InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations

Chih-Hui Ho, Chun Hu, Po-Jung Lai

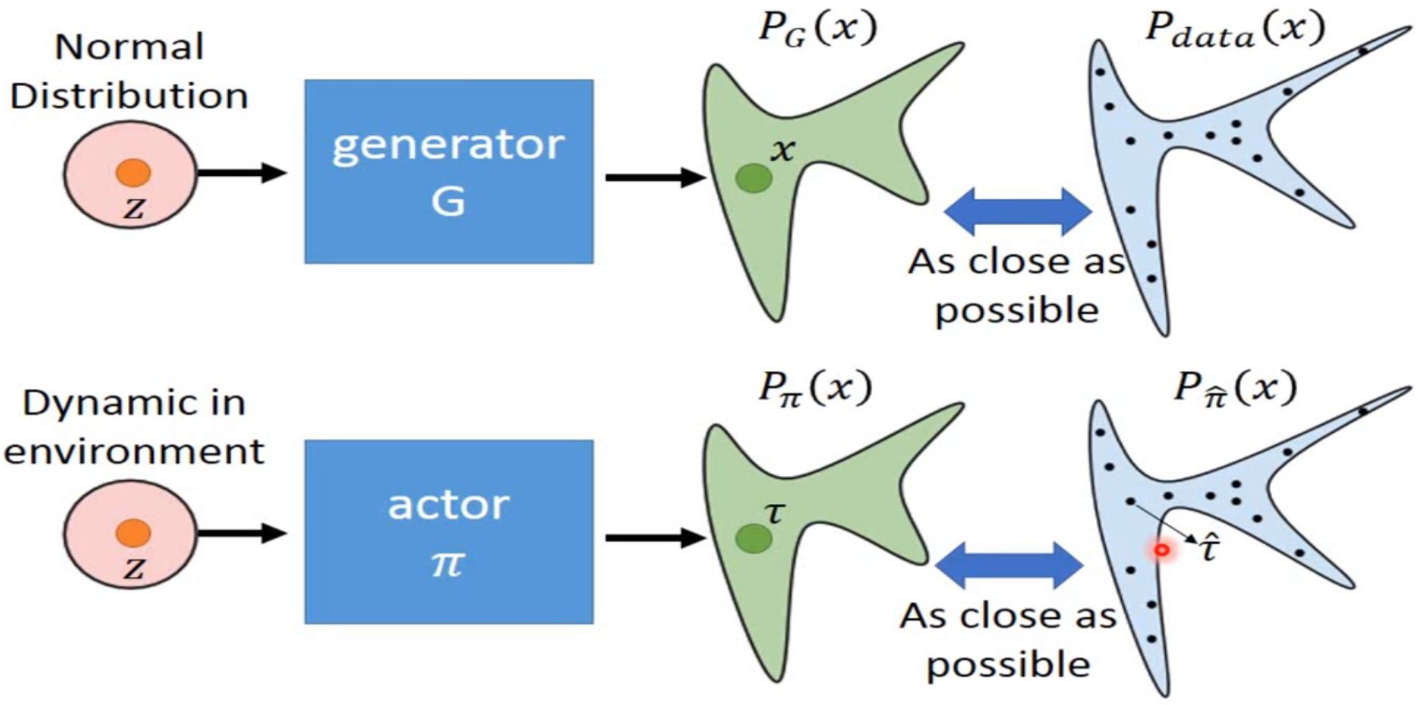
Outline

1. Introduction
2. Related work
   - Generative adversarial imitation learning (GAIL)
3. Proposed method
4. Experiment results
5. Conclusion

Introduction

- A reward function is important in RL task
- Hard to design reward function in some scenario (e.g. autonomous driving)
- Imitation learning allows agents to learn how to perform task like an expert
  - Generative Adversarial Imitation Learning (GAIL, [12])
  - Generative adversarial nets (GANs, [13])
- Expert demonstrations varies significantly
  - Multiple experts might have multiple policies
  - Need external latent factors to better represent the observed behavior
- Goal: To develop an imitation learning framework that is able to automatically **discover and disentangle the latent factors of variation underlying expert demonstrations**

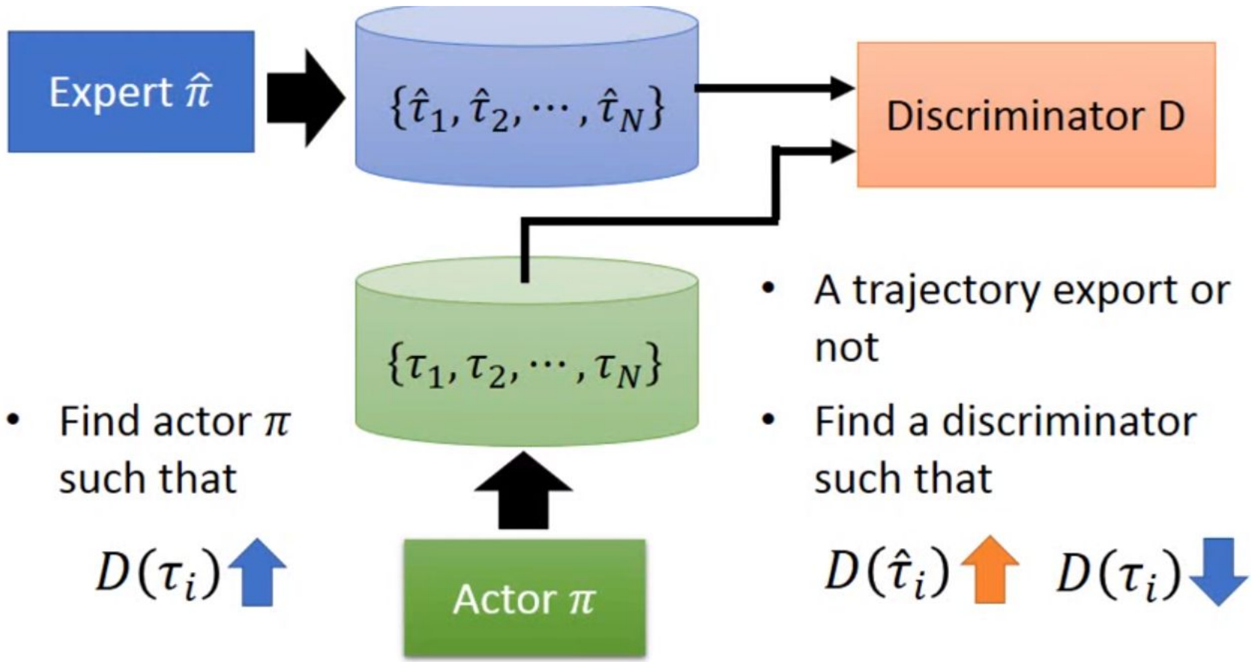
GAN for imitation learning (GAIL)



https://www.youtube.com/watch?v=rOho-2oJFeA

GAN for imitation learning (GAIL)

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



Proposed method

- Introduce a latent factor c to represent the variation under expert demonstrations
- In GAIL, action is chosen as $\pi(a|s)$
- Proposed method chooses action as $\pi(a|s,c)$
- Maximize the mutual information L_I between latent code c and {state, action}.
- L_I is a function of $Q(c|s,a)$

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

InfoGAIL $$\boxed{\min_{\pi,Q}} \max_{D} \mathbb{E}_{\pi}[\log D(s,a)] + \mathbb{E}_{\pi_E}[\log(1-D(s,a))] - \boxed{\lambda_1 L_I(\pi,Q)} - \lambda_2 H(\pi)$$

Proposed method

- Discriminator D_{ω_i} maximizes
- Mutual information Q_{ψ_i} minimizes
- Policy π_θ updates with TRPO[2]

$$\min_{\pi,Q}\max_{D}\mathbb{E}_\pi[\log D(s,a)]+\mathbb{E}_{\pi_E}[\log(1-D(s,a))]-\lambda_1 L_I(\pi,Q)-\lambda_2 H(\pi)$$

Algorithm 1 InfoGAIL

Input: Initial parameters of policy, discriminator and posterior approximation $\theta_0, \omega_0, \psi_0$; expert trajectories $\tau_E \sim \pi_E$ containing state-action pairs.
Output: Learned policy π_θ
for $i = 0, 1, 2, \ldots$ **do**
Sample a batch of latent codes: $c_i \sim p(c)$
Sample trajectories: $\tau_i \sim \pi_{\theta_i}(c_i)$, with the latent code fixed during each rollout.
Sample state-action pairs $\chi_i \sim \tau_i$ and $\chi_E \sim \tau_E$ with same batch size.
Update ω_i to ω_{i+1} by ascending with gradients

$$\Delta_{\omega_i} = \hat{\mathbb{E}}_{\chi_i}[\nabla_{\omega_i}\log D_{\omega_i}(s,a)] + \hat{\mathbb{E}}_{\chi_E}[\nabla_{\omega_i}\log(1-D_{\omega_i}(s,a))]$$

Update ψ_i to ψ_{i+1} by descending with gradients

$$\Delta_{\psi_i} = -\lambda_1\hat{\mathbb{E}}_{\chi_i}[\nabla_{\psi_i}\log Q_{\psi_i}(c|s,a)]$$

Take a policy step from θ_i to θ_{i+1}, using the TRPO update rule with the following objective:

$$\hat{\mathbb{E}}_{\chi_i}[\log D_{\omega_{i+1}}(s,a)] - \lambda_1 L_I(\pi_{\theta_i}, Q_{\psi_{i+1}}) - \lambda_2 H(\pi_{\theta_i})$$

end for

Proposed method

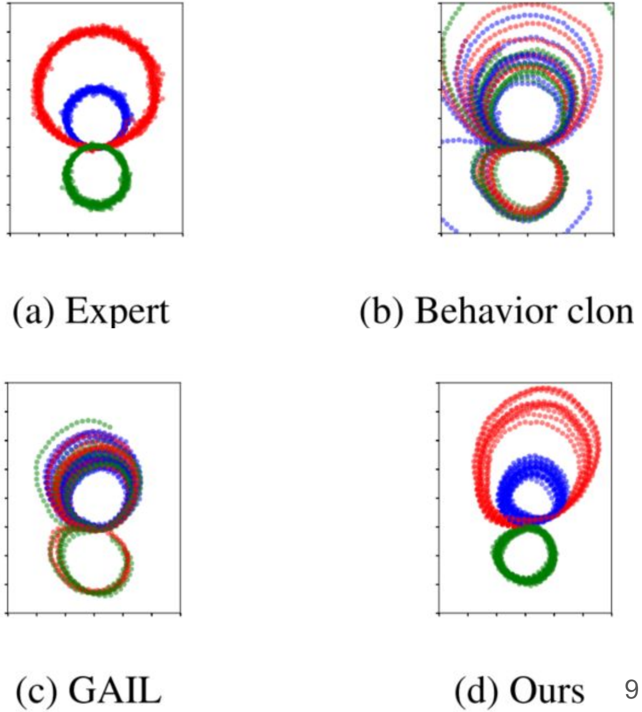
- Reward augmentation
  - Helps when expert perform sub-optimally
  - Hybrid between RL and imitation learning

$$\min_{\theta,\psi} \max_{\omega} \mathbb{E}_{\pi_\theta}[\log D_\omega(s,a)] + \mathbb{E}_{\pi_E}[\log(1 - D_\omega(s,a))] \boxed{- \lambda_0 \eta(\pi_\theta)} - \lambda_1 L_I(\pi_\theta, Q_\psi) - \lambda_2 H(\pi_\theta)$$

- Replace vanilla GAN with WGAN[26]
  - More stable and easier to train
  - $$\min_{\theta,\psi} \max_{\omega} \mathbb{E}_{\pi_\theta} \boxed{[D_\omega(s,a)] - \mathbb{E}_{\pi_E}[D_\omega(s,a)]} - \lambda_0 \eta(\pi_\theta) - \lambda_1 L_I(\pi_\theta, Q_\psi) - \lambda_2 H(\pi_\theta)$$

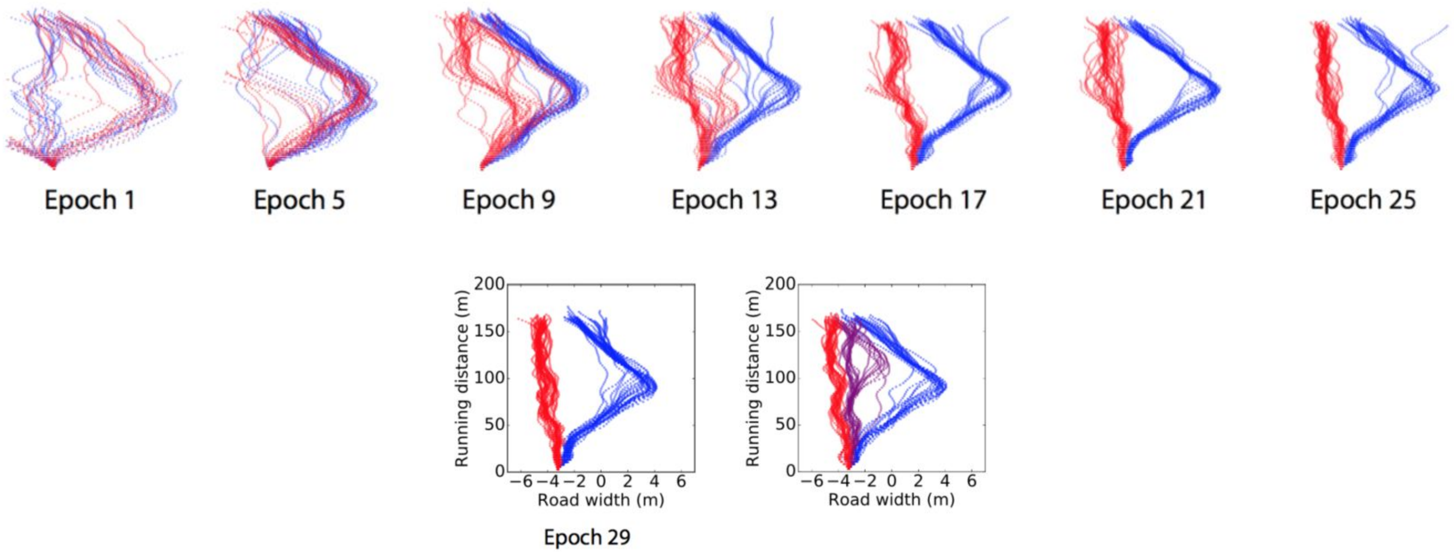
Experiment Result - Learning to Distinguish Trajectories

- The driving experiment are conducted on Open Source Race Car Simulator
- Each color denotes one specific latent code
    - Different experts have different trajectories



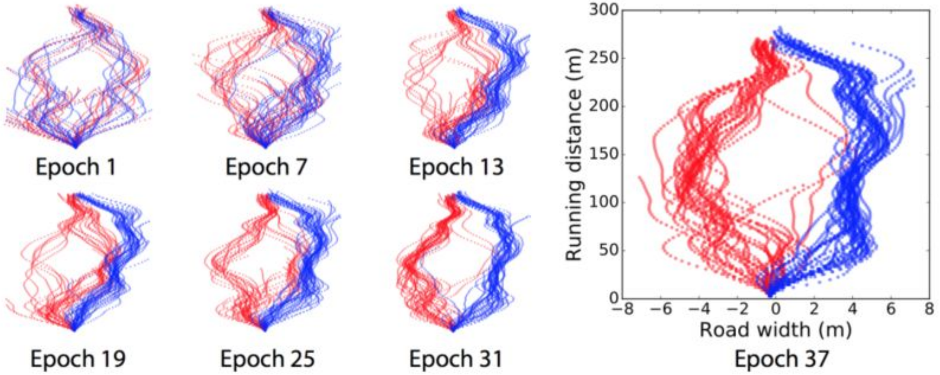
Experiment Result - Interpretable Imitation Learning

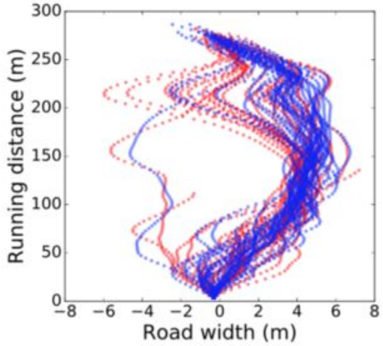
- Blue and red indicate policies under different latent codes
- They correspond to “turning from inner lane” and “turning from outer lane” respectively



Experiment Result - Interpretable Imitation Learning

- Different latent codes correspond to passing from right or left





InfoGAIL **GAIL**

Experiment

Method	Avg. rollout distance
Behavior Cloning	701.83
GAIL	914.45
InfoGAIL \ RB	1031.13
InfoGAIL \ RA	1123.89
InfoGAIL \ WGAN	1177.72
InfoGAIL (Ours)	**1226.68**
Human	1203.51

Conclusion

- Automatically distinguish certain driving behaviors by introducing the latent factors
- Discovering the latent factors without direct supervision
- Perform imitation learning by using only visual inputs
- Learning a policy that can imitate and even outperform the human experts

Demo Video