

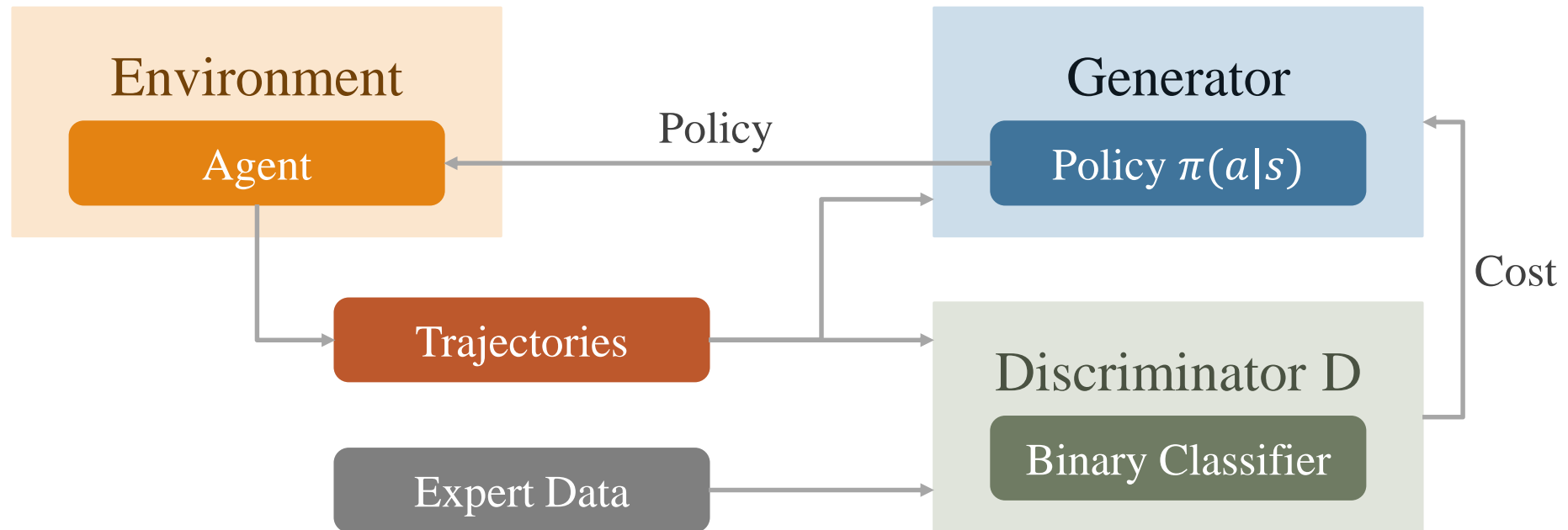
InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations

PRESENTER: YIN-HUNG CHEN

Motivation

Recap: GAIL

- A generator producing a policy π competes with a discriminator distinguishing π and the expert.

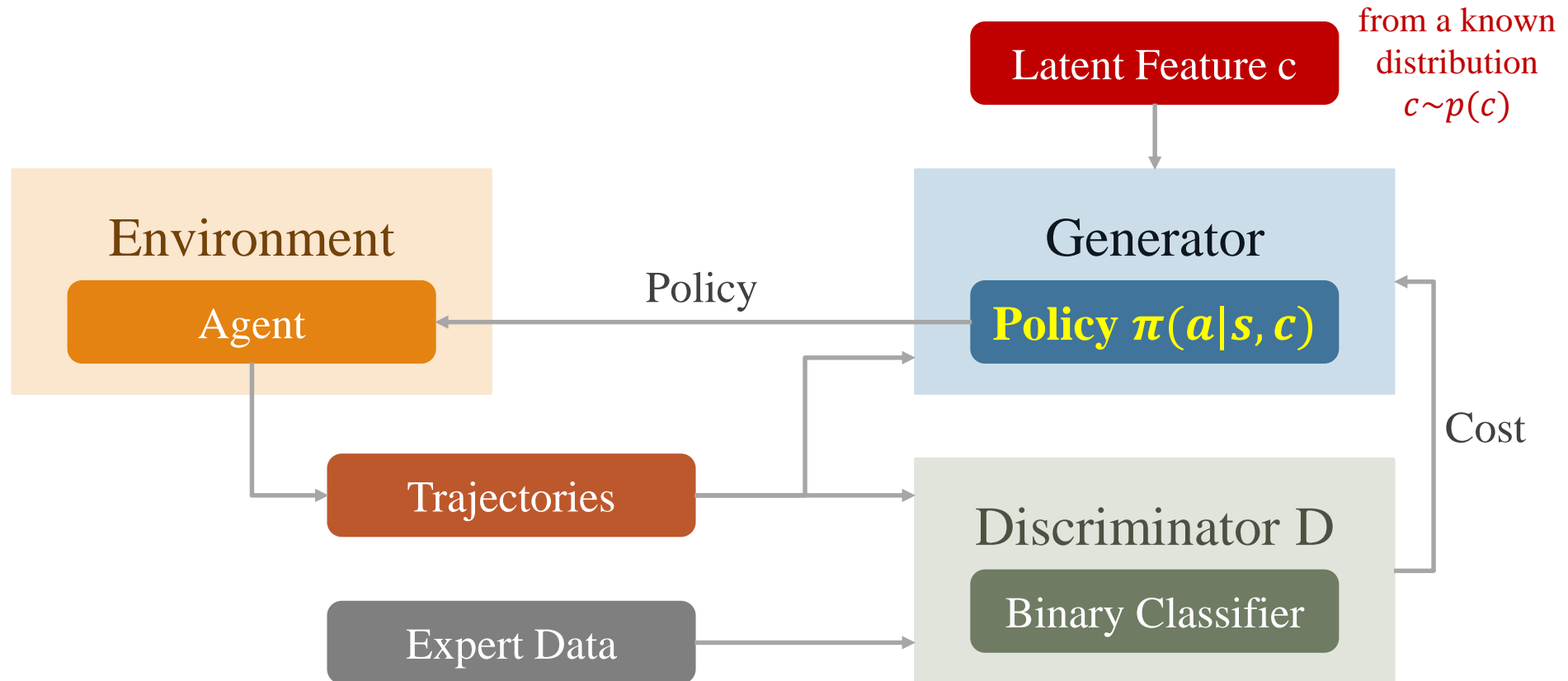


Drawbacks of GAIL

- In reality, expert demonstrations can show significant **variability**.
- The observations might have been sampled from **different experts with different skills and habits**.
- **External latent factors** of variation are not explicitly captured by GAIL, but they can significantly affect the observed behaviors.

InfoGAIL

Modified GAIL



Objective Function

- GAIL

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

where π is learner policy, and π_E is expert policy.

- InfoGAIL

- Discriminator: same with GAIL
- Generator: simply introducing latent factor c into $\pi \rightarrow \pi(a|s, c)$

However, applying GAIL to $\pi(a|s, c)$ could simply ignore c and fail to separate different expert behaviors \rightarrow **adding more constraints over c**

Constraints over Latent Features

- There should be high mutual information between the latent factor c and learner trajectory τ .

$$I(c; \tau) = \sum_{\tau} p(\tau) \sum_c p(c|\tau) \log_2 \frac{p(c|\tau)}{p(c)}$$

- Independent

$$p(c|\tau) = \frac{p(c) \times p(\tau)}{p(\tau)}, \frac{p(c|\tau)}{p(c)} = 1, \log_2 \frac{p(c|\tau)}{p(c)} = 0$$

- Maximizing mutual information $I(c; \tau)$
 - hard to maximize directly as it requires the posterior $P(c|\tau)$
 - using $Q(c|\tau)$ to estimate $P(c|\tau)$

Constraints over Latent Features

- Introducing the lower bound $L_I(\pi, Q)$ of $I(c; \tau)$

$$I(c; \tau)$$

$$= H(c) - H(c|\tau)$$

$$= \mathbb{E}_{a \sim \pi(\cdot|s,c)} \left[\mathbb{E}_{c' \sim P(c|\tau)} [\log P(c'|\tau)] \right] + H(c)$$

$$= \mathbb{E}_{a \sim \pi(\cdot|s,c)} \left[D_{KL}(P(\cdot|\tau) \parallel Q(\cdot|\tau)) + \mathbb{E}_{c' \sim P(c|\tau)} [\log Q(c'|\tau)] \right] + H(c)$$

$$\geq \mathbb{E}_{a \sim \pi(\cdot|s,c)} \left[\mathbb{E}_{c' \sim P(c|\tau)} [\log Q(c'|\tau)] \right] + H(c)$$

$$= \mathbb{E}_{c \sim P(c), a \sim \pi(\cdot|s,c)} [\log Q(c|\tau)] + H(c)$$

$$= L_I(\pi, Q)$$

Constraints over Latent Features

- There should be high mutual information between the latent factor c and learner trajectory τ .
- Maximizing mutual information $I(c; \tau)$
 - hard to maximize directly as it requires the posterior $P(c|\tau)$
 - using $Q(c|\tau)$ to estimate $P(c|\tau)$
- Maximizing $I(c; \tau)$ through maximize the lower bound $L_I(\pi, Q)$

Objective Function

- GAIL

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

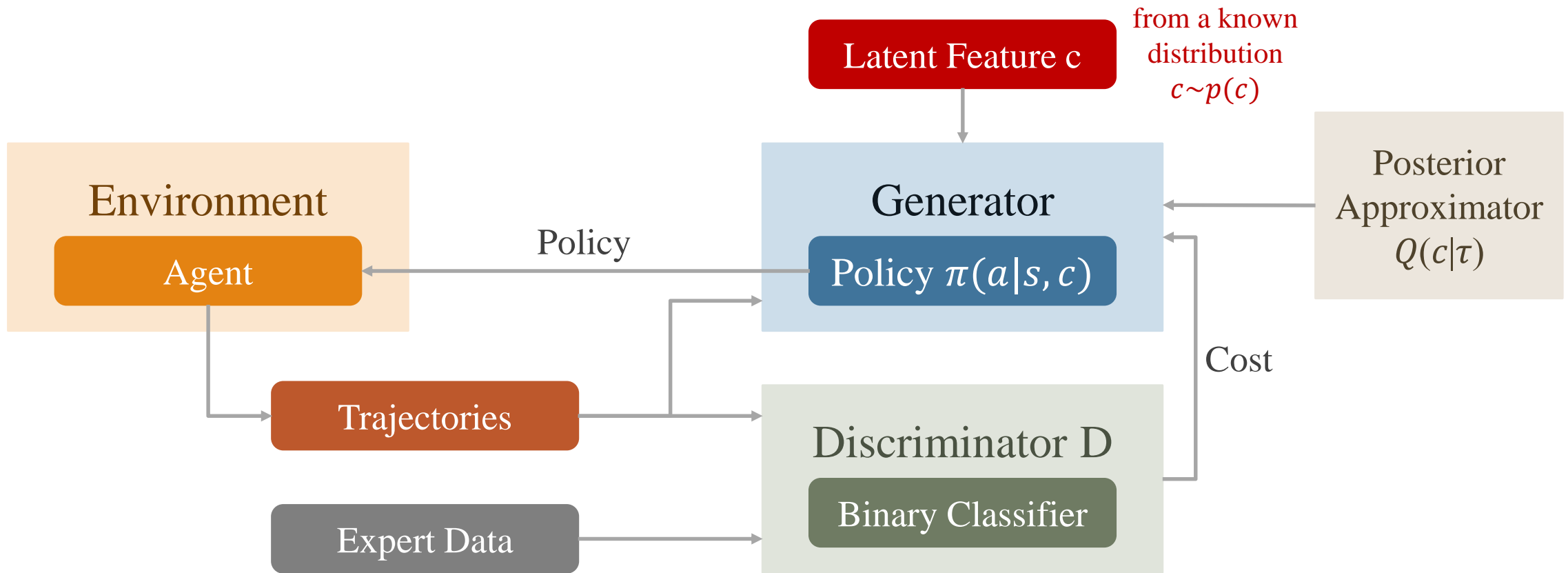
where π is learner policy, and π_E is expert policy.

- InfoGAIL

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

where $\lambda_1 > 0$ and $\lambda_2 > 0$.

InfoGAIL



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 $i = 0, 1, 2, \dots$ **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

Additional Optimization

Reward Augmentation

- If the expert is performing sub-optimally, then any policy trained under the recovered rewards will be also suboptimal.
- Reward augmentation: providing **additional incentives to incorporate prior knowledge** to the agent without interfering with the imitation learning process.
 - specifying a **surrogate state-based reward** $\eta(\pi_\theta) = \mathbb{E}_{s \sim \pi_\theta}[\mathbf{r}(s)]$ that reflects our bias over the desired agent's behavior.

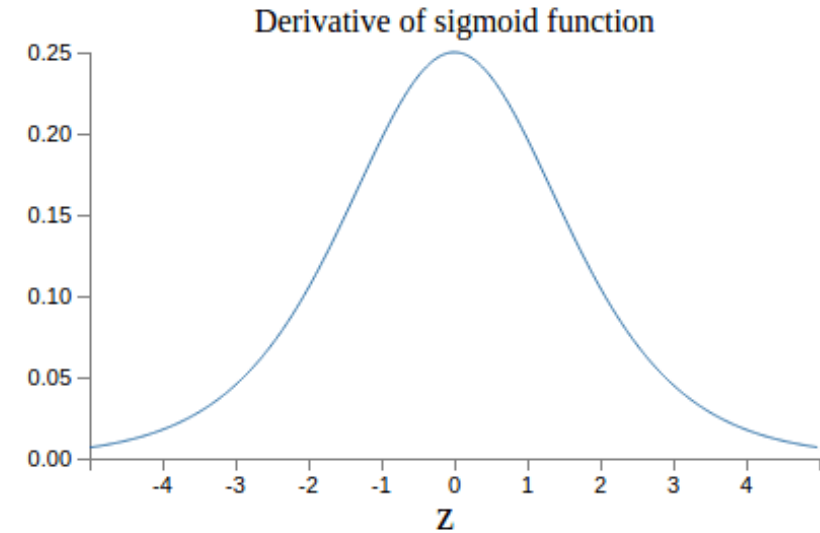
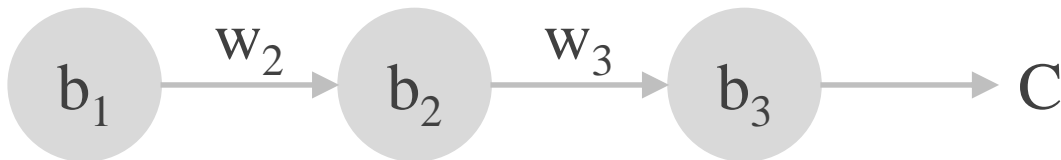
$$\min_{\theta, \psi} \max_{\omega} \mathbb{E}_{\pi_\theta} [\log D_\omega(s, a)] + \mathbb{E}_{\pi_E} [\log(1 - D_\omega(s, a))] - \lambda_0 \eta(\pi_\theta) - \lambda_1 L_I(\pi_\theta, Q_\psi) - \lambda_2 H(\pi_\theta)$$

where $\lambda_0 > 0$.

Improved Optimization

- The traditional GAN objective suffers from vanishing gradient and mode collapse problems.
- Vanishing gradient

$$\begin{aligned}\frac{\partial C}{\partial b_1} &= \frac{\partial C}{\partial y_3} \frac{\partial y_3}{\partial z_3} \frac{\partial z_3}{\partial x_3} \frac{\partial x_3}{\partial z_2} \frac{\partial z_2}{\partial x_2} \frac{\partial x_2}{\partial z_1} \frac{\partial z_1}{\partial b_1} \\ &= \frac{\partial C}{\partial y_3} \sigma'(z_3) w_3 \sigma'(z_2) w_2 \sigma'(z_1)\end{aligned}$$



Improved Optimization

- The traditional GAN objective suffers from vanishing gradient and mode collapse problems.
- Mode collapse: generator tends to produce the same type of data
→ generator yields the same $G(z)$ for different z

Improved Optimization

- The traditional GAN objective suffers from vanishing gradient and mode collapse problems.
→ using the Wasserstein GAN (WGAN)

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

Algorithm 2 InfoGAIL with extensions

Input: Expert trajectories $\tau_E \sim \pi_E$; initial policy, discriminator and posterior parameters $\theta_0, \omega_0, \psi_0$; replay buffer $B = \emptyset$;

Output: Learned policy π_θ

for $i = 0, 1, 2, \dots$ **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.

 Update the replay buffer: $B \leftarrow B \cup \tau_i$.

 Sample $\chi_i \sim B$ and $\chi_E \sim \tau_E$ with same batch size.

 Update ω_i to ω_{i+1} by ascending with gradients

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

 Clip the weights of ω_{i+1} to $[-0.01, 0.01]$.

 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 (without reward augmentation):

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

 or (with reward augmentation):

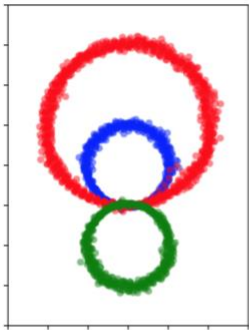
$$\hat{\mathbb{E}}_{\chi_i}[D_{\omega_{i+1}}(s, a)] - \lambda_0 \eta(\pi_{\theta_i}) - \lambda_1 L_I(\pi_{\theta_i}, Q_{\psi_{i+1}}) - \lambda_2 H(\pi_{\theta_i})$$

end for

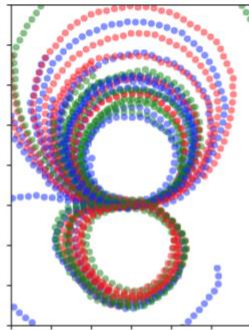
Experiments

Learning to Distinguish Trajectories

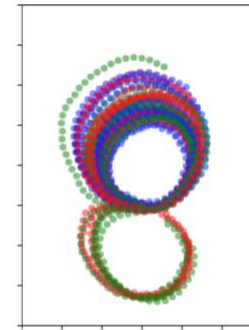
- The observations at time t are positions from $t - 4$ to t .
- The latent code is a one-hot encoded vector with 3 dimensions and a uniform prior.



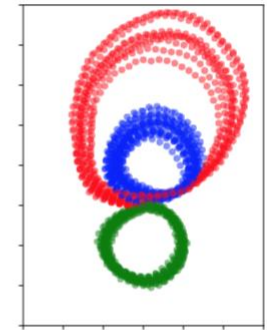
(a) Expert



(b) Behavior cloning



(c) GAIL

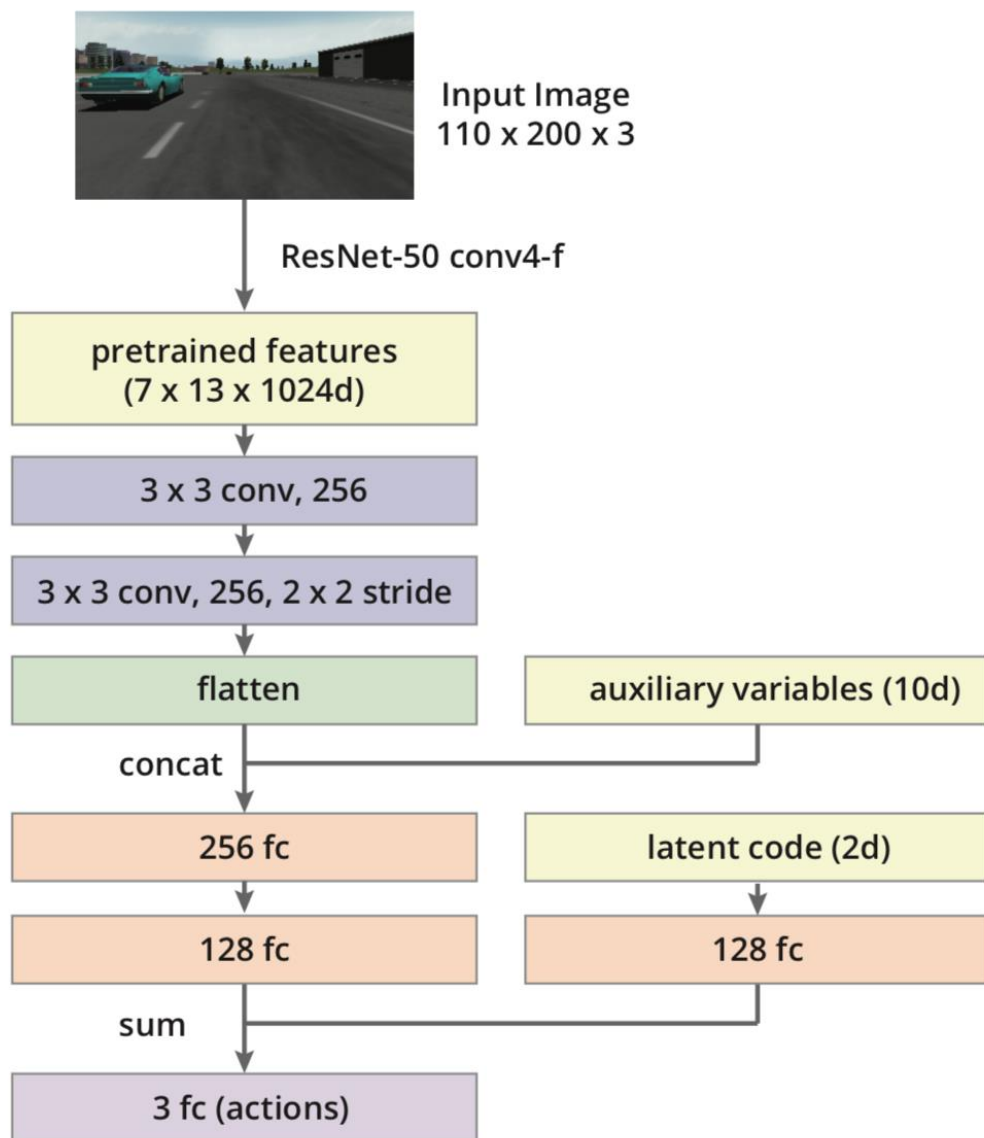


(d) Ours

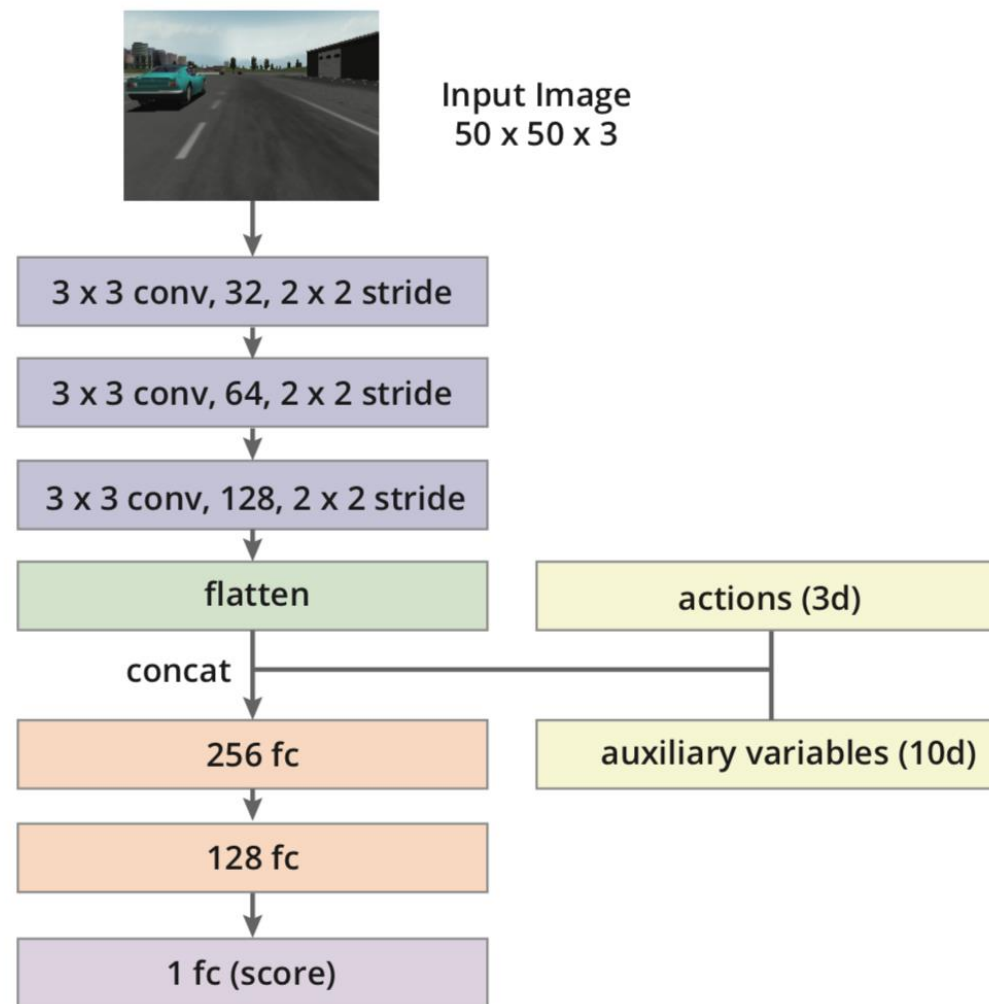
Self-driving car in the TORCS Environment

- The demonstrations collected by manually driving
- Three-dimensional continuous action composed of *steering*, *acceleration*, and *braking*
- Raw visual inputs as the only external inputs for the state
- Auxiliary information as internal input, including velocity at time t , actions at time $t - 1$ and $t - 2$, and damage of the car
- Pre-trained ResNet on ImageNet





(a) Network architecture for the policy/generator π_{θ} .

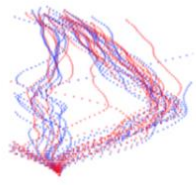


(b) Network architecture for the discriminator D_{ω} .

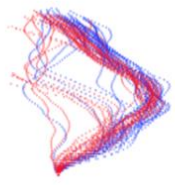
Performance

● Turn

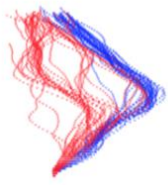
$[0, 1]$ corresponds to using the inside lane (blue lines), while $[1, 0]$ corresponds to the outside lane (red lines).



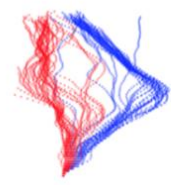
Epoch 1



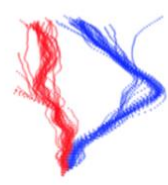
Epoch 5



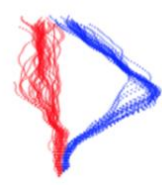
Epoch 9



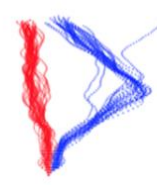
Epoch 13



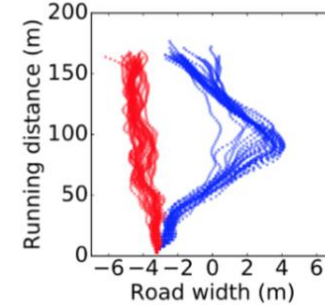
Epoch 17



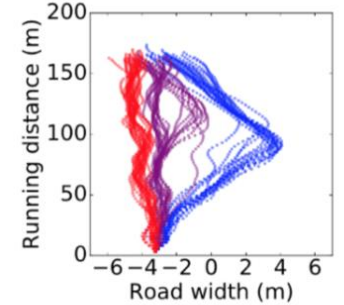
Epoch 21



Epoch 25



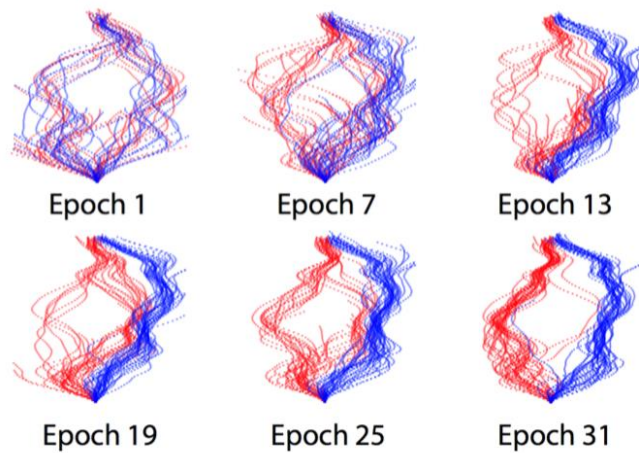
Epoch 29



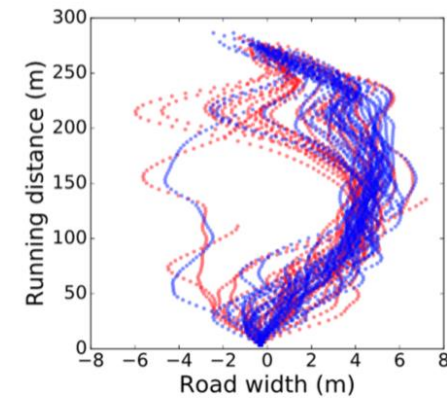
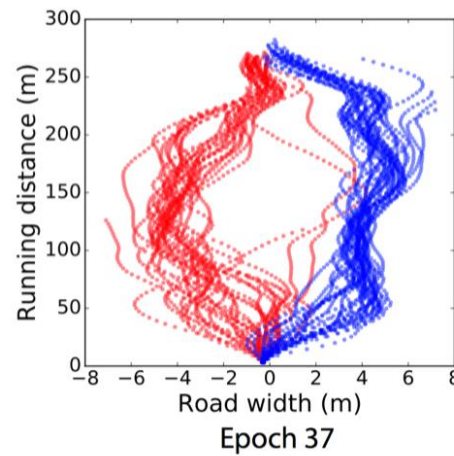
Performance

● Pass

$[0, 1]$ corresponds to passing from right (red lines), while $[1, 0]$ corresponds to passing from left (blue lines).



InfoGAIL



GAIL

Performance

- Classification accuracies of $Q(c|\tau)$
- Reward augmentation encouraging the car to drive faster

Table 1: Classification accuracies for *pass*.

| Method | Accuracy |
|------------------------|--------------|
| Chance | 50% |
| K-means | 55.4% |
| PCA | 61.7% |
| InfoGAIL (Ours) | 81.9% |
| SVM | 85.8% |
| CNN | 90.8% |

Table 2: Average rollout distances.

| 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 |

Q&A