LAMDA
Learning And Mining from DatA
http://lamda.nju.edu.cn

南京大學

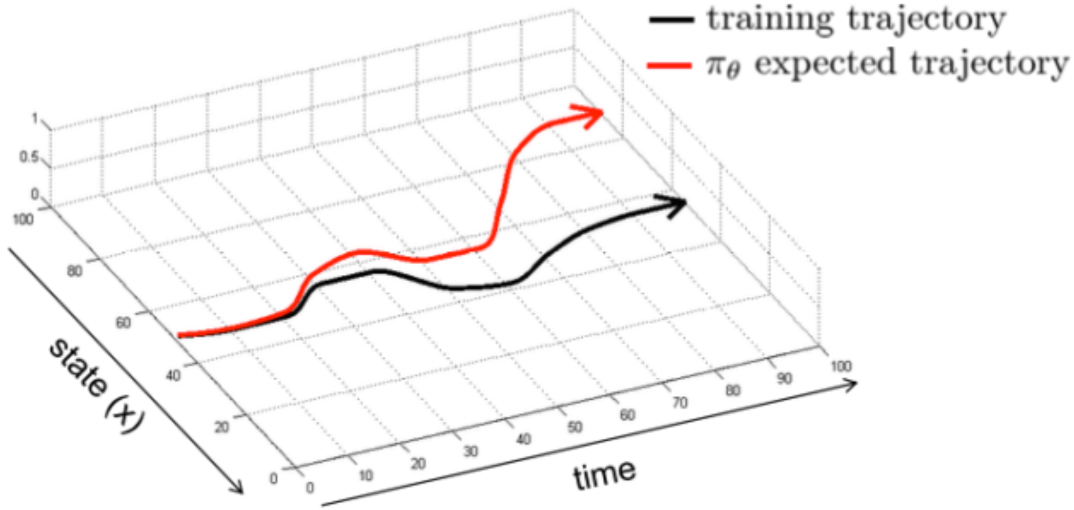
Resent advances in Multi-Modal GAN-ILs

Presented by Quan He

Imitation Learning : Overview

Learning And Mining from DatA
http://lamda.nju.edu.cn

- Behavior Cloning : Supervised learning on expert data (state-action pair)
- Advantage: Simple & efficient
- Disadvantage: Cumulative error on long-term trajectory (especially on stochastic transition models)



Imitation Learning : Overview

Learning And Mining from DatA
http://lamda.nju.edu.cn

- Behavior Cloning : Supervised learning on expert data
- Use when:

    1.1-step deviations not too bad

    2.Learning reactive behaviors （short-term behavior）

    3.Expert trajectories can cover state space（small state space）

But if:

    1.Multi-step deviations is catastrophic?

    2.Learning long-term behavior?

    3.State space too large for expert trajectories to cover?

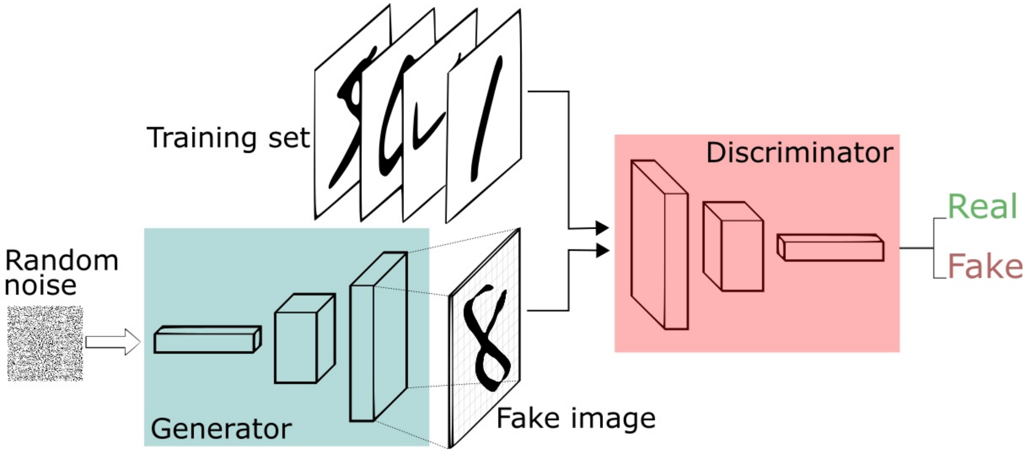
MDP and GAN notations

Learning And Mining from DatA
http://lamda.nju.edu.cn

- <S, A, P, R, γ, π>
- S is the set of states of the environment
- A is the set of actions
- P describes the dynamics of the system in the form of transition probabilities $\boldsymbol{P(s'|s,a)}$
- R is the immediate reward function $\boldsymbol{R(s, a)}$ that describes the reward of selecting $\boldsymbol{a}$ in $\boldsymbol{s}$
- $\gamma \in [0,1)$ is the discount factor
- $\pi_\theta(s|a)$ is the possibility that choose action $\boldsymbol{a}$ in $\boldsymbol{s}$ under policy π with parameter vector θ
- G(z; θ_g) is a sample generated from a random noise z and a generator parameter vector θ_g
- D(x; θ_d) is a probability that sample x come from data rather than a generator, which is judged by a generator parameter vector θ_d
- λ_x, ω : hyper parameter to control the influence of different part x

GAN

- GAN-IL is base on GANs and GAIL
- GANs: Using a generative model G and a discriminative model D, try to minimize the “distance” between the true sample distribution and the generated sample distribution



Generative Adversarial Nets [Goodfellow et al., NIPS 2014]

GAN

Learning And Mining from DatA
http://lamda.nju.edu.cn

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**
  for k steps **do**
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\boxed{\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].}$$

Update discriminator parameter θ_d

  end for
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\boxed{\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).}$$

Update generator parameter θ_g

end for

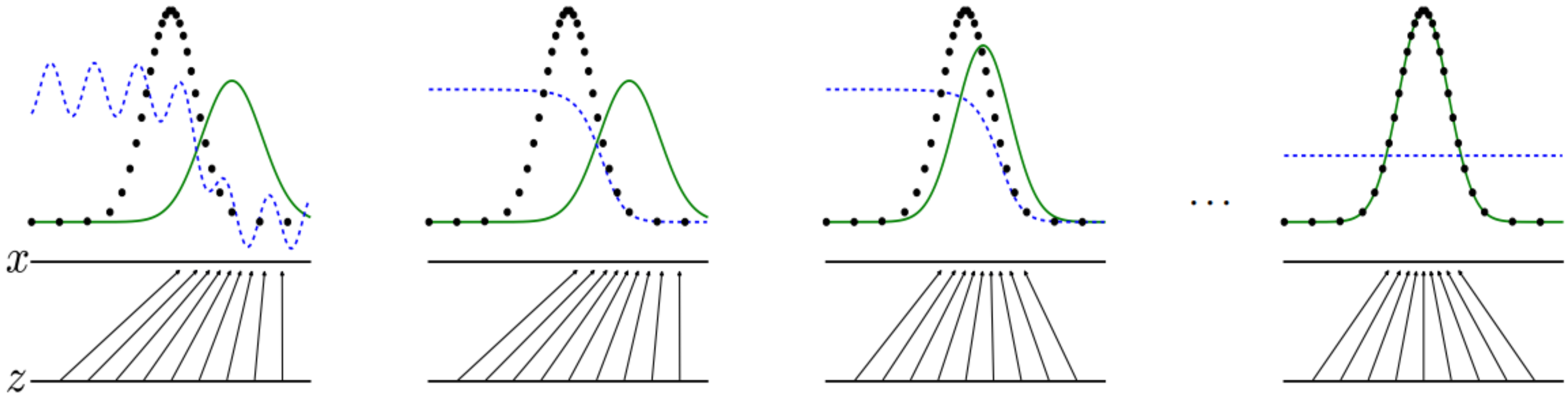
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

GAN

LAMDA Learning And Mining from DatA http://lamda.nju.edu.cn

GAN goal: minimize the Jensen-Shannon divergence between generative distribution and data distribution

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$



(a) (b) (c) (d)

GAIL

Learning And Mining from DatA
http://lamda.nju.edu.cn

GAIL goal: minimize the Jensen-Shannon divergence between generative policy and expert policy and the entropy of the policy

$$\min_{\theta}\max_{\omega}\mathbb{E}_{\pi_\theta}[\log(D_\omega(s,a))]+\mathbb{E}_{\pi_E}[\log(1-D_\omega(s,a))]-\lambda H(\pi_\theta)$$

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

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

Update discriminator parameter w

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

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

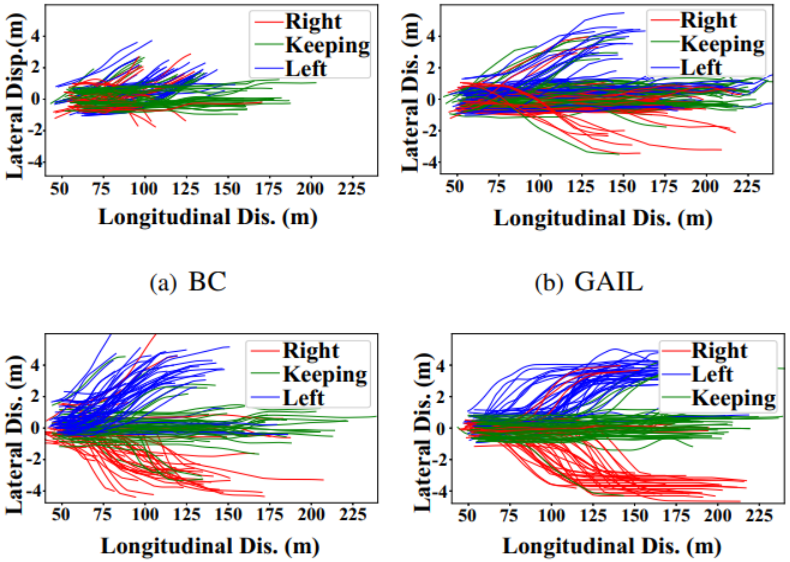
Update policy parameter θ with Q-value from the discriminator with parameter w

6: **end for**

Generative adversarial imitation learning (GAIL) [Ho & Ermon, NIPS 2016]

Conditional GAN/Conditional GAIL

- The input data could have different “model”(turn right, left or go straight in driving)
- GAN does not consider the model of the input data, and neither does GAIL, which would cause model collapse (模态坍缩)



(a) BC (b) GAIL

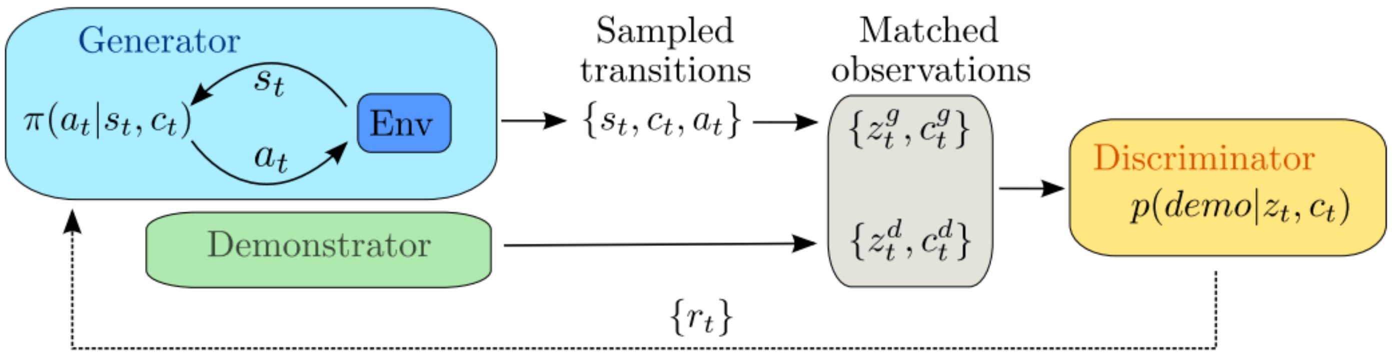
(c) CGAIL (d) Triple-GAIL

Conditional GAN/Conditional GAIL

- The simplest way making use of the condition information : add constraints on the model of sample/trajectories (suppose we know the condition of the sample/trajectories)

CGAIL goal: minimize the Jensen-Shannon divergence between generated policy and expert policy under same condition (GAIL per condition)

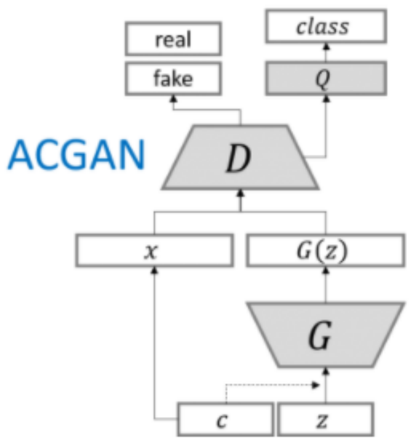
$$\min_{\pi} \max_{D} L_{\mathrm{CGAIL}}(\pi, D, c) = \mathbb{E}_{\pi}[\log D(s, a \mid c)] + \mathbb{E}_{\pi_E}[\log(1 - D(s, a \mid c))]$$

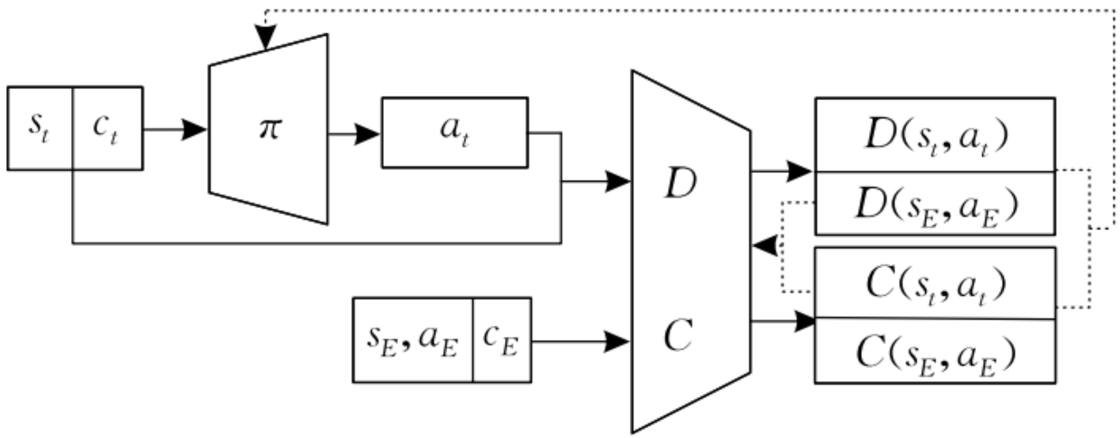


Learning human behaviors from motion capture by adversarial imitation (CGAIL) [Merel et al., Arxiv 2017]

ACGAN/ACGAIL

- ACGAIL: add a Auxiliary Classifier C to determine the condition of the samples/ trajectories
- $C(c|s, a)$ is the probability that state-action pair come from label c in a policy judged by Classifier C
- Classifier C and determinister D could share same input and hidden layer parameter





ACGAIL: Imitation Learning About Multiple Intentions with Auxiliary Classifier GANs [Lin & Zhang, PRICAI 2018]

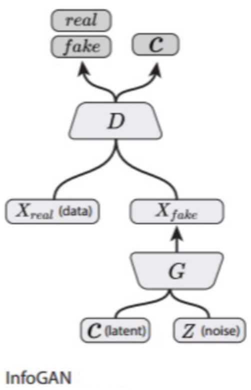
ACGAN/ACGAIL

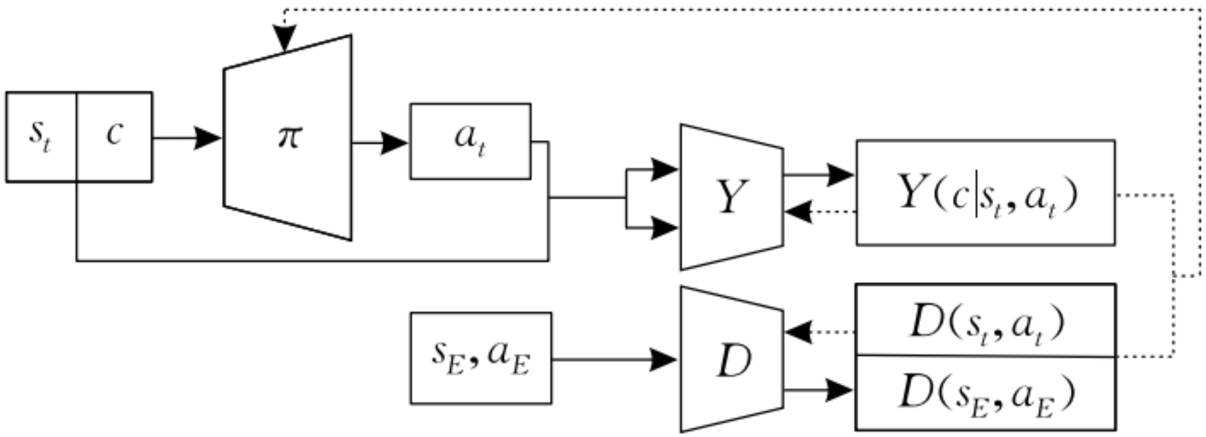
- ACGAIL goal: minimize the Jensen-Shannon divergence between generated policy and expert policy and the Cross entropy of true label and the label judged by Classifier C
- $\min_{\pi,c} \max_{D} L_{ACGAIL}(\pi, D, C) = \mathbb{E}_{\pi}[log D(s,a)] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))] +$
$$\lambda_c\{\mathbb{E}_{\pi}\big[H\big(c, C(c|s,a)\big)\big] + \mathbb{E}_{\pi_E}\big[H\big(c, C(c|s,a)\big)\big]\}$$
- $H\big(c, C(c|s,a)\big)$: the cross entropy of expert label of expert trajectory and the label judged by Classifier C under a policy
- λ_c: a hyper parameter to control the influence of Classifier C
- Reward function for state-action pair:
$$r(s,a) = -log D(s,a) - \lambda_c\big[H\big(c, C(c|s,a)\big)\big]$$

ACGAIL: Imitation Learning About Multiple Intentions with Auxiliary Classifier GANs [Lin & Zhang, PRICAI 2018]

Info-GAIL

- Let us back to CGAIL, what if we do not know the label of the expert trajectories? (back to GAIL condition)
- Info-GAIL view condition c as a latent variable like Info-GAN, and try to minimize the Mutual information of condition c and state-action pair $s-a$, to maximize the relevance between the condition and the state-action pair





InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations [Li, Song & Ermon, NIPS 2017]

Info-GAIL

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 $B = \varnothing$;
Output: Learned policy π_θ
for $i = 0, 1, 2, ...$ **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):

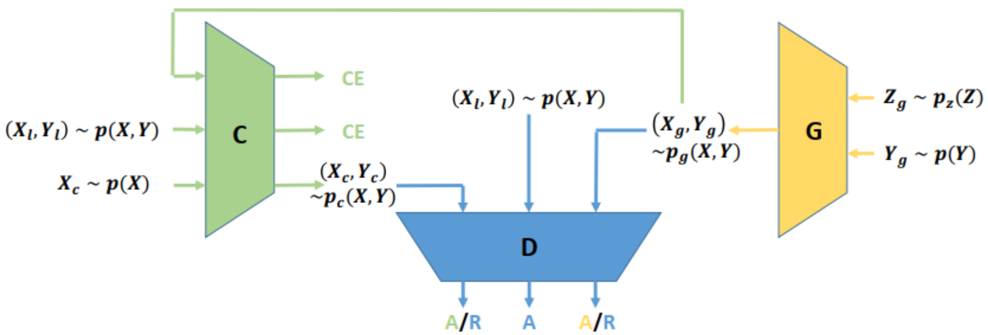
$$\hat{\mathbb{E}}_{\chi_i}[D_{\omega_{i+1}}(s,a)] - \lambda_0 \eta(\pi_{\theta_i}) - \lambda_1 L_I(\pi_{\theta_i}, Q_{\psi_{i+1}}) - \lambda_2 H(\pi_{\theta_i})$$

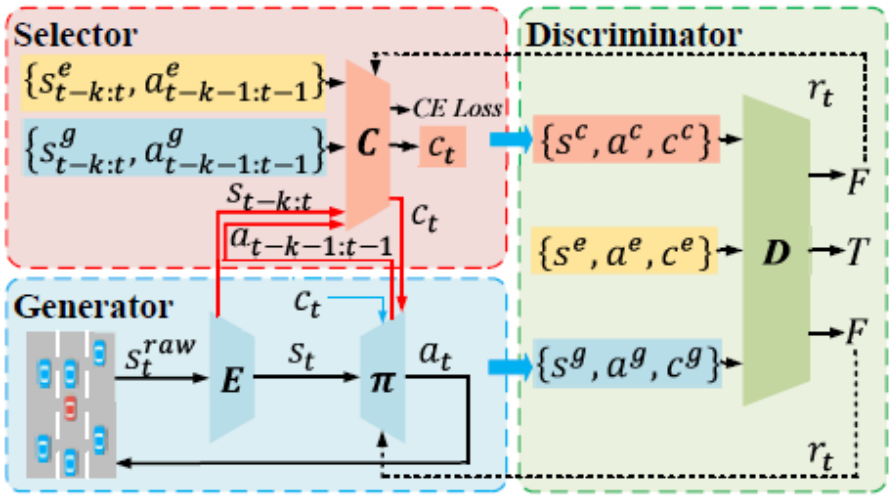
end for

InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations [Li, Song & Ermon, NIPS 2017]

Triple-GAIL

- Triple-GAIL: also add a Classifier C to determine the condition of the trajectories, but Determiner D determines state-action-condition pair, thus Classifier C is dependence of Determiner D (like Triple-GAN)





Triple-GAIL: A multi-modal imitation learning framework with generative adversarial nets
[Fei et al., IJCAI 2020]

Triple-GAIL

- Triple-GAIL goal: minimize the Jensen-Shannon divergence between generated policy and expert policy and the Cross entropy of true label and the label judged by Classifier C

$$\min_{\alpha,\theta}\max_{\psi}\mathbb{E}_{\pi_E}[\log(1-D_\psi(s,a,c))]+\omega\mathbb{E}_{\pi_\theta}[\log D_\psi(s,a,c)]+(1-\omega)\mathbb{E}_{C_\alpha}[\log D_\psi(s,a,c)]+\lambda_E R_E+\lambda_G R_G-\lambda_H H(\pi_\theta)$$

$$\begin{aligned}R_E &= \mathbb{E}_{\pi_E}[-\log p_{C_\alpha}(c|s,a)]\\ &\approx -\frac{1}{N}\sum_{i=0}^{N}\frac{1}{T}\sum_{t=1}^{T}c_{i,t}^{e}\log p_{C_\alpha}\left(c_{i,t}^{c}|s_{i,t}^{e},a_{i,t-1}^{e}\right)\end{aligned}$$

$$\begin{aligned}R_G &= \mathbb{E}_{\pi_\theta}[-\log p_{C_\alpha}(c|s,a)]\\ &\approx -\frac{1}{N}\sum_{i=0}^{N}\frac{1}{T}\sum_{t=1}^{T}c_{i,t}^{g}\log p_{C_\alpha}\left(c_{i,t}^{c}|s_{i,t}^{g},a_{i,t-1}^{g}\right)\end{aligned}$$

Triple-GAIL: A multi-modal imitation learning framework with generative adversarial nets
[Fei et al., IJCAI 2020]

Triple-GAIL

Algorithm 1 The Training Procedure of Triple-GAIL

Input: The multi-intention trajectories of expert τ_E; **Parameter**: The initial parameters θ_0, α_0 and ψ_0

for $i = 0, 1, 2, \cdots$ **do**
  for $j = 0, 1, 2, \cdots, N$ **do**
    Reset environments by the demonstration episodes with fixed label c_j;
    Run policy $\pi_\theta(\cdot|c_j)$ to sample trajectories: $\tau_{c_j} = (s_0, a_0, s_1, a_1, ...s_{T_j}, a_{T_j}|c_j)$
  end for
  Update the parameters of π_θ via TRPO with rewards: $r_{t_j} = -\log D_\psi(s_{t_j}, a_{t_j}, c_j)$
  Update the parameters of D_ψ by gradient ascending with respect to:

$$\nabla_\psi \frac{1}{N_e}\sum_{n=1}^{N_e}\log(1-D_\psi(s_n^e, a_n^e, c_n^e)) + \frac{1}{N}\sum_{j=1}^{N}\left[\frac{\omega}{T_j}\sum_{t=1}^{T_j}\log D_\psi(s_t^g, a_t^g, c_j^g) + \frac{1-\omega}{T_j}\sum_{t=1}^{T_j}\log D_\psi(s_t^c, a_t^c, c_j^c)\right] \tag{9}$$

  Update the parameters of C_α by gradient descending with respect to:

$$\nabla_\alpha \frac{1}{N}\sum_{j=1}^{N}\left[\frac{1-\omega}{T_j}\sum_{t=1}^{T_j}\log D_\psi(s_t^c, a_t^c, c_j^c) - \frac{\lambda_E}{T_j}\sum_{t=1}^{T_j} c_j^e \log p_{C_\alpha}(c_t^c|s_t^e, a_{t-1}^e) - \frac{\lambda_G}{T_j}\sum_{t=1}^{T_j} c_j^e \log p_{C_\alpha}(c_t^c|s_t^g, a_{t-1}^g)\right] \tag{10}$$

end for

Triple-GAIL: A multi-modal imitation learning framework with generative adversarial nets
[Fei et al., IJCAI 2020]

ACGAIL: Supervised learning ACGAN

Info-GAIL：Unsupervised learning Info-GAN

Triple-GAIL: Semi-supervised learning Triple-GAN

What’s the next GAN-IL?（How to find a suitable GAN？）

Thanks