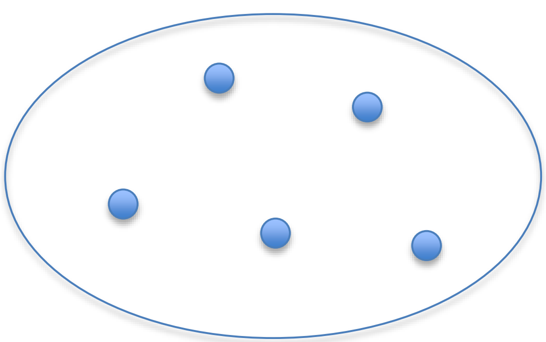
Theorem: ψ-regularized inverse reinforcement learning, implicitly, **seeks a policy whose occupancy measure is close to the expert's**, as measured by ψ^*

- 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)

Special cases

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

- If ψ(c)=constant, then $\rho_{\tilde{\pi}} = \rho_{\pi_E}$
  - Not a useful algorithm. In practice, we only have sampled trajectories
- **Overfitting:** Too much flexibility in choosing the cost function (and the policy)



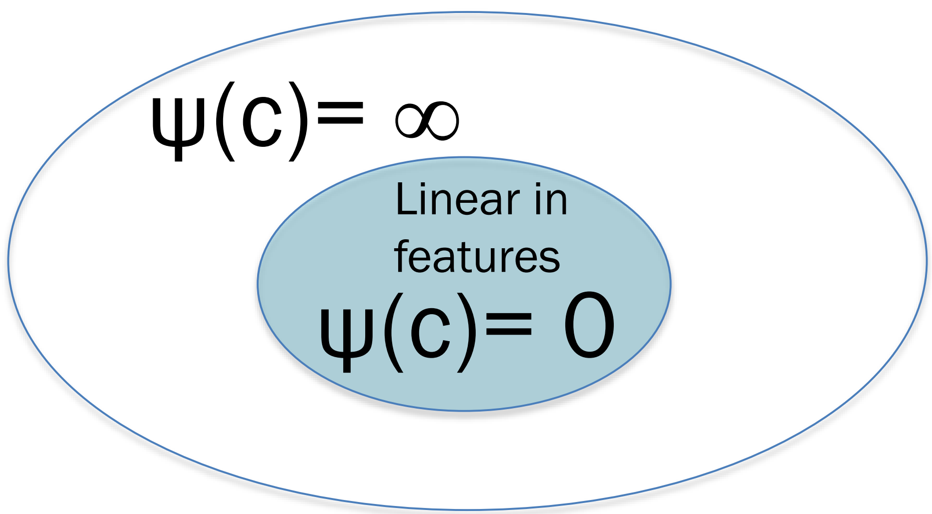
All cost functions ψ(c)=constant

Towards Apprenticeship learning

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

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

Only these “simple” cost functions are allowed



All cost functions

Apprenticeship learning

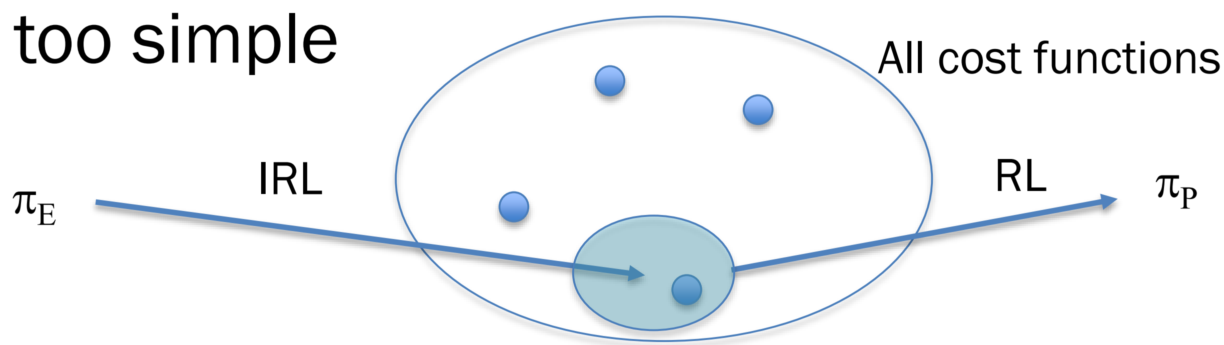
- For that choice of ψ, $\mathrm{RL} \circ \mathrm{IRL}_{\psi}$ framework gives apprenticeship learning

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

- Apprenticeship learning: find π performing better than π_E over costs linear in the features
  - Abbeel and Ng (2004)
  - Syed and Schapire (2007)

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

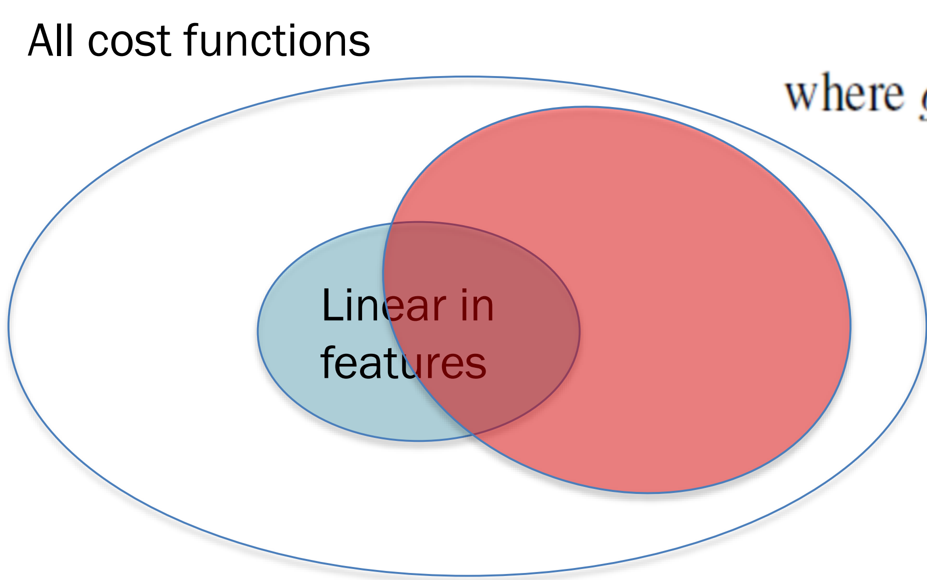


Generative Adversarial Imitation Learning

- **Solution:** use a more expressive class of cost functions

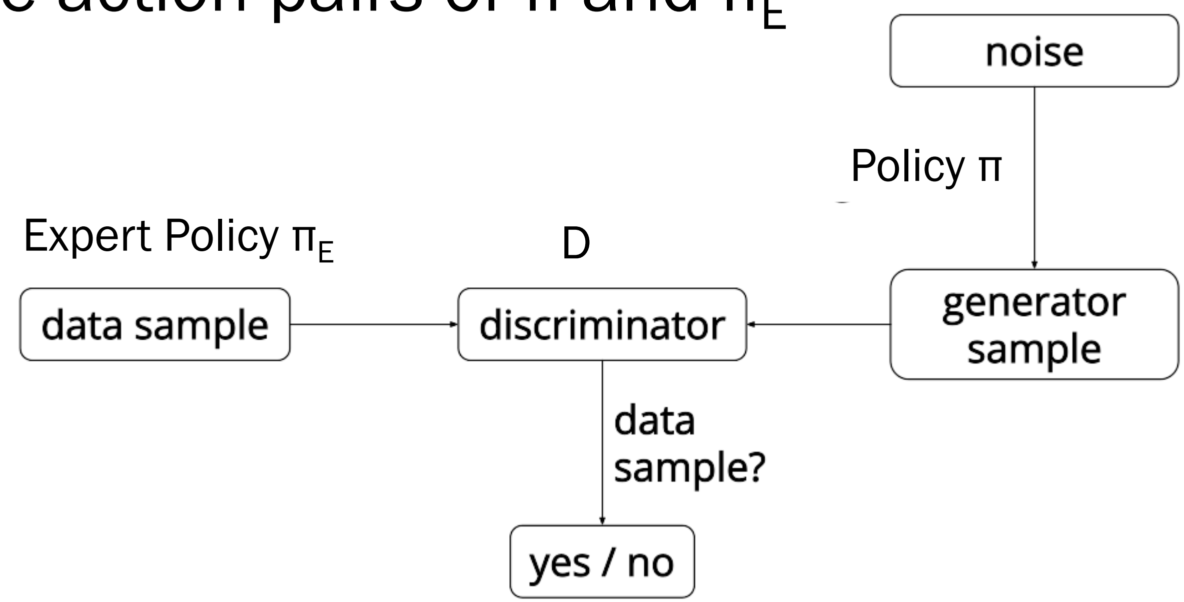
$$\psi_{\mathrm{GA}}(c) \triangleq \begin{cases} \mathbb{E}_{\pi_E}[g(c(s,a))] & \text{if } c < 0 \\ +\infty & \text{otherwise} \end{cases}$$

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



Generative Adversarial Imitation Learning

- ψ* = optimal negative log-loss of the binary classification problem of distinguishing between state-action pairs of π and π_E



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

Generative Adversarial Networks

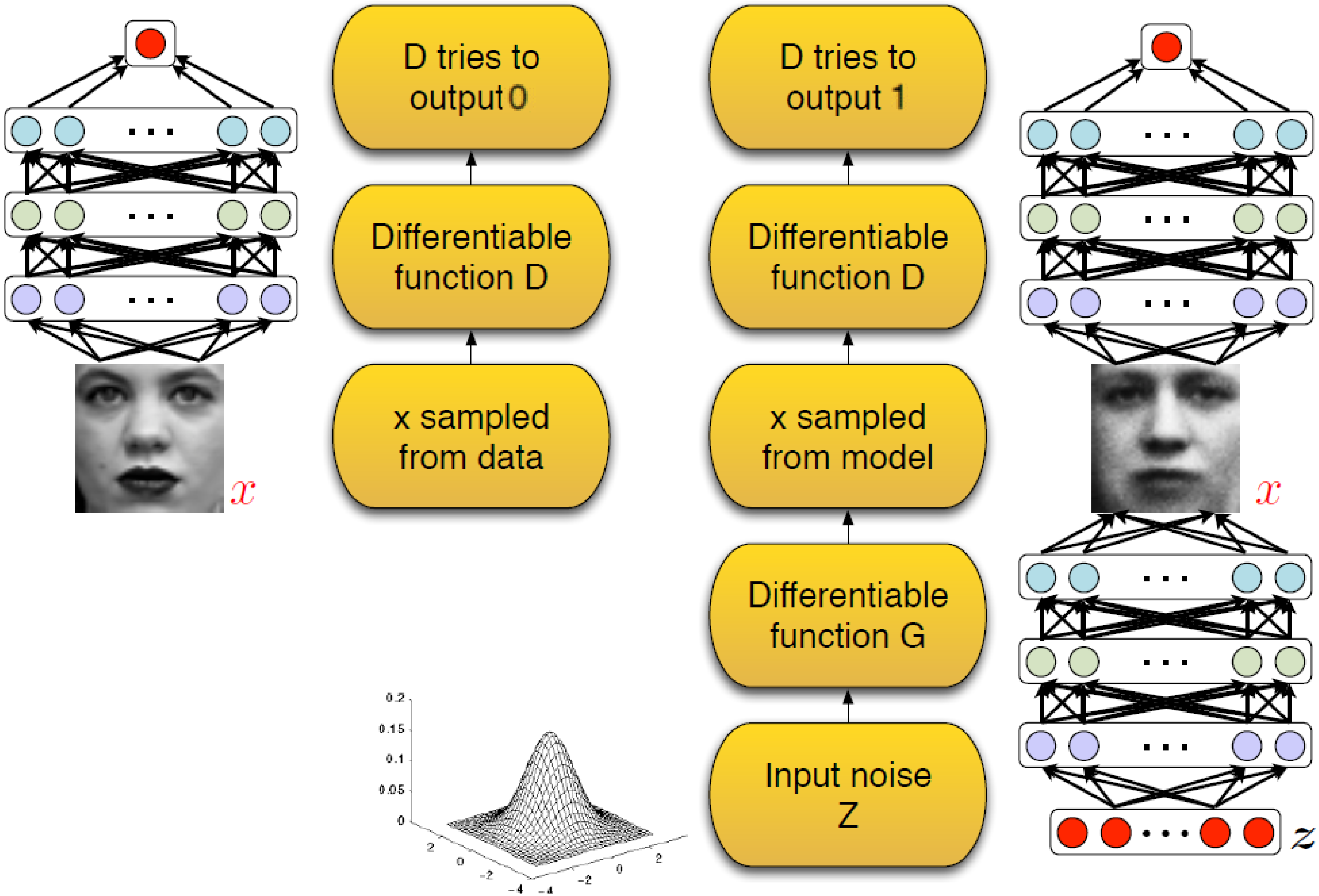
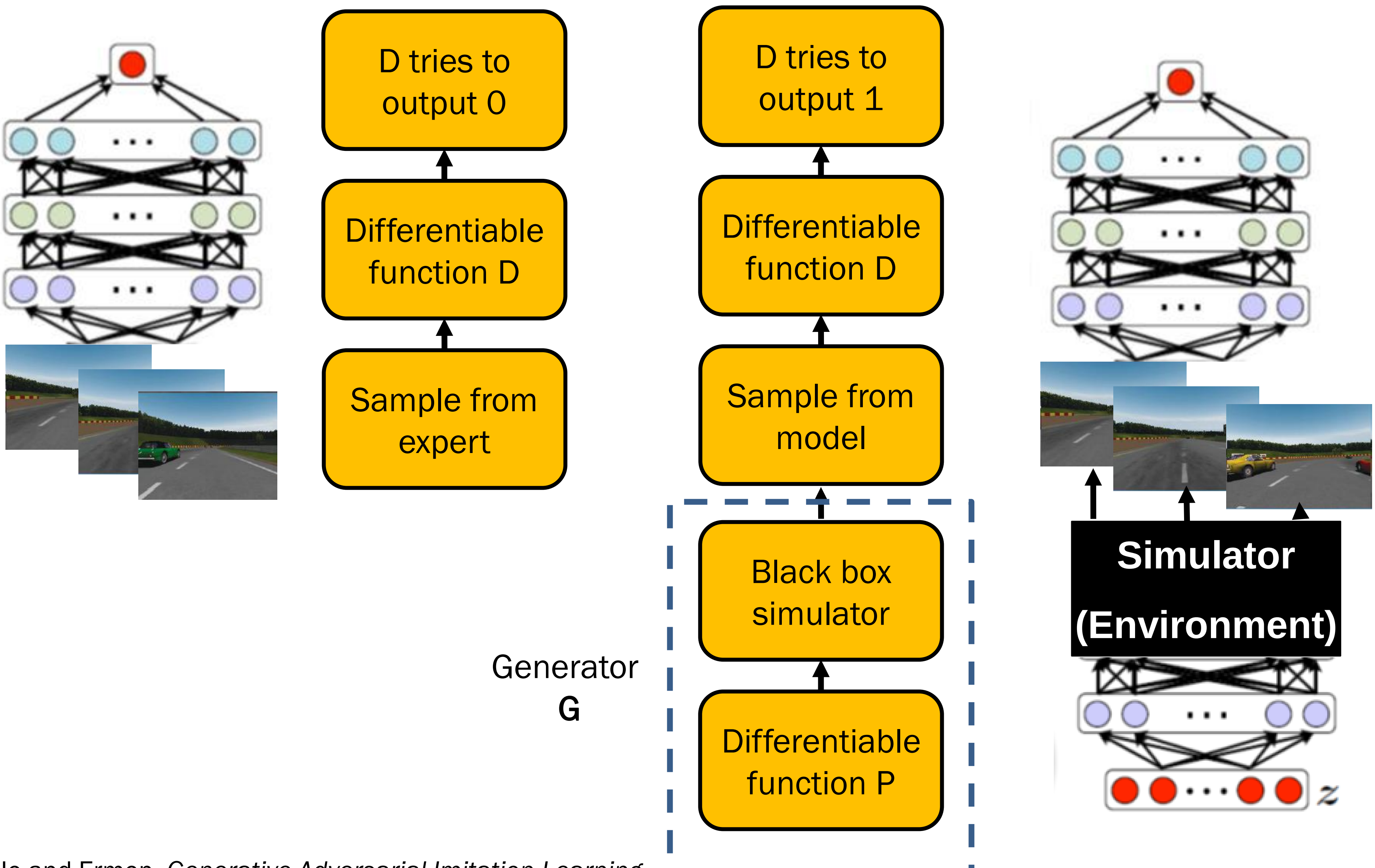


Figure from Goodfellow et al, 2014

GAIL



Ho and Ermon, *Generative Adversarial Imitation Learning*

How to optimize the objective

- Previous Apprenticeship learning work:
  - Full dynamics model
  - Small environment
  - Repeated RL
- We propose: gradient descent over policy parameters (and discriminator)

J. Ho, J. K. Gupta, and S. Ermon. Model-free imitation learning with policy optimization. ICML 2016.

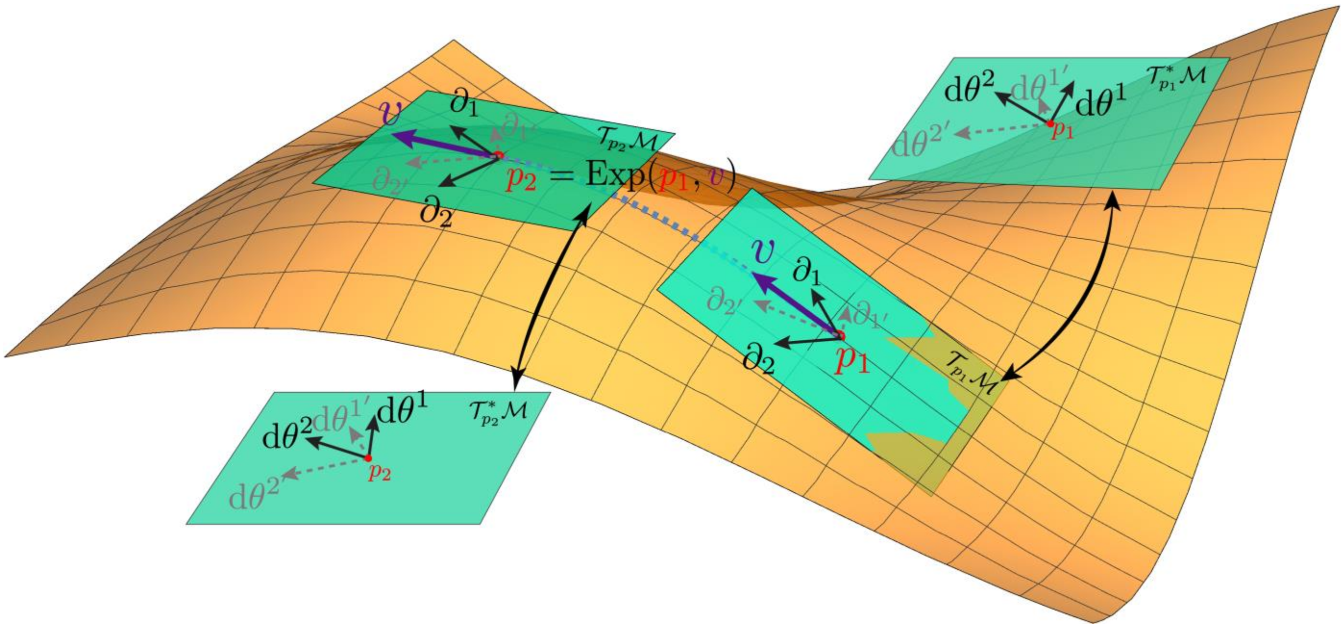
Properties

- Inherits pros of policy gradient
  – Convergence to local minima
  – Can be model free
- Inherits cons of policy gradient
  – High variance
  – Small steps required

Properties

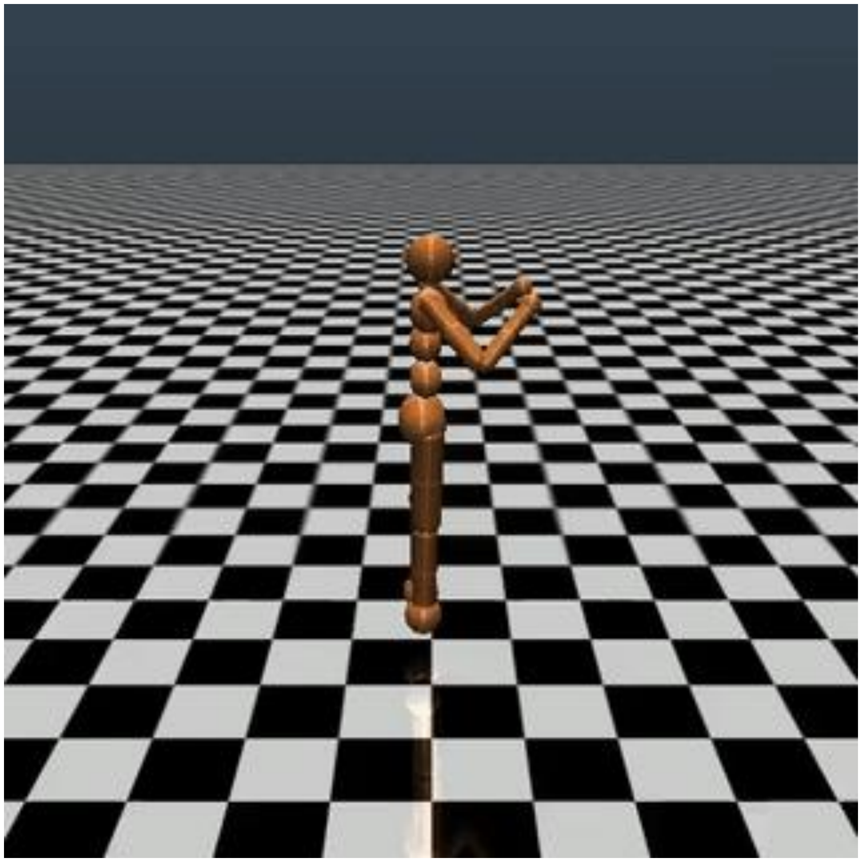
- Inherits pros of policy gradient
  - Convergence to local minima
  - Can be model free
- Inherits cons of policy gradient
  - High variance
  - Small steps required
- **Solution: trust region policy optimization**

TRPO



$p_2 = \mathrm{Exp}(p_1, v)$
$\mathcal{T}_{p_2}\mathcal{M}$
$\mathcal{T}_{p_1}\mathcal{M}$
$\mathcal{T}^*_{p_2}\mathcal{M}$
$\mathcal{T}^*_{p_1}\mathcal{M}$
v
∂_1
∂_2
$\partial_{1'}$
$\partial_{2'}$
p_1
p_2
$\mathrm{d}\theta^1$
$\mathrm{d}\theta^2$
$\mathrm{d}\theta^{1'}$
$\mathrm{d}\theta^{2'}$

Results



Results

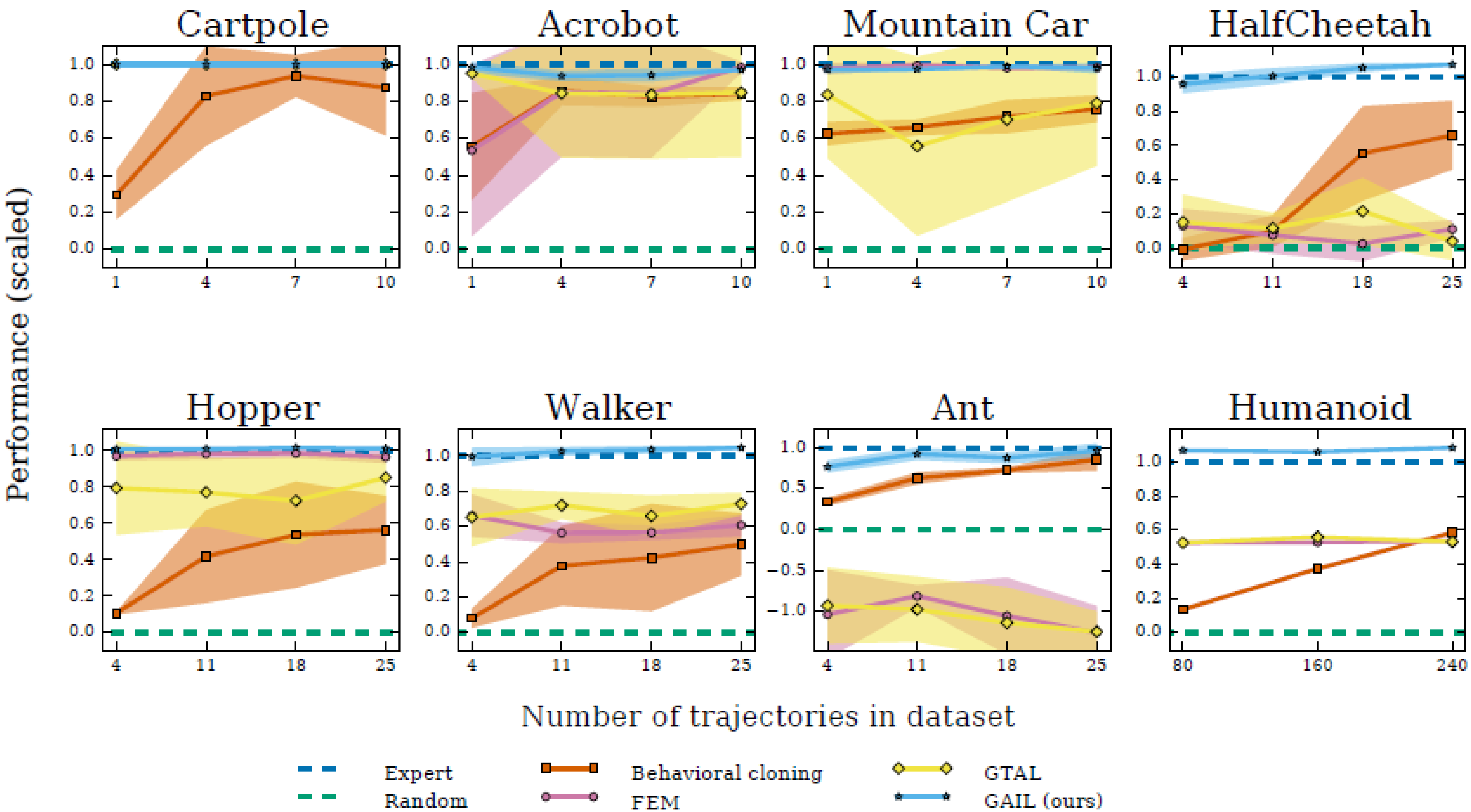
Input: driving demonstrations (Torcs)

Output policy:

From raw visual inputs

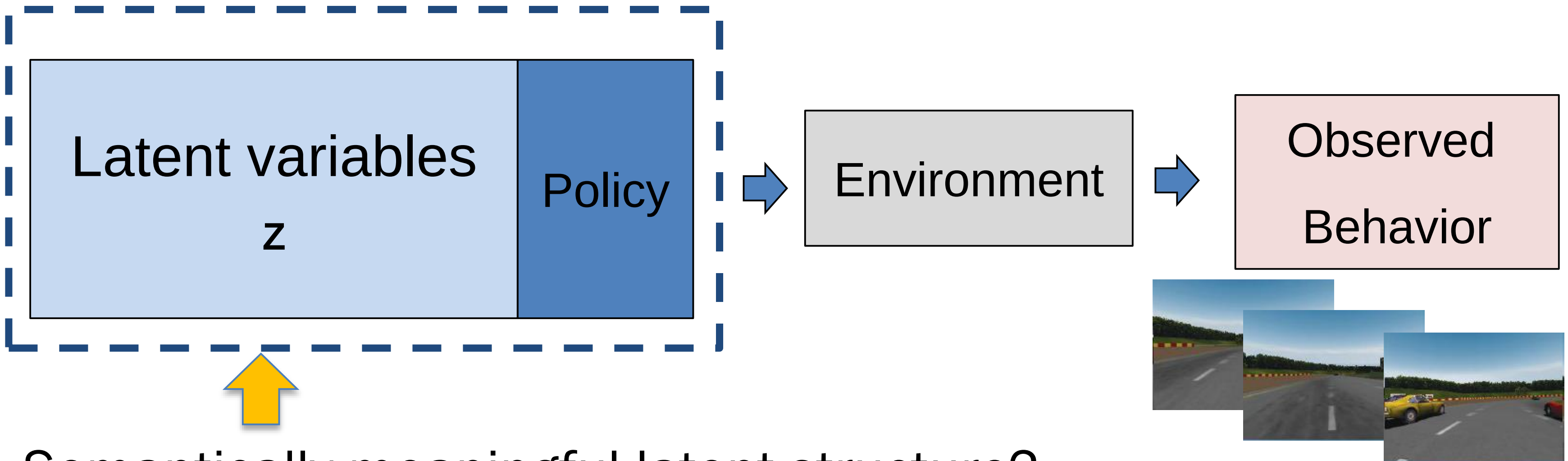
Li et al, 2017. InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations

Experimental results

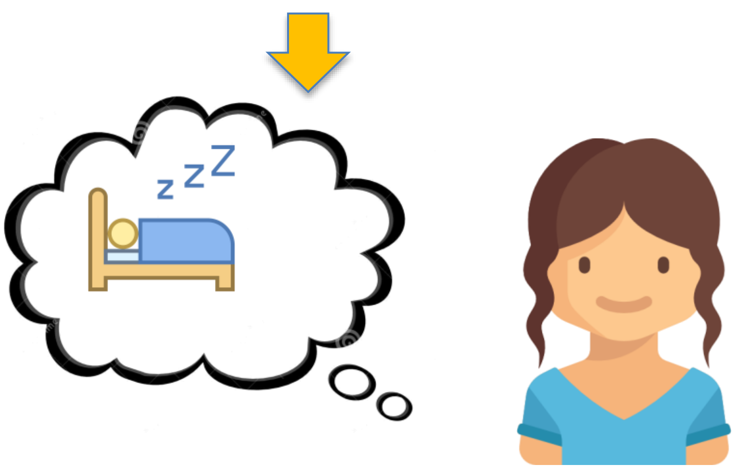


Latent structure in demonstrations

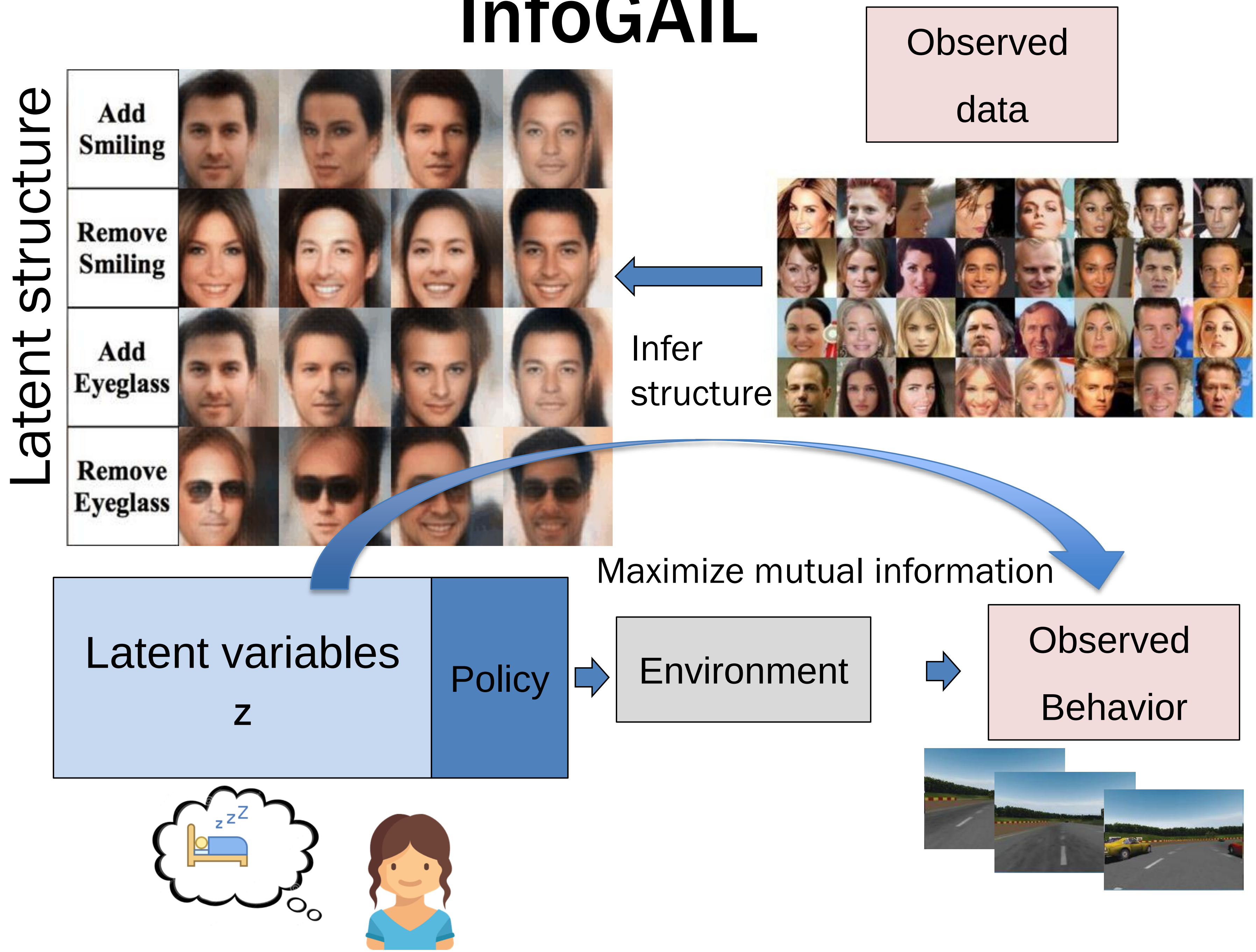
Human model



Semantically meaningful latent structure?

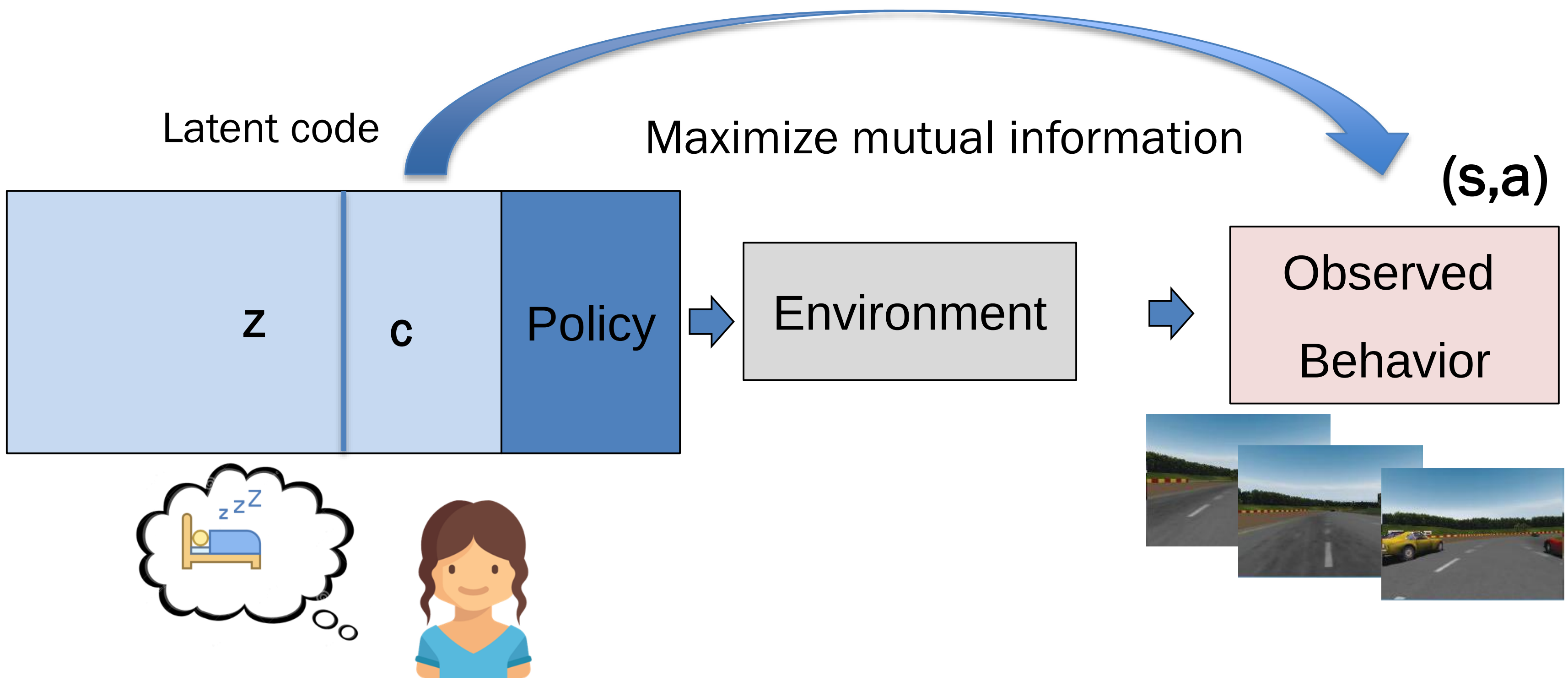


InfoGAIL

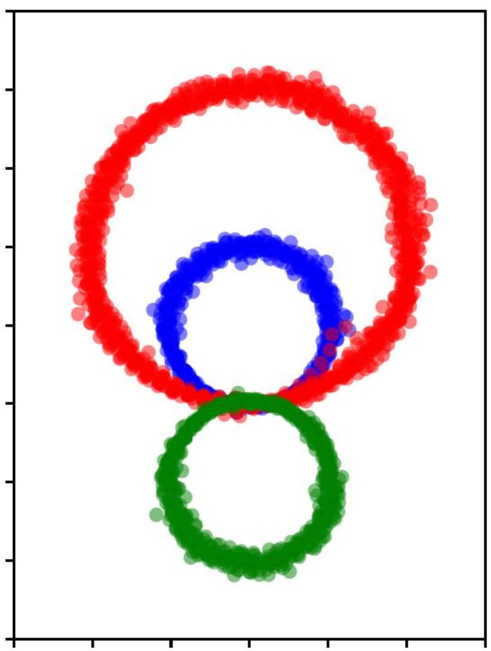

Observed data
Latent structure
Add Smiling
Remove Smiling
Add Eyeglass
Remove Eyeglass
Infer structure
Maximize mutual information
Latent variables z
Policy
Environment
Observed Behavior

InfoGAIL

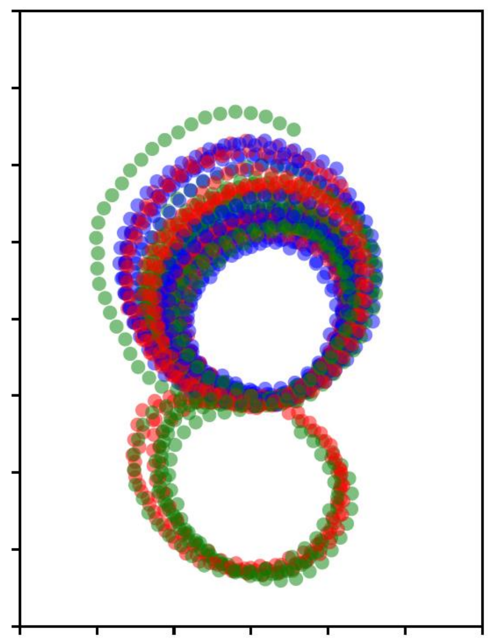
$$
\begin{aligned}
L_I(\pi_\theta, Q_\psi) &= \mathbb{E}_{c\sim p(c), a\sim \pi_\theta(\cdot|s,c)}[\log Q_\psi(c|s,a)] + H(c) \\
&\leq I(c; s, a)
\end{aligned}
$$



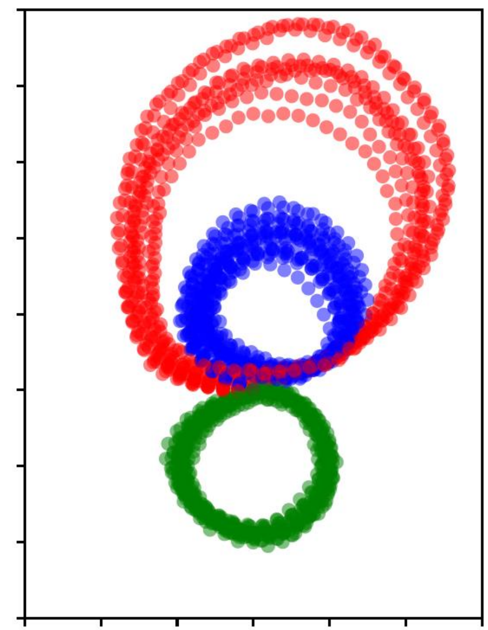
Synthetic Experiment



Demonstrations



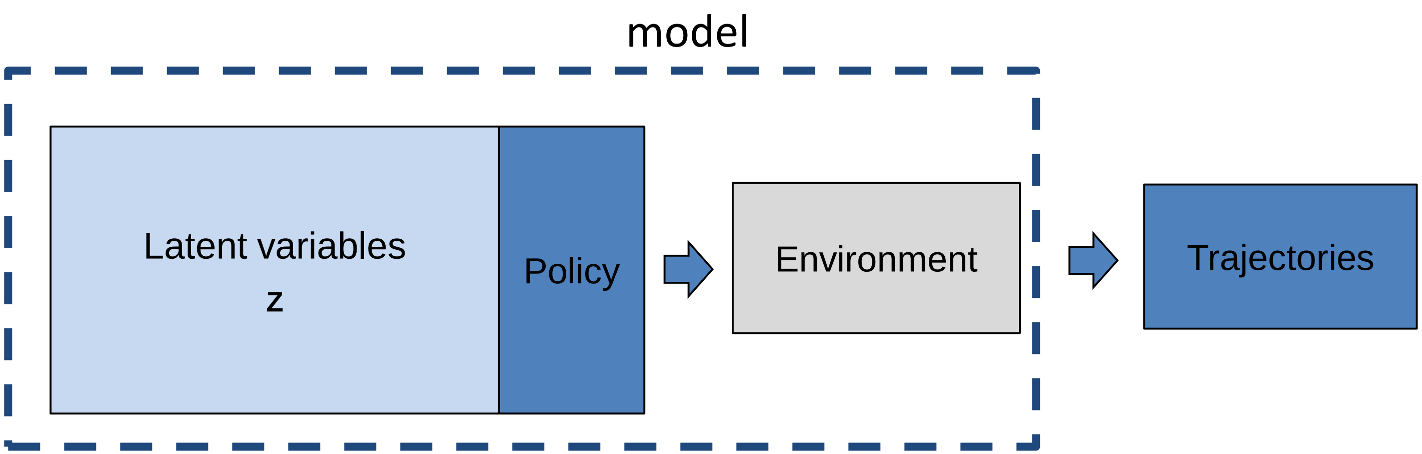
GAIL



Info-GAIL

Li et al, 2017. InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations

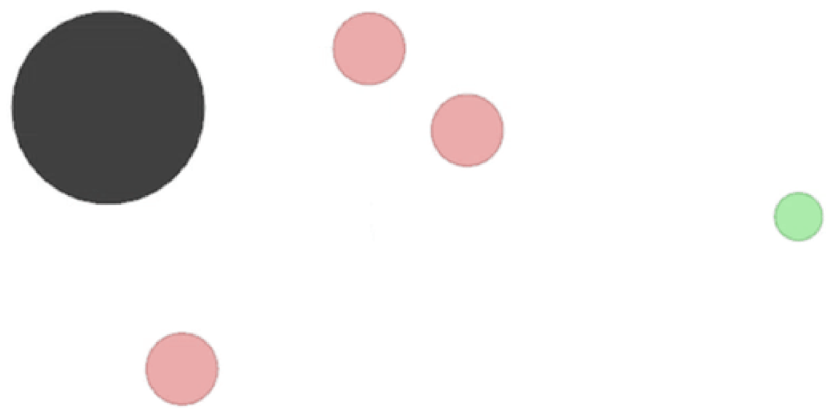
InfoGAIL



Pass left (z=0)

Pass right (z=1)

Multi-agent environments



What are the goals of these 4 agents?

Problem setup

Cost Functions
$c_1(s,a_1)$
..
$c_N(s,a_N)$

MA Reinforcement Learning (MARL)

Environment (Markov Game)

Optimal policies $\pi 1$

...

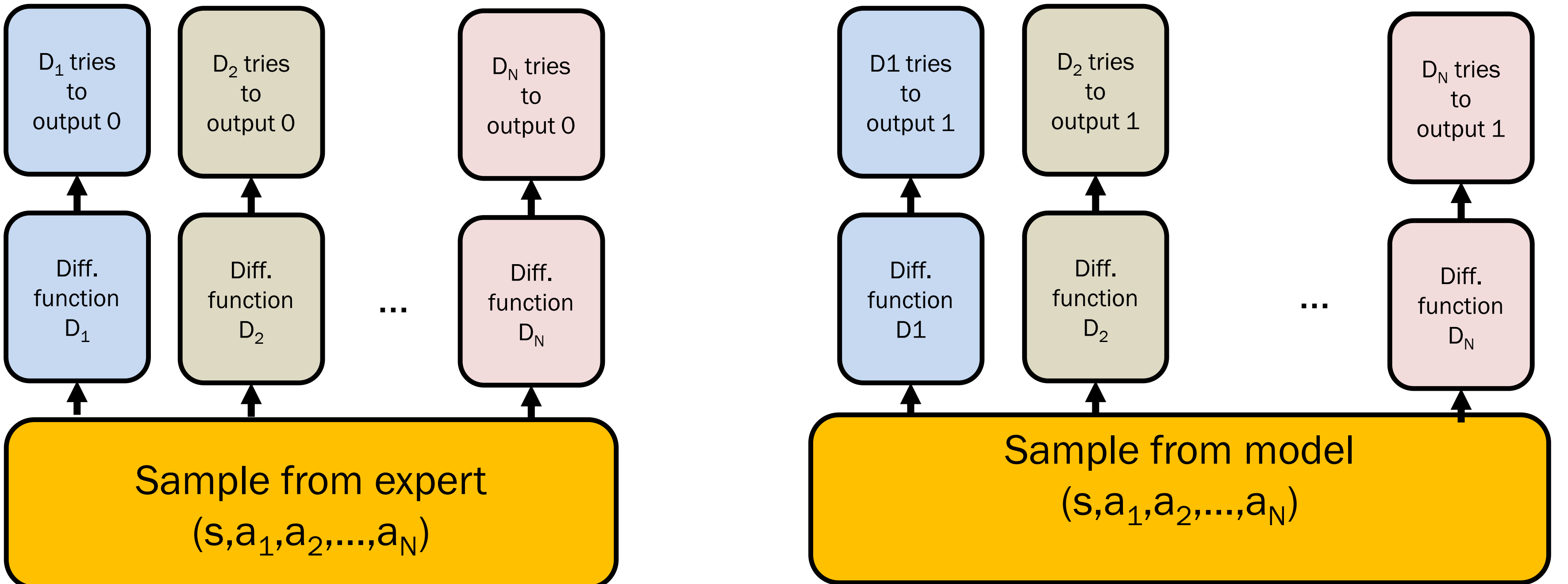
Optimal policies πK

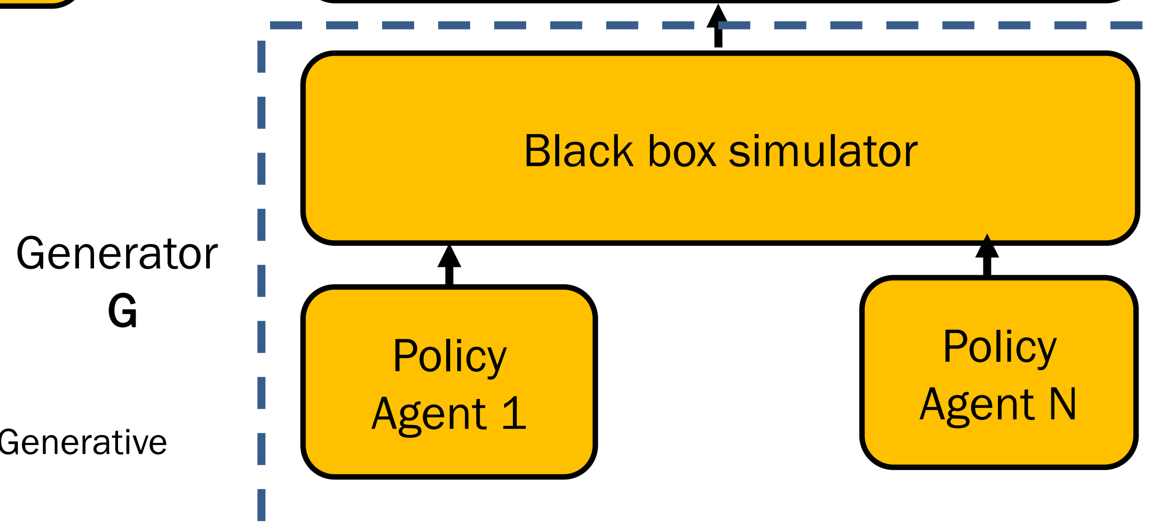
	R	L
R	0,0	10,10
L	10,10	0,0

DRIVE ON LEFT

DRIVE ON RIGHT

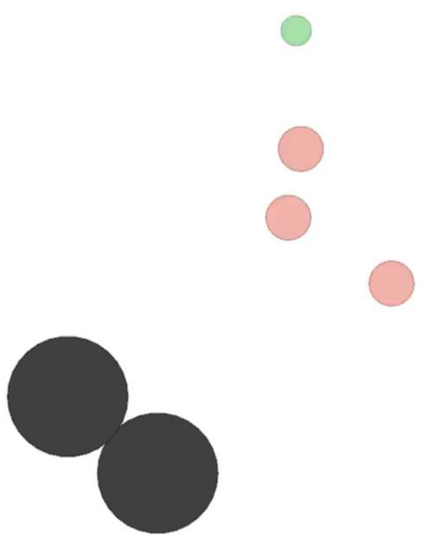
MAGAIL



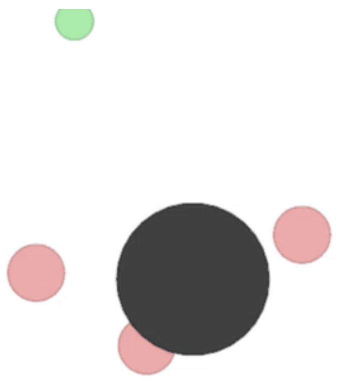


Song, Ren, Sadigh, Ermon, Multi-Agent Generative Adversarial Imitation Learning

Environments



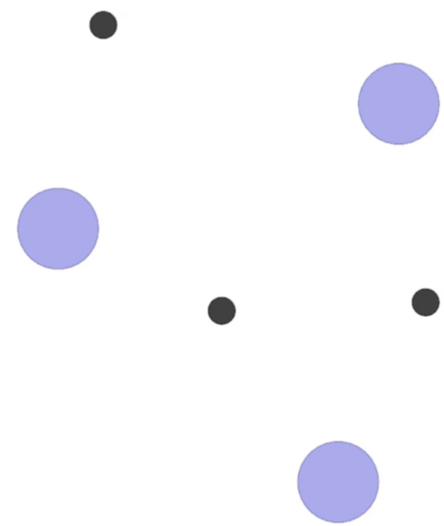
Demonstrations



MAGAIL

Environments

Demonstrations



MAGAIL

Conclusions

- IRL is a dual of an occupancy measure matching problem (generative modeling)
- Might need flexible cost functions
  - GAN style approach
- Policy gradient approach
  - Scales to high dimensional settings
- Towards unsupervised learning of latent structure from demonstrations