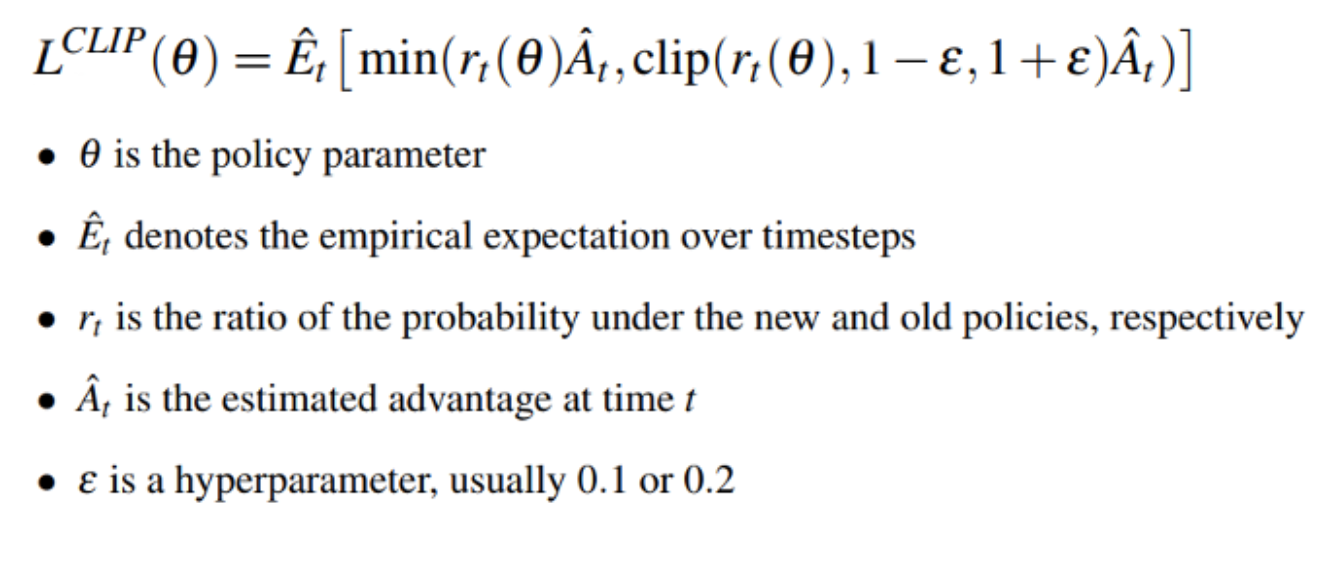
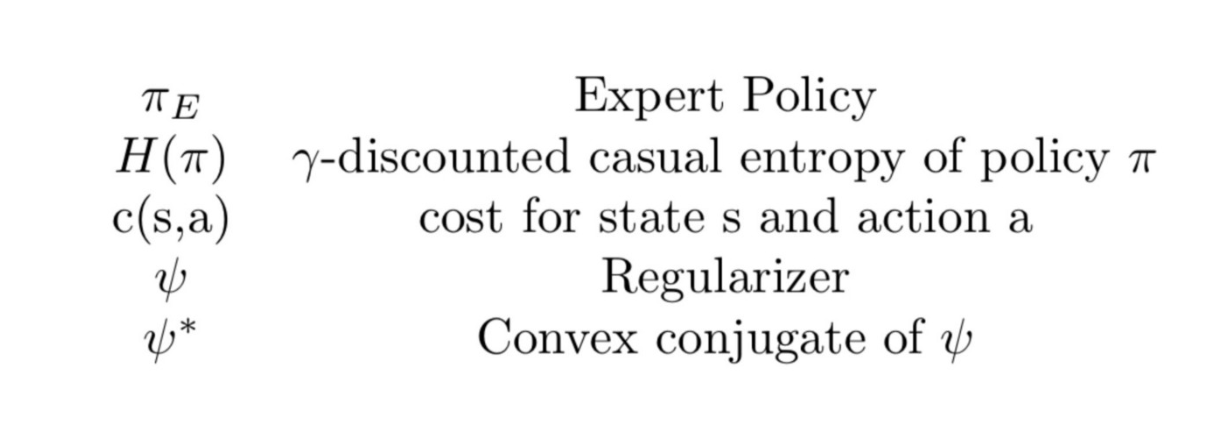
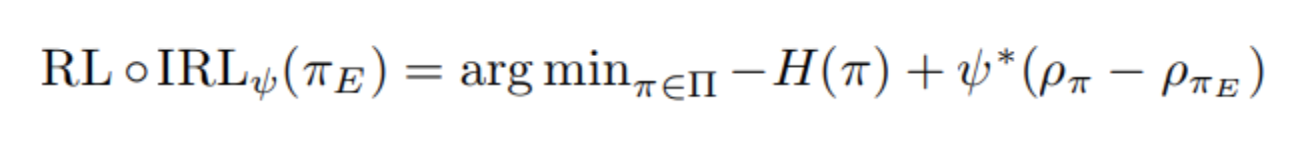
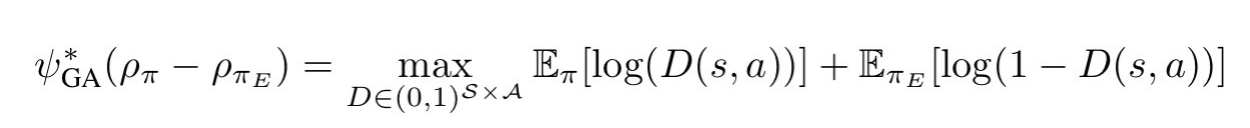
Generative Adversarial Imitation Learning(GAIL)

1. Train experts
   1. python3 run\_ppo.py
   2. ­­전문가 행동을 만들기 위해서 ( 학습 ).
      1. Env-CartPole-V0
      2. 2개의Network 생성 =(Policy, Old Policy)(action을 예측함)
         1. Policy\_net, Value\_net 2개 만들고, action stochastic(Policy\_net사용)하게 함
         2. 2개의 Network ->를 PPO\_network로 넣음(new, old) 참조([https://jay.tech.blog/2018/10/09/trpo와-ppo/](https://jay.tech.blog/2018/10/09/trpo%EC%99%80-ppo/)) <https://reinforcement-learning-kr.github.io/2018/06/22/7_ppo/>
      3. PPO(Proximal Policy Optimal)는 2개의 학습네트워크의 ratio of Probability가 계산에 필요
         1. 기본 트레이닝 세팅
         2. Old, New -> action\_probability 를 가져옴
         3. 
         4. 수행
         5. 각 Network의 Policy -> entropy계산 , Value\_prob -> MSE계산
         6. Training Algorithm
            1. Policy network ( action from action\_prob, value\_prob, reward, state를 에피소드가 끝날 때 까지 모음)A close up of a logo

               Description automatically generated
            2. (Reward + gamma \* V\_next - V) -> advantage term
            3. PPO training
            4. Action probability는 학습은 ->PPO
         7. A screenshot of a social media post

            Description automatically generated
2. Sample Trajectory 만들기 -> GAIL를 수행하기 위해서 data를 생성
   1. python3 sample\_trajectory.py
      1. 기존에 PPO에 학습된 agent를 불러서 cartpole 를 환경에서 action과 state를 가지며 각각을 list로 저장
      2. Trajectory directory에 저장
3. GAIL학습
   1. Python3 run\_gail 실행
      1. 
      2. Policy net (old, new) -> PPO -> Discriminator(env) -> train(PPO)
      3. Discriminator -> Learning agent(not cost)
      4. Expert와 학습되는 Agent의 state\_action을 가져옴 
         1. 여기서 convex conjugate 부분을 Discriminator 학습네트워크로 표현
      5. 각 Agent를 네트워크를 만들고 Probability를 가져옴 ->전문가의 A, S 과 현재 Agent의 A, S를 가져와서 전문가와 유사하게 샘플링함
      6. GAN term 을 Loss function으로 만듬
      7. Reward를 Agent 의 reward로 가짐
         1. Because D(expert|a,s) = 1-D(agent|a,s)
      8. GAIL로 부터 만들어진 Reward를 가지고, PPO의 advantage term(gaes)를 만든이후 PPO학습
4. Behavior\_learning
   1. Python run\_behavior\_clone.py
5. Test policy:
   1. Python test\_policy.py ->gail basic
   2. Python test\_policy.py –alg=bc --model=’number’ : bc
   3. Python test\_policy.py –alg=ppo --model=’number’: ppo
6. 테스트 확인
   1. tensorboard --logdir=log
7. 웹상
   1. localhost:6006 입력
8. test 결과
   1. A screenshot of a cell phone

      Description automatically generated