Generative Adversarial Imitation Learning

Sep 7, 2020

Hoe Sung Ryu

CONTENTS

Imitation Learning

Generative Adversarial Imitation Learning

Jonathan Ho
OpenAI
hoj@openai.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Abstract

Consider learning a policy from example expert behavior, without interaction with the expert or access to a reinforcement signal. One approach is to recover the expert's cost function with inverse reinforcement learning, then extract a policy from that cost function with reinforcement learning. This approach is indirect and can be slow. We propose a new general framework for directly extracting a policy from data as if it were obtained by reinforcement learning following inverse reinforcement learning. We show that a certain instantiation of our framework draws an analogy between imitation learning and generative adversarial networks, from which we derive a model-free imitation learning algorithm that obtains significant performance gains over existing model-free methods in imitating complex behaviors in large, high-dimensional environments.

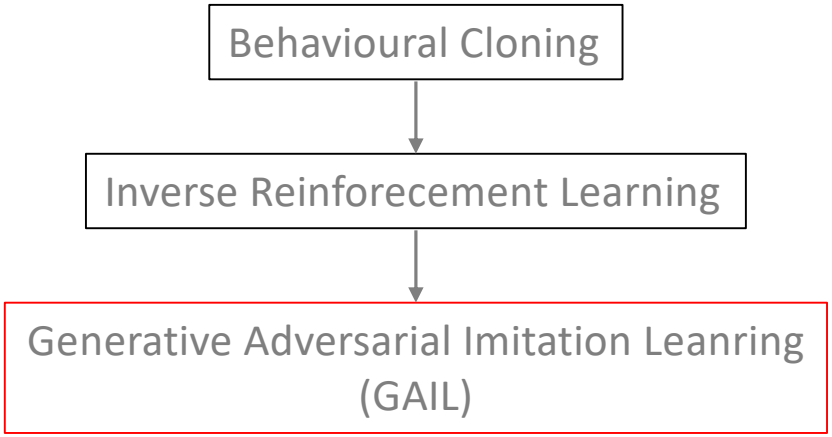
01 Imitation Learning

Imitation learning

There are several RL algorithms which use the recieved rewards as t he main approach to approximate the best policy. However, rewards are sparse thus we manually designed the rewards functions to prov ide the agent with more frequent rewards but designing a reward fu nction can be complicated.

So a feasible solution to this problem is imitation learning

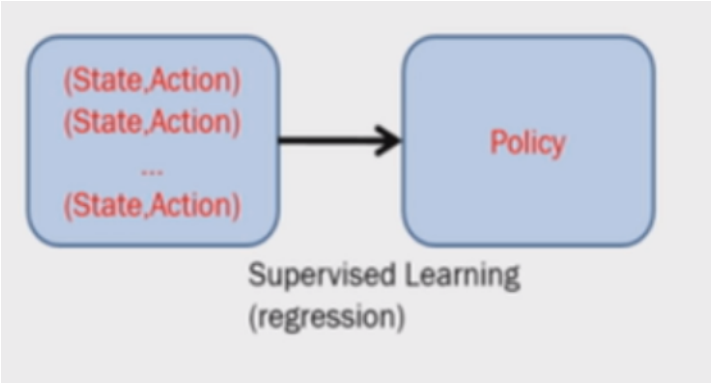
Type of Imitation Learning



Behavioral Cloning

- Given the expert's demonstrations, we divide these into state-act ion pairs, we treat these pairs as i.i.d[(1)] examples
- Then, we apply supervised learning. The loss function can depen d on the application .
- Therefore, the algorithm is the following:

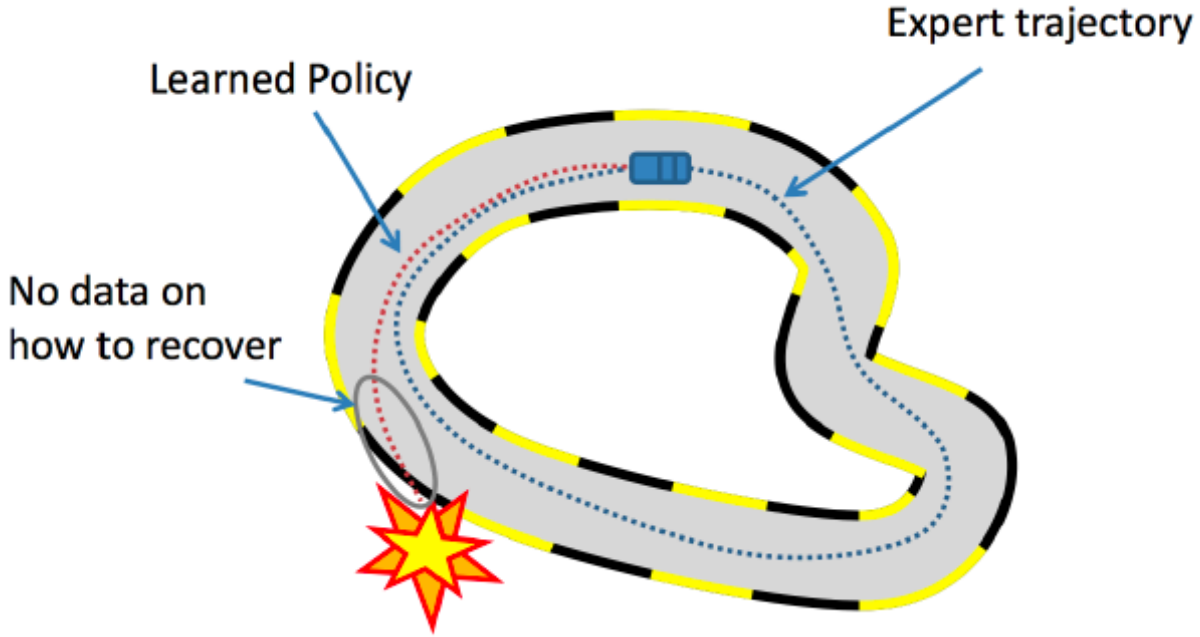
1. Collect demonstrations (τ^* trajectories) from expert
2. Treat the demonstrations as i.i.d. state-action pairs: $(s_0^*, a_0^*), (s_1^*, a_1^*), \ldots$
3. Learn π_θ policy using supervised learning by minimizing the loss function $L\left(a^*, \pi_\theta(s)\right)$



(1) i.i.d(independent identically distributed)

Limitation of BL

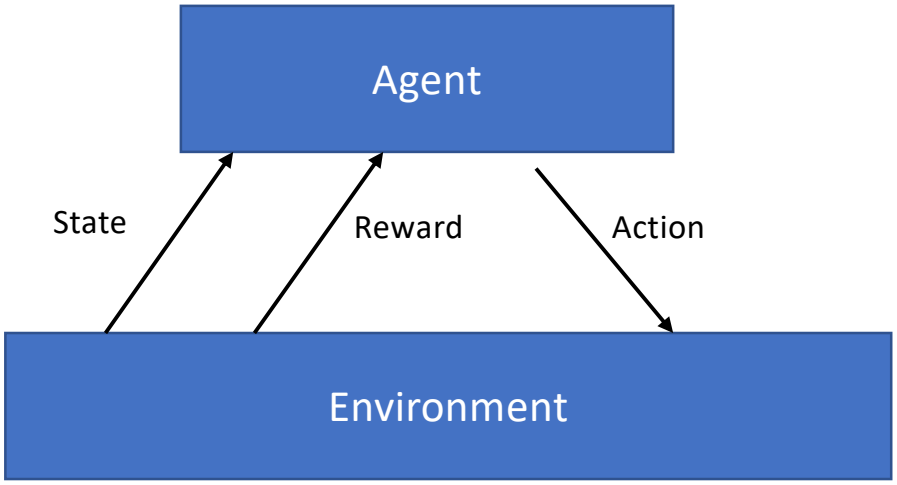
- Behavioral cloning can be problematic because of i.i.d. assumption
- In MDP an action in a given state induces the next state, which breaks the previous assumption. This means errors made in different states a dd up so the agent can easily put it into a state that the expert has nev er visited.



Inverse Reinforcement Learning

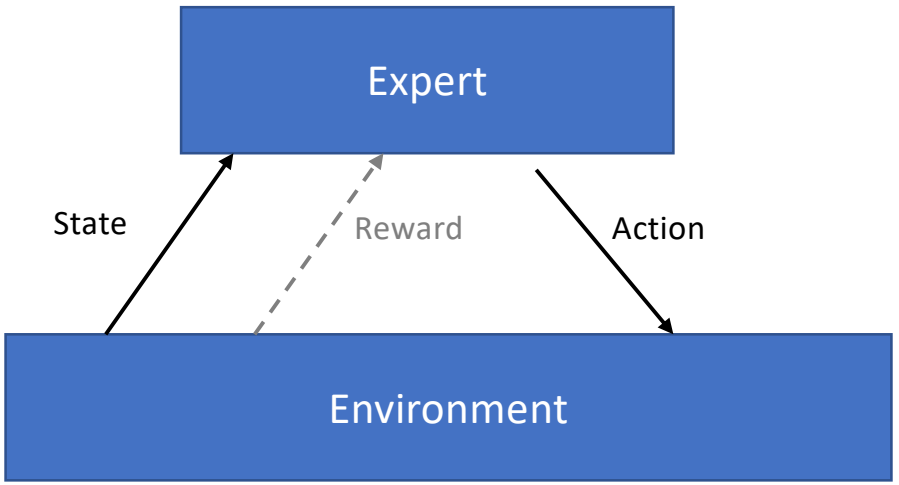
Inverse reinforcement learning (IRL) is a different approach of imitat ion learning, where the main idea is to learn the reward function of the environment based on the expert's demonstrations, and then fi nd the optimal policy that maximizes this reward function using rein forcement learning.

Reinfocement leanring



Goal: Find policy π that maximize rewards

Inverse Reinfocement leanring

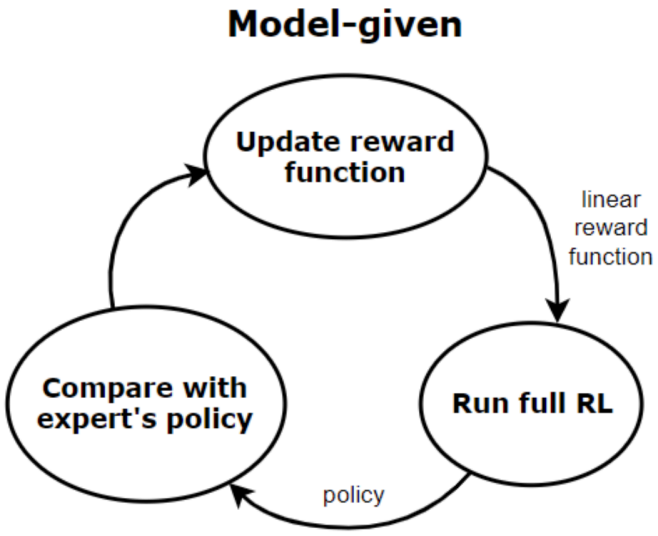


Goal: Find reward function that expert is implicitly optimizing

Inverse Reinforcement Learning

- Model-given

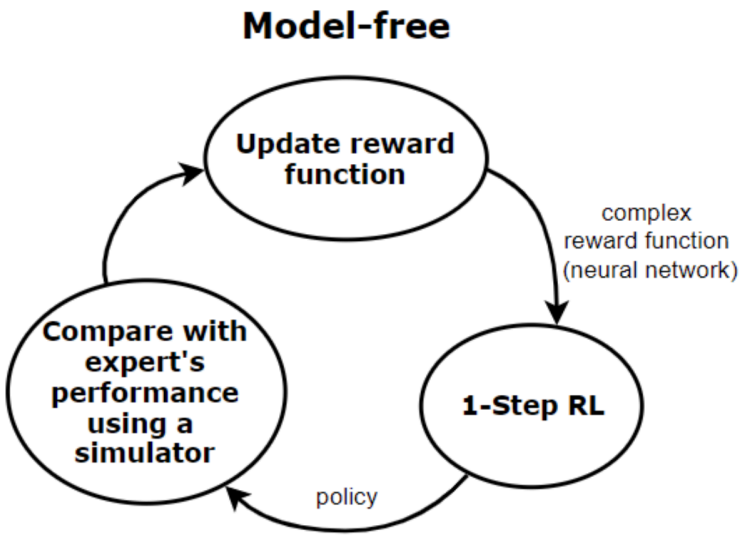
  Our reward function is linear. In each iteration, we need to solve the full R L problem, so to be able to do this efficiently, we assume, that the enviro nment's (the MDP's) state space is small. We also suppose that we know the state transition dynamics of the environment. This is needed so that we can compare our learned policy with the expert's one effectively.



Inverse Reinforcement Learning

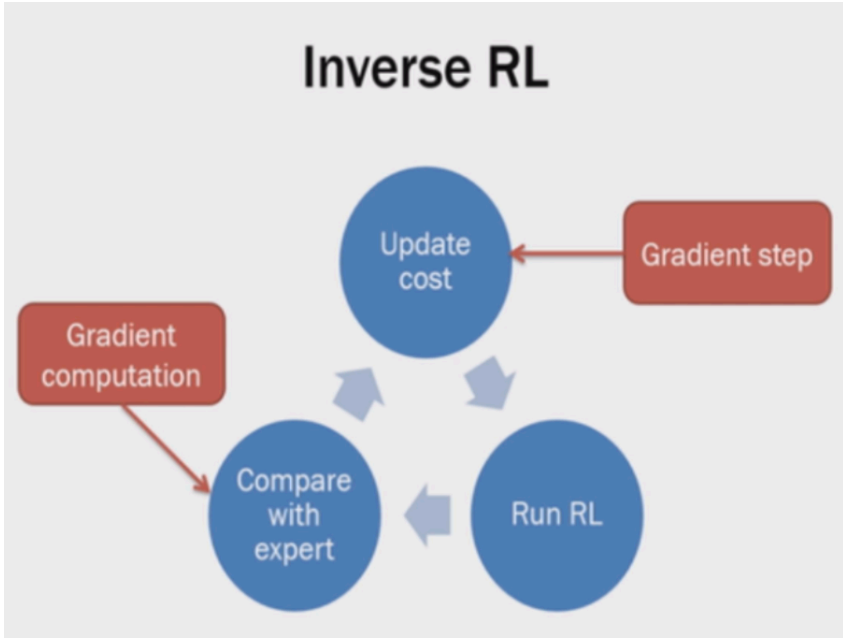
- Model-free

  The **model-free** approach is a more general case. Here, we assume that t he reward function is more complex, which we usually model with a neu ral network. We also suppose that the state space of the MDP is large or continuous, therefore in each iteration, we only solve a single step of the RL problem.



Limitation of IRL

1. Indirectly learns optimal policy from the function (using RL)
2. Expensive to run(Inner loop has RL)

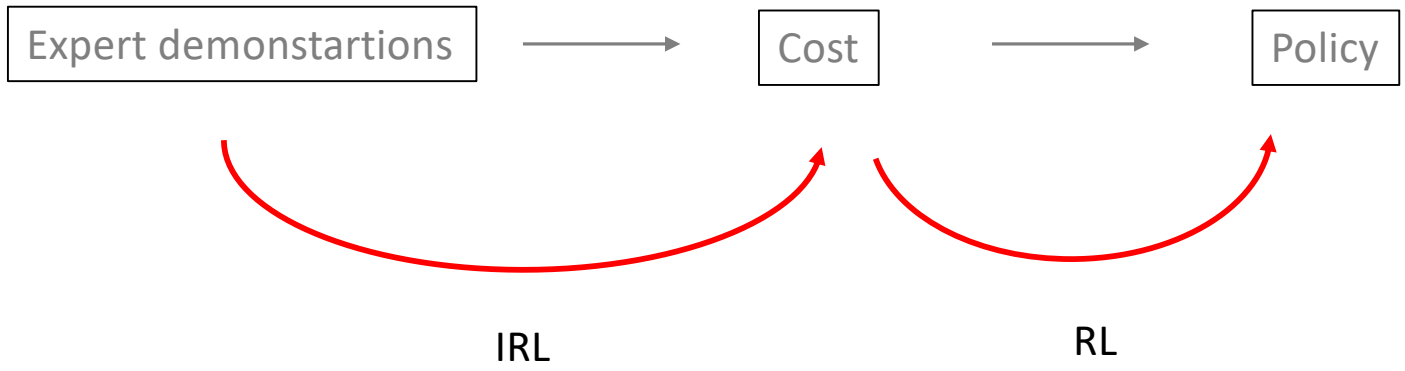


Preliminaries

02 Preliminaries

Preliminaries $\overline{\mathbb{R}}$ will denote the extended real numbers $\mathbb{R} \cup \{\infty\}$. Section 3 will work with finite state and action spaces $\mathcal{S}$ and $\mathcal{A}$, but our algorithms and experiments later in the paper will run in high-dimensional continuous environments. Π is the set of all stationary stochastic policies that take actions in $\mathcal{A}$ given states in $\mathcal{S}$; successor states are drawn from the dynamics model $P(s'|s,a)$. We work in the γ-discounted infinite horizon setting, and we will use an expectation with respect a policy $\pi \in \Pi$ to denote an expectation with respect to the trajectory it generates: $\mathbb{E}_\pi[c(s,a)] \triangleq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t c(s_t,a_t)\right]$, where $s_0 \sim p_0$, $a_t \sim \pi(\cdot|s_t)$, and $s_{t+1} \sim P(\cdot|s_t,a_t)$ for $t \geq 0$. We will use $\hat{\mathbb{E}}_\tau$ to denote an empirical expectation with respect to trajectory samples τ, and we will always use π_E to refer to the expert policy.

Inverse Reinforcement Learning



Inverse Reinforcement Learning

- Maximum casual entropy IRL (Expert demonstrations → cost)

  Try to find a cost function $c \in C$ that assigns low cost to the expert policy π_E and high cost to other policies π

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

- RL procedure (cost → policy)

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

Proposed Method

03 Proposed Method

Main idea of Generative Adversarial Imitaion Learning(GAIL)

1. Directly extracting a policy form data as if it were obtained by RL following IRL.
2. Draws an analogy between imitation learning and generative adv ersarial networks(GAN)
3. Derive a model-free imitationn leanring algorithm with significan t performance improvement with low sample and computational complexity

Generative Adversarial Imitation Learning

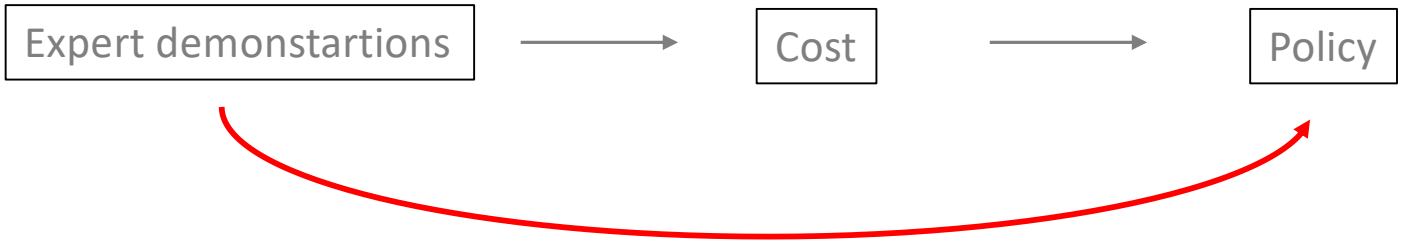
Jonathan Ho
OpenAI
hoj@openai.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Abstract

Consider learning a policy from example expert behavior, without interaction with the expert or access to a reinforcement signal. One approach is to recover the expert's cost function with inverse reinforcement learning, then extract a policy from that cost function with reinforcement learning. This approach is indirect and can be slow. We propose a new general framework for directly extracting a policy from data as if it were obtained by reinforcement learning following inverse reinforcement learning. We show that a certain instantiation of our framework draws an analogy between imitation learning and generative adversarial networks, from which we derive a model-free imitation learning algorithm that obtains significant performance gains over existing model-free methods in imitating complex behaviors in large, high-dimensional environments.

Directly extracting a policy



Directly extracting a policy

- Use the largest possible set of cost functions

$$C = \mathbb{R}^{S\times A} = \{c: S\times A \to \mathbb{R}\}$$

- Such a large C, IRL easily overfit thus we use a convex costfunction regularizer ψ:

$$\psi: \mathbb{R}^{S\times A} \to \mathbb{R}\cup\infty$$

- With ψ, IRL proceducre can be written as

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

- Let $\tilde{c} \in IRL_{\psi}(\pi_E)$, we are interseted in $\pi = RL(\tilde{c})$

Directly extracting a policy

1. Occupancy Measure

   For a policy $\pi \in \Pi$, $\rho_\pi(s,a) = \pi(a|s)\sum_{t=0}^{\infty}\gamma^t P(s_t = s|\pi)$

   It arrows us to write:

$$\mathbb{E}_\pi[c(s,a)] = \Sigma_{s,a}\rho_\pi(s,a)c(s,a) \quad (1)$$

2. Convex conjugate

   For a function f: $\mathbb{R}^{S\times A} \to \mathbb{R}\cup\infty$, its convex conjugate f^*: $\mathbb{R}^{S\times A} \to \mathbb{R}\cup\infty$ is

$$f^*(x) = \sup_{y\in\mathbb{R}^{S\times A}} x^T y - f(y) \quad (2)$$

$$\textbf{Then, } RL \odot IRL_\psi(\pi_E) = \operatorname{argmin}_{\pi\in\Pi} -H(\pi) + \psi^*(\rho_\pi - \rho_{\pi_E})$$

Policy Cost

$$\boldsymbol{RL} \odot \boldsymbol{IRL_{\psi}}(\boldsymbol{\pi_E}) = \underset{\boldsymbol{\pi \in \Pi}}{\operatorname{argmin}} -\boldsymbol{H}(\boldsymbol{\pi}) + \boldsymbol{\psi}^*\left(\boldsymbol{\rho_{\pi}} - \boldsymbol{\rho_{\pi_E}}\right)$$

Proof)

Let, $\mathrm{IRL}_{\psi}(\pi_{\mathrm{E}}) = \underset{c \in \mathbb{R}^{S \times A}}{\operatorname{argmax}} -\psi(c) + (\min_{\pi \in \Pi} -\mathrm{H}(\pi) + \mathbb{E}_{\pi}[c(s,a)]) - \mathbb{E}_{\pi_E}[c(s,a)]$

Since (*),

$$\mathrm{IRL}_{\psi}(\pi_{\mathrm{E}}) = \underset{c \in \mathbb{R}^{S \times A}}{\operatorname{argmax}} -\psi(c) + (\min_{\pi \in \Pi} -\mathrm{H}(\pi) + \Sigma_{s,a}\rho_{\pi}(s,a)c(s,a)) - \Sigma_{s,a}\rho_{\pi_E}(s,a)c(s,a)$$

By the definition of (**),

$$f^*(x) = \sup_{y \in \mathbb{R}^{S \times A}} x^T y - f(y)$$

$$\Leftrightarrow \psi^*(x) = \sup_{c \in \mathbb{R}^{S \times A}} x^T c - \psi(c)$$

$$\Leftrightarrow \psi^*(\boldsymbol{\rho_{\pi}} - \boldsymbol{\rho_{\pi_E}}) = \sup_{c \in \mathbb{R}^{S \times A}} \left(\boldsymbol{\rho_{\pi}} - \boldsymbol{\rho_{\pi_E}}^T\right)c - \psi(c)$$

Therefore, $\underset{\boldsymbol{\pi \in \Pi}}{\operatorname{argmin}} -\boldsymbol{H}(\boldsymbol{\pi}) + \boldsymbol{\psi}^*\left(\boldsymbol{\rho_{\pi}} - \boldsymbol{\rho_{\pi_E}}\right)$

GAN and GAIL

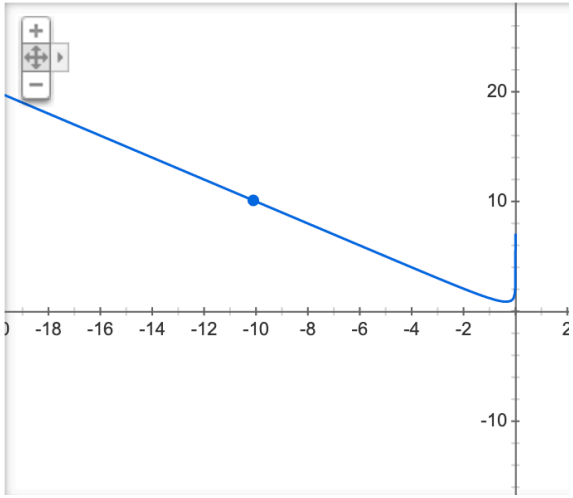
- Generative Adversarial Imitation learning

  We propose the following new cost regularizer:

$$\psi_{\mathrm{GA}}(c) \triangleq \begin{cases} \mathbb{E}_{\pi_E}[g(c(s,a))] & \text{if } c < 0 \\ +\infty & \text{otherwise} \end{cases} \quad \text{where } g(x) = \begin{cases} -x - \log(1 - e^x) & \text{if } x < 0 \\ +\infty & \text{otherwise} \end{cases}$$

  - x is far from 0 : low penalty
  - x is close to 0 : high penalty

Graph for (-x)-log(1-e^x)



GAN and GAIL

- Our choice of ψ^*_{GA} is motivated by the folowing fact

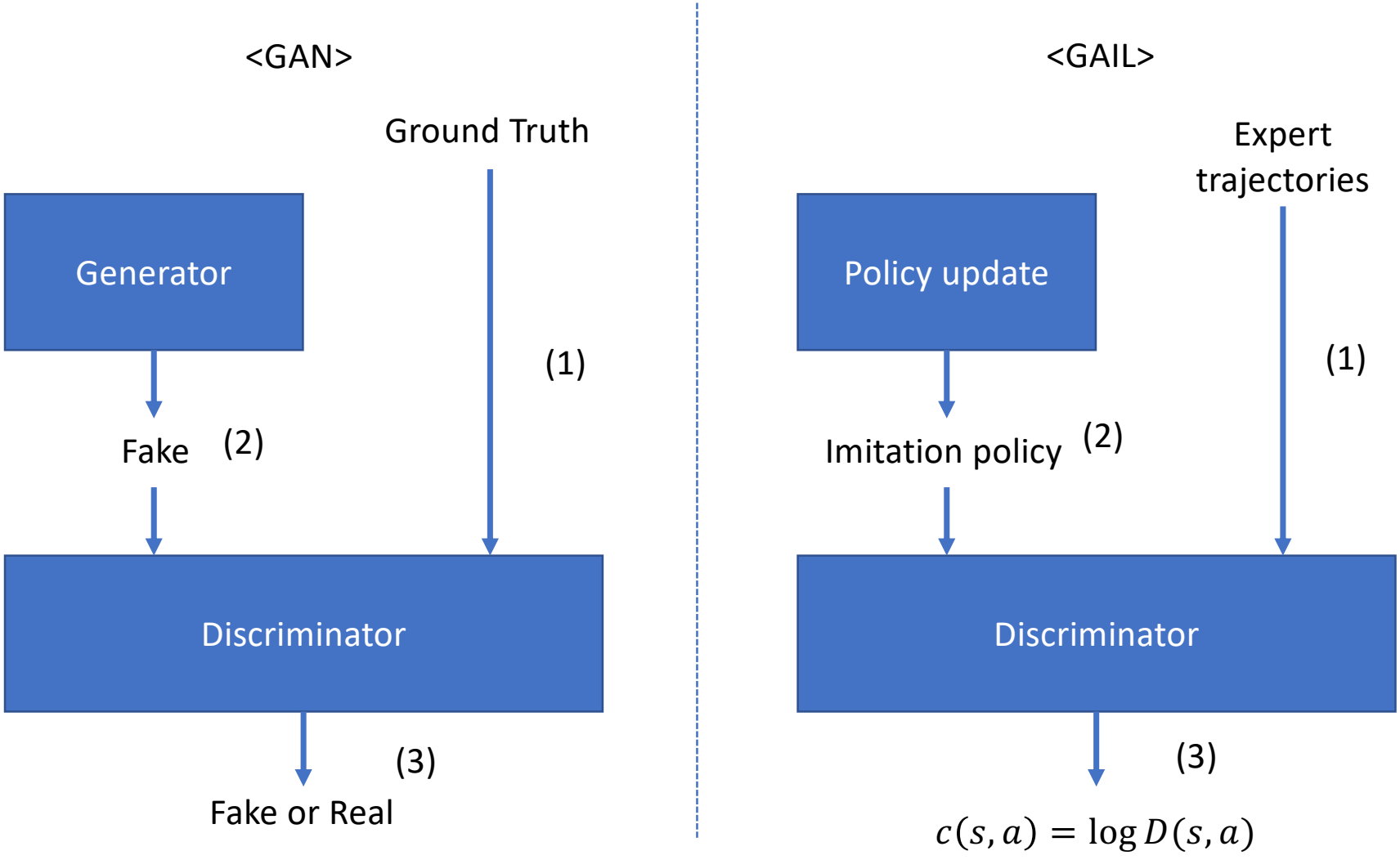
$$\psi^*_{\mathrm{GA}}(\rho_\pi - \rho_{\pi_E}) = \sup_{D \in (0,1)^{\mathcal{S}\times\mathcal{A}}} \mathbb{E}_\pi[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))]$$

- We now present a practival imitation learning algorithm

$$\min_\pi \max_{D \in (0,1)^{\mathcal{S}\times\mathcal{A}}} \mathbb{E}_\pi[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))] - \lambda H(\pi)$$

We find a saddle point (π, D)

GAN and GAIL


<GAN>
Ground Truth
Generator
(1)
Fake
(2)
Discriminator
(3)
Fake or Real
<GAIL>
Expert trajectories
Policy update
(1)
Imitation policy
(2)
Discriminator
(3)
$c(s,a) = \log D(s,a)$

03 Proposed Method

Algorithm 1 Generative adversarial imitation learning

1: **Input:** Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameters θ_0, w_0
2: **for** $i = 0, 1, 2, \ldots$ **do**
3: Sample trajectories $\tau_i \sim \pi_{\theta_i}$
4: Update the discriminator parameters from w_i to w_{i+1} with the gradient

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w \log(1 - D_w(s,a))] \tag{17}$$

5: Take a policy step from θ_i to θ_{i+1}, using the TRPO rule with cost function $\log(D_{w_{i+1}}(s,a))$. Specifically, take a KL-constrained natural gradient step with

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_\theta \log \pi_\theta(a|s) Q(s,a)] - \lambda \nabla_\theta H(\pi_\theta), \quad \text{where } Q(\bar{s},\bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{w_{i+1}}(s,a)) \mid s_0 = \bar{s}, a_0 = \bar{a}] \tag{18}$$

6: **end for**

- Initalize the policy π_θ and a discrimator $D_w: S \times A \rightarrow (0,1)$
- Alternatively update θ and w
  - Adam for gradient step on w increase; Discriminator update
  - TRPO step on θ to decrease; policy update
- Discriminator network is a local cost function providing learning signal to th e policy.
- Taking a policy step that decreases expected cost w.r.t $c(s,a) = \log D(s,a)$

Conclusions

04 Conclusions

- GAIL is generally efficient in terms of expert data, but it is not particularly sample efficient in terms of environment interacti ons during training as it is model free
- It uses TRPO to train the policies form reinforcement learning.

Q & A

THANK YOU.