Reinforcement Learning from Imperfect Demonstrations

Yang Gao* [1] **Huazhe(Harry) Xu*** [1] **Ji Lin** [2] **Fisher Yu** [1] **Sergey Levine** [1] **Trevor Darrell** [1]

ICML 2018

Contents

Introduction

- Current approaches to learning from demonstration perform supervised learning on expert demonstration data.
- It is difficult to jointly optimize and such methods can be sensitve to noisy demonstrations.
- We propose Normalized Actor-Critic (NAC) that effectively normalizes the Q-function and reducing the Q-values of actions unseen in the demonstration data.

Preliminaries - Maximum Entropy Reinforcement Learning (2017 ICML)

Stronger exploration ability

- Standard reinforcement learning setting **Find a optimal path**

$$\pi_{std} = \underset{\pi}{\operatorname{argmax}} \sum_{t} \gamma^t \underset{s_t, a_t \sim \pi}{\mathbb{E}} [R_t]$$

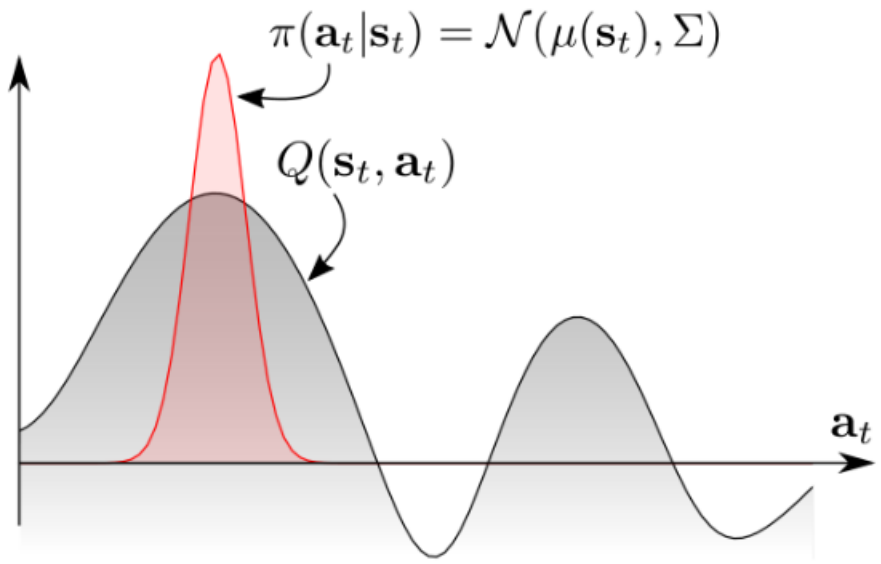
- Maximum entropy police learning **Find all optimal path**

让action尽可能分散，而不是集中在一个action上

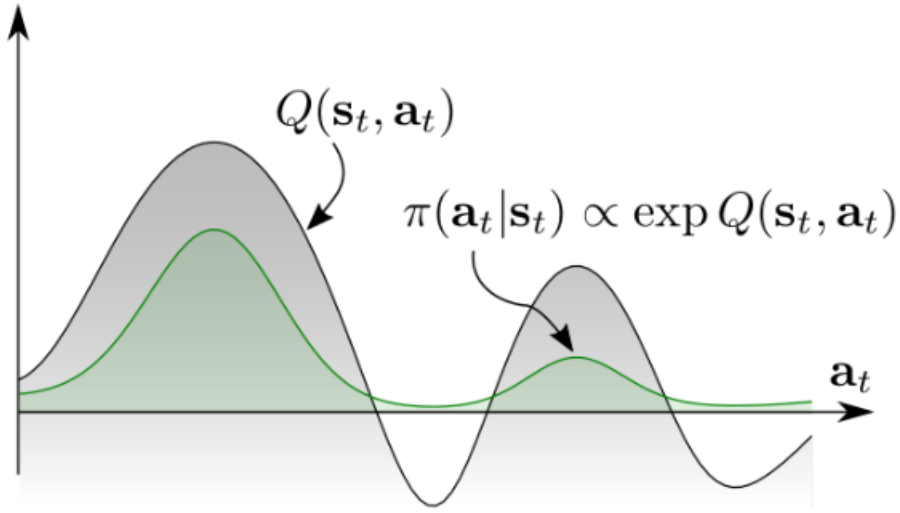
$$\pi_{ent} = \underset{\pi}{\operatorname{argmax}} \sum_{t} \gamma^t \underset{s_t, a_t \sim \pi}{\mathbb{E}} [R_t + \alpha H(\pi(\cdot|s_t))]$$

where a is a weighting term to balance the importance of the entropy.

Preliminaries - Soft Value Functions



3a



3b

Figure 3: A multimodal Q-function.

Preliminaries - Soft Q-learning and Policy Gradient

- The soft Q-learning gradient

$$\nabla_\theta Q_\theta(s,a)(Q_\theta(s,a) - \hat{Q}(s,a))$$

$$R(s,a) + \gamma V_Q(s')$$

- Policy gradient

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(\hat{Q}_\pi - b(s_t)) + \alpha \nabla_\theta H(\pi_\theta(\cdot|s_t))\right]$$

where b(.) is some arbitrary baseline.

NAC Method

- Actor

contribution

$$\nabla_\theta J_{PG} = \mathop{\mathbb{E}}_{s,a\sim\pi_Q}\left[(\nabla_\theta Q(s,a) - \nabla_\theta V_Q(s))(Q(s,a) - \hat{Q})\right]$$

- Critic

$$\nabla_\theta J_V = \mathop{\mathbb{E}}_{s}\left[\nabla_\theta \frac{1}{2}(V_Q(s) - \hat{V}(s))^2\right]$$

where V_Q and π_Q are deterministic functions of Q:

$$V_Q(s) = \alpha \log \sum_a \exp(Q(s,a)/\alpha)$$

$$\pi_Q(a|s) = \exp((Q(s,a) - V_Q(s))/\alpha)$$

$\hat{Q}(s,a), \hat{V}(s)$ are obtained by:

$$\hat{Q}(s,a) = R(s,a) + \gamma V_Q(s')$$

$$\hat{V}(s) = \mathop{\mathbb{E}}_{a\sim\pi_Q}[R(s,a) + \gamma V_Q(s')] + \alpha H(\pi_Q(\cdot|s))$$

NAC Method – Importance Sampling

- In the demonstration set, we only have transitions from expert policy μ . To have a proper policy gradient algorithm, we can employ importance sampling to correct the mismatch.
- To be specific

$$\mathrm{E}_{(s,a)\sim\pi_Q}[f(s,a)] \approx \mathrm{E}_{(s,a)\sim\mu}[f(s,a)\beta]$$

$$\beta = \min\{\frac{\pi_Q(a\,|\,s)}{\mu(a\,|\,s)}, c\}$$

We find in our empirical evaluation that inclusion of these weights consistently reduces the performance of our method

NAC Method - Algorithm

Pre-training with Demonstration

Actor-Critic?
Actor is Q, and Critic is also Q

Algorithm 1 Normalized Actor-Critic for Learning from Demonstration

θ: parameters for the rapid Q network, θ': parameters for the target Q network, $\mathcal{D}$: demonstrations collected by human or a trained policy network, T: target network update frequency, $\mathcal{M}$: replay buffer, k: number of steps to train on the demonstrations

for step t $\in \{1, 2, ...\}$ **do**
  if t $\leq k$ **then**
    Sample a mini-batch of transitions from $\mathcal{D}$
  else
    Start from s, sample a from π, execute a, observe (s', r) and store (s, a, r, s') in $\mathcal{M}$
    Sample a mini-batch of transitions from $\mathcal{M}$
  end if
  Update θ with gradient: $\nabla_\theta J_{PG} + \nabla_\theta J_V$
  if t mod T = 0 **then**
    $\theta' \leftarrow \theta$
  end if
end for

NAC Method - Analysis

- Comparing the actor update of NAC with the soft Q-learning update, our method includes an extra term in the gradient:

$$-\nabla_\theta V_Q(s) \qquad Q(s,a)\uparrow \iff V_Q(s)\downarrow$$

Normalization

- NAC is also less sensitive to noisy demonstrations.
  - NAC is an RL algorithm, it is naturally resistant to poor behaviors.
  - When there is a negative reward in the demonstrations, **Q** tends to decrease and **V** tends to increase, hence having the normalizing behavior in a reverse direction.

Experiments

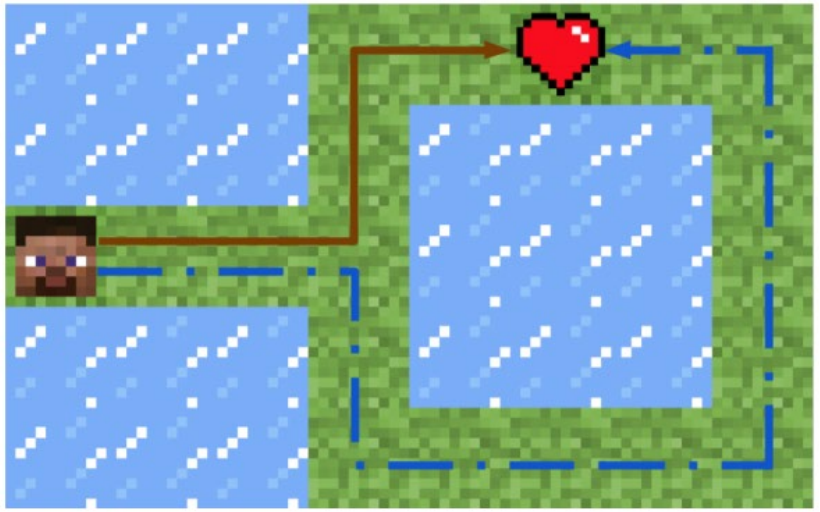
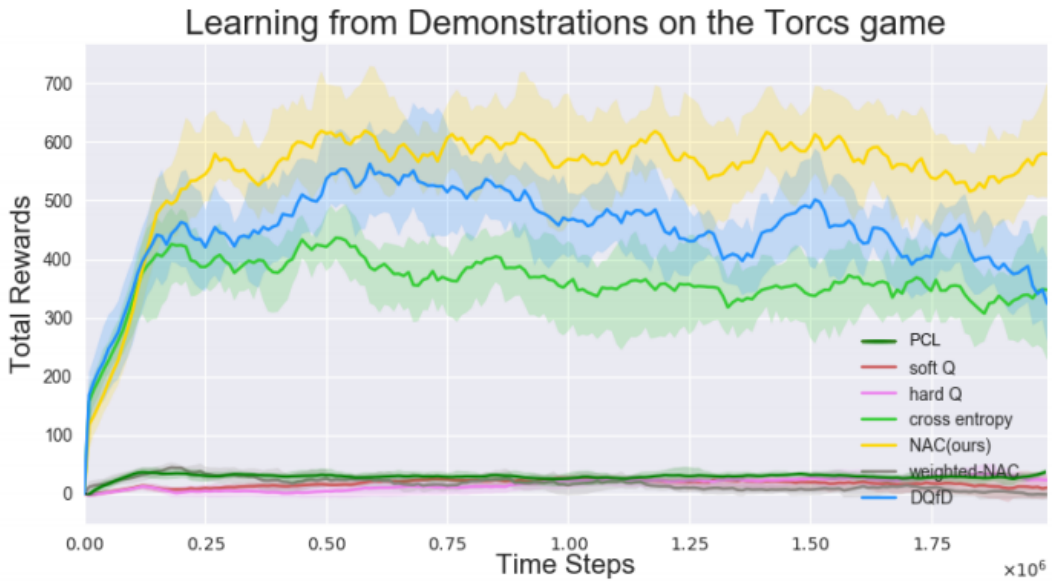


Figure 1. Sample frames from Torcs (upper) and GTA (lower).

- Environments
  - Toy Minecraft
  - Torcs
  - GTA
- Comparisons
  - DQfD (AAAI 2018)
  - Q-learning
  - Soft Q-learning
  - Behavior cloning with Q-learning (Pre training with cross-entropy + Q)
  - NAC with importance sampling
  - PCL

Experiments - Torcs game



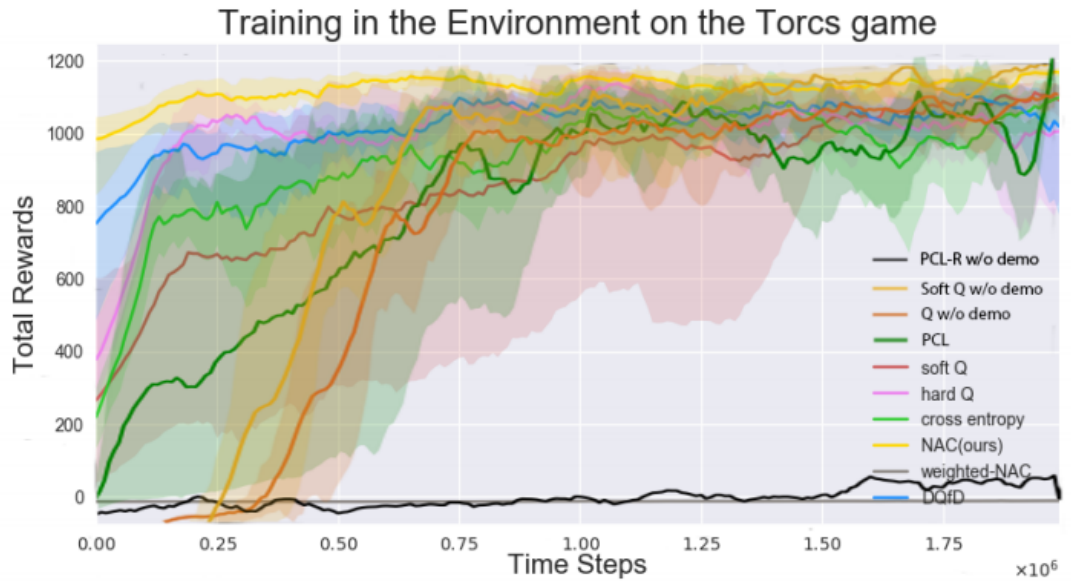
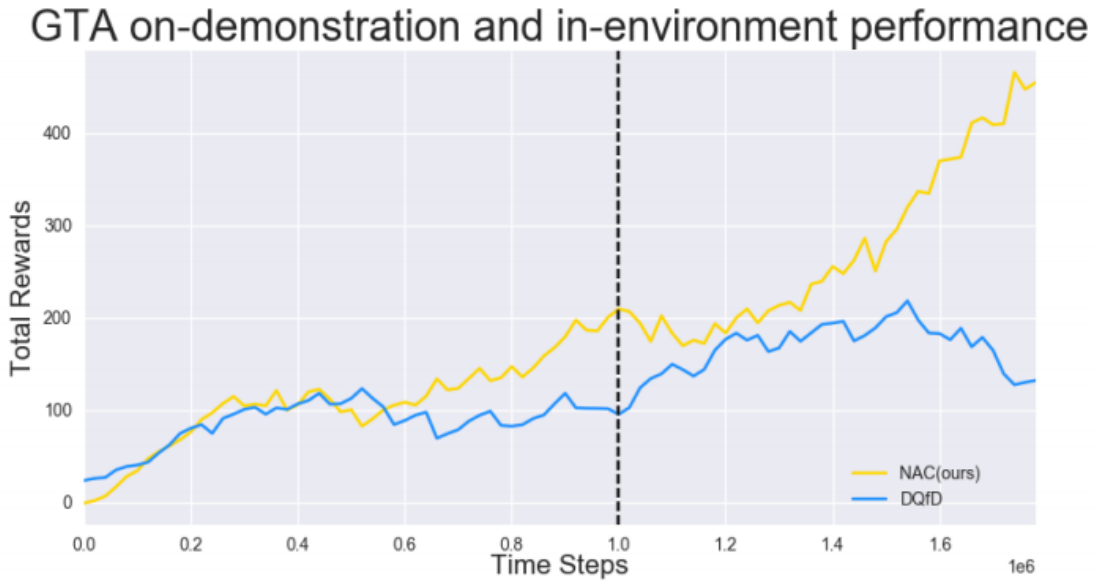
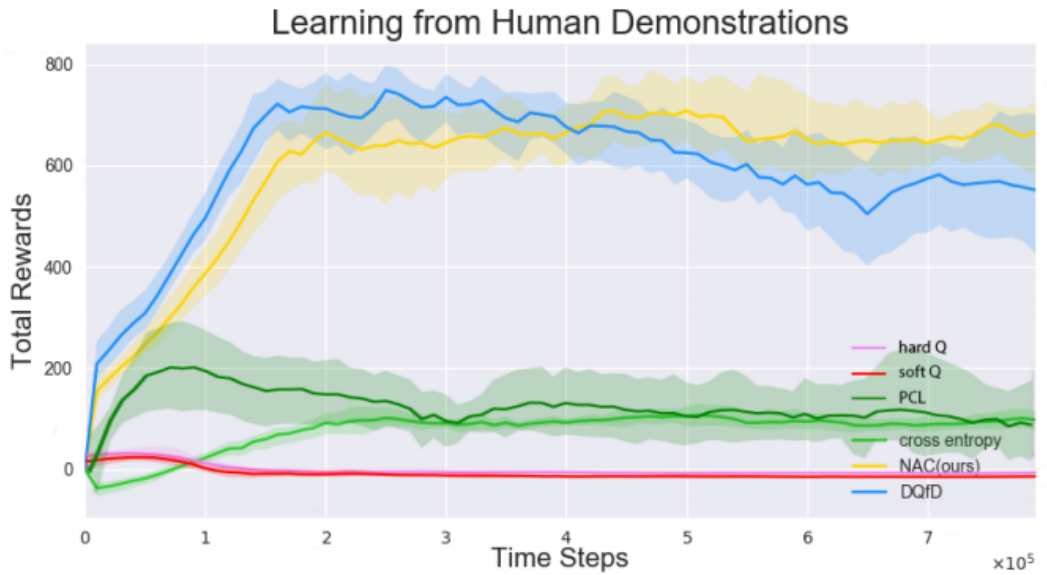


Figure 3. Performances on the Torcs game. The x-axis shows the training iterations. The y-axis shows the average total rewards. Solid lines are average values over 10 random seeds. Shaded regions correspond to one standard deviation. The left figure shows the performance for each agent when they only learn from demonstrations, while the right one shows the performance for each agent when they interact with the environments after learning from demonstrations. Our method consistently outperforms other methods in both cases.

Experiments - GTA & Torcs game with human demonstrations

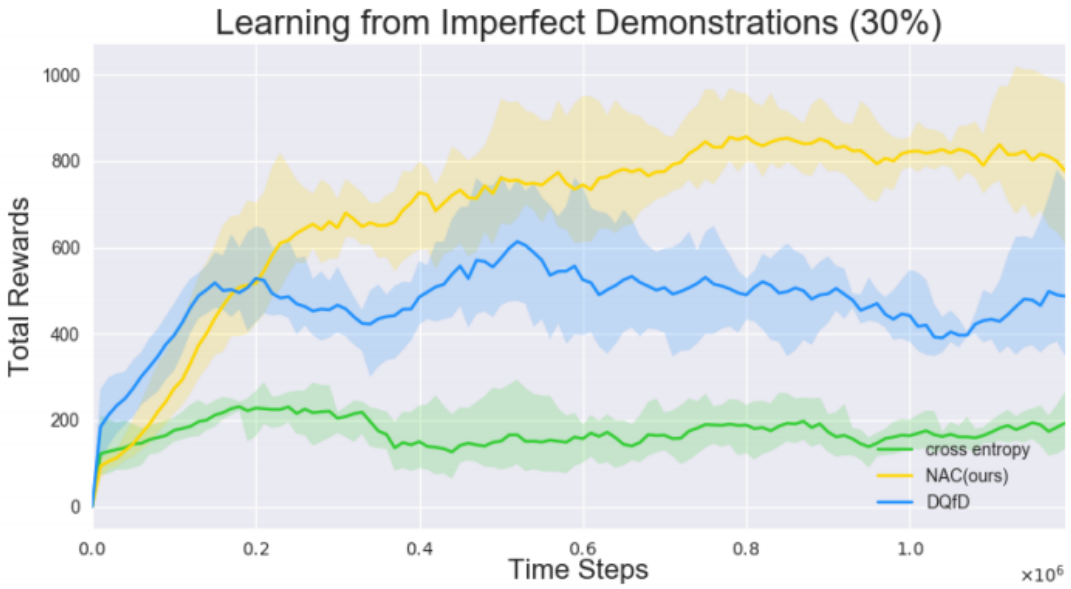


(a) The on-demonstration and in-environment performance of the NAC and DQfD methods on GTA. The vertical line separates the learning from demonstration phase and finetuning in environment phase. Our method consistently outperforms DQfD in both phases.



(b) Performances on the Torcs game with human demonstrations. DQfD performs well in the beginning, but overfits in the end. The behavior cloning method is much worse than NAC and DQfD. Our NAC method performs best at convergence.

Experiments - Torcs game



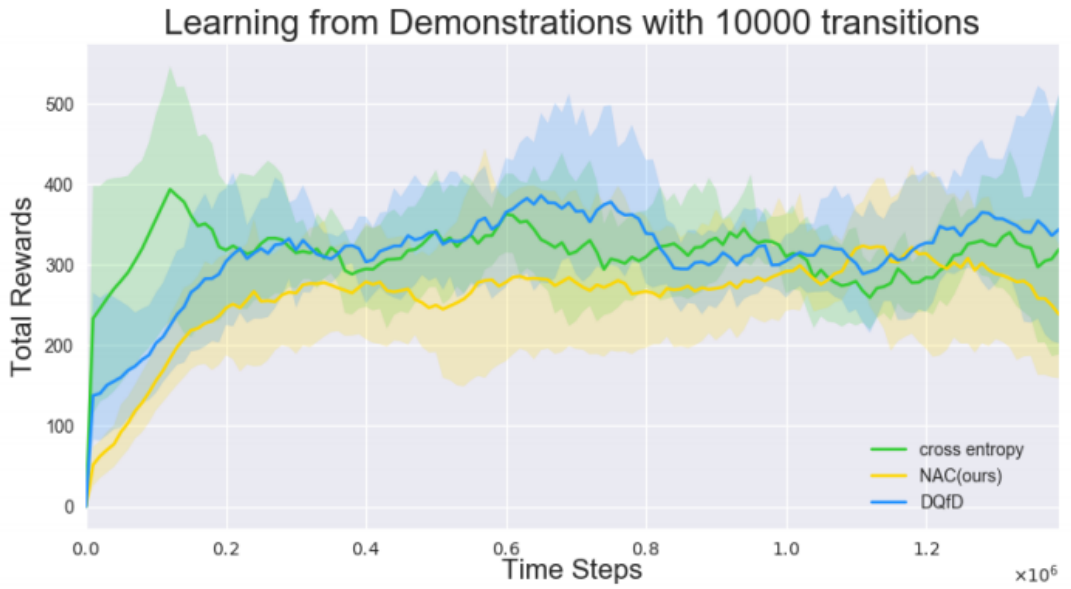


Figure 5. Left: Learning from imperfect data when the imperfectness is 30%. Our NAC method does not clone suboptimal behaviors and thus outperforms DQfD and behavior cloning. Right: Learning from a limit amount of demonstrations. Even with only 30 minutes (10k transitions) of experience, our method could still learn a policy that is comparable with supervised learning methods. More results are available in the appendix, including 50% and 80% imperfect data ablations, as well as 150k and 300k data amount studies.

Experiments

$$Q(s_i, a_i)$$

$$Q(s_i, a_i^*)$$

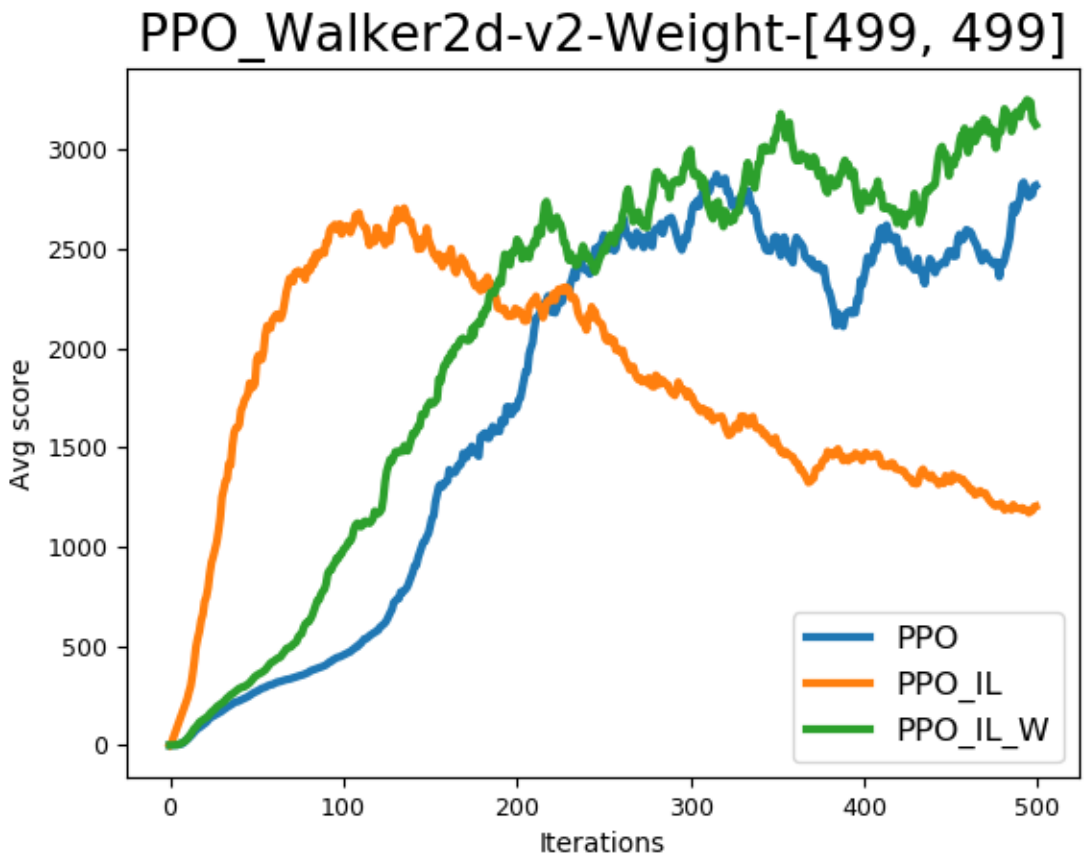
$$R_i^* = \sum_{t=i}^{m} \gamma^{t-i} r_t^*$$

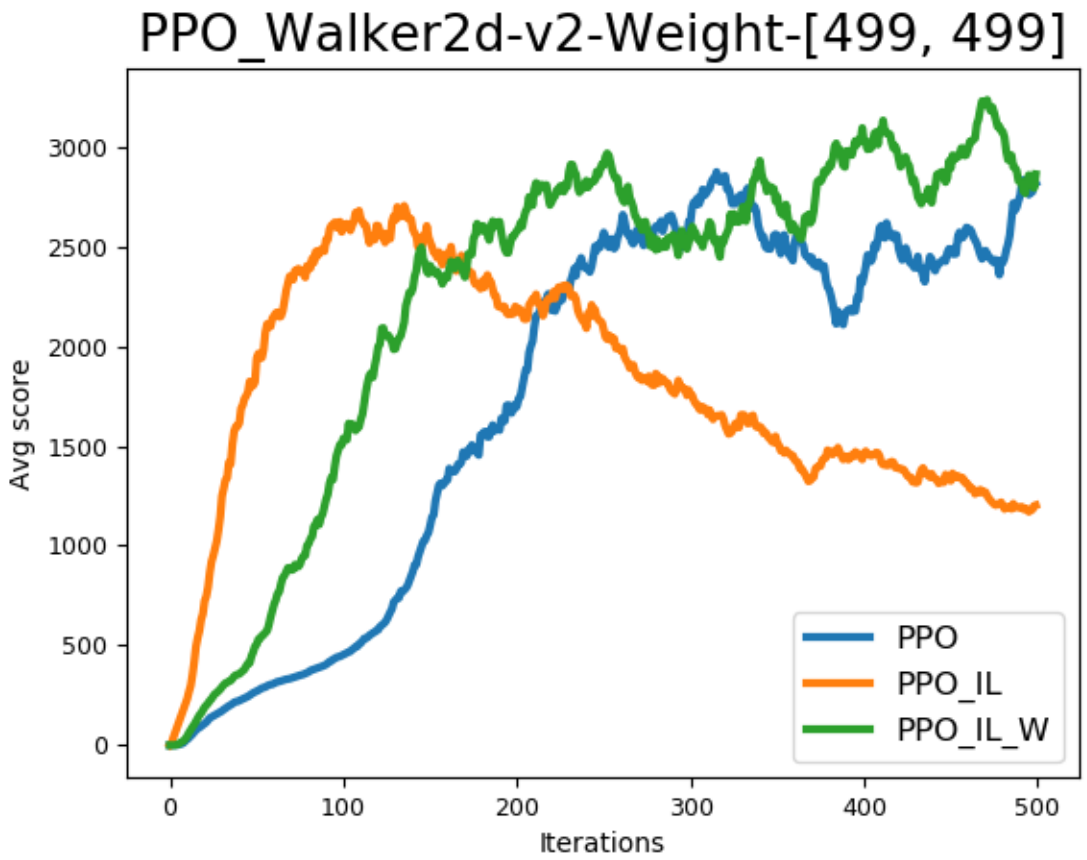
- Imitation learning
- Mujoco Environment

$$\ell = \ell_{PG} + \frac{1}{m} \sum_{i=1}^{m} \lambda_i * \ell_{SL}(s_i, a_i^*)$$

where $$\lambda_i = 1\{Q(s_i, a_i) - R_i^* \geq 1\}$$

Experiments - PPO_Walker2d

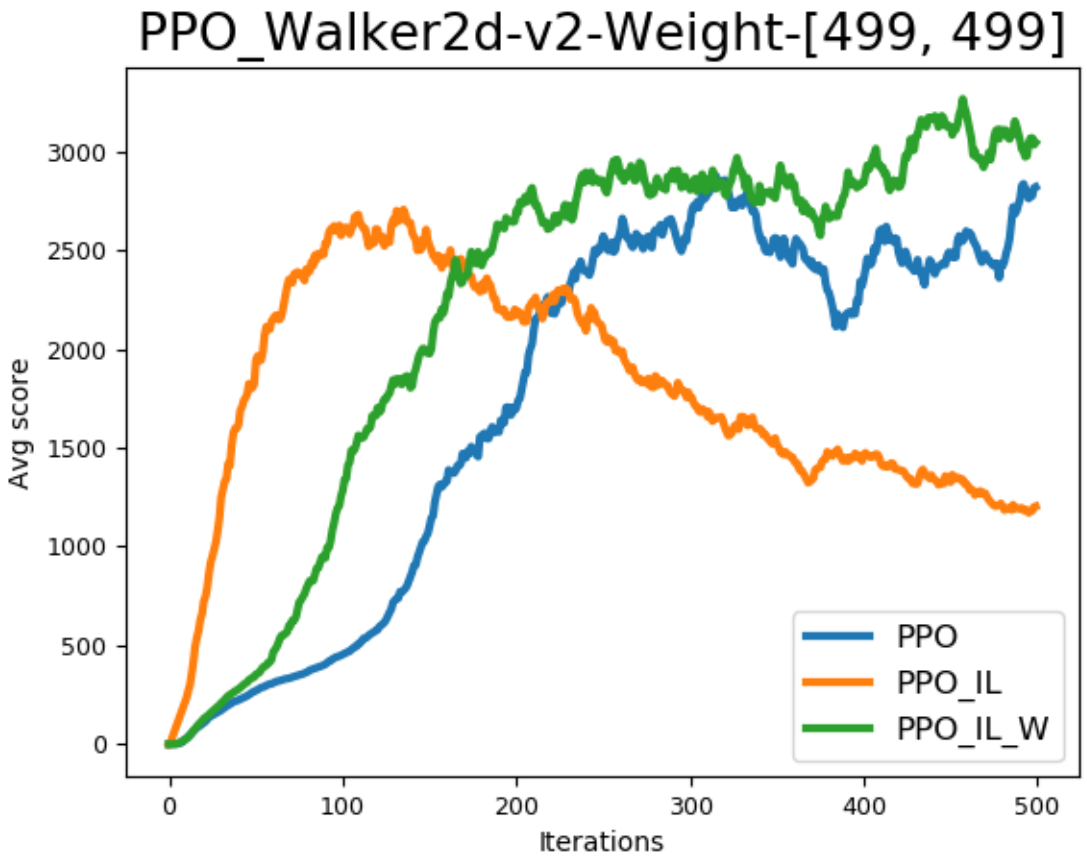


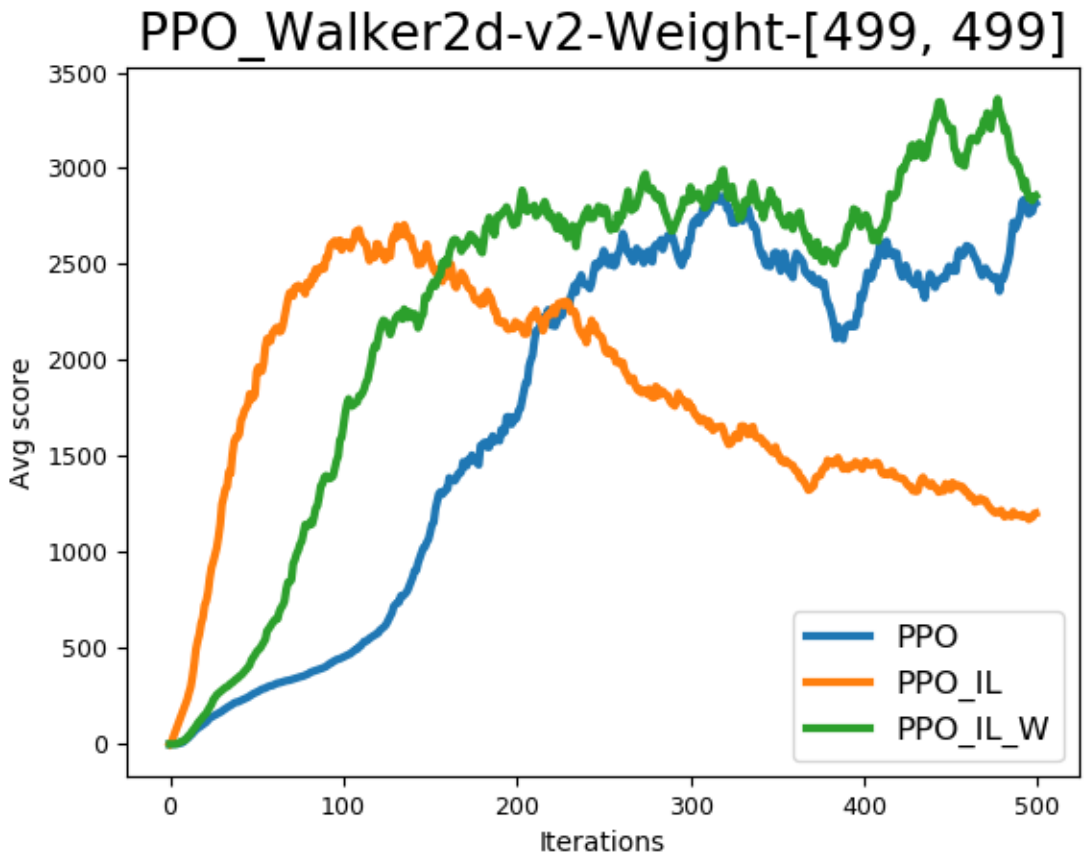


$$\lambda_i = 1\{Q(s_i, a_i) - R_i \geq 1\}$$

$$\lambda_i = \max\{\frac{Q(s_i, a_i) - R_i}{\delta}, 0\}$$

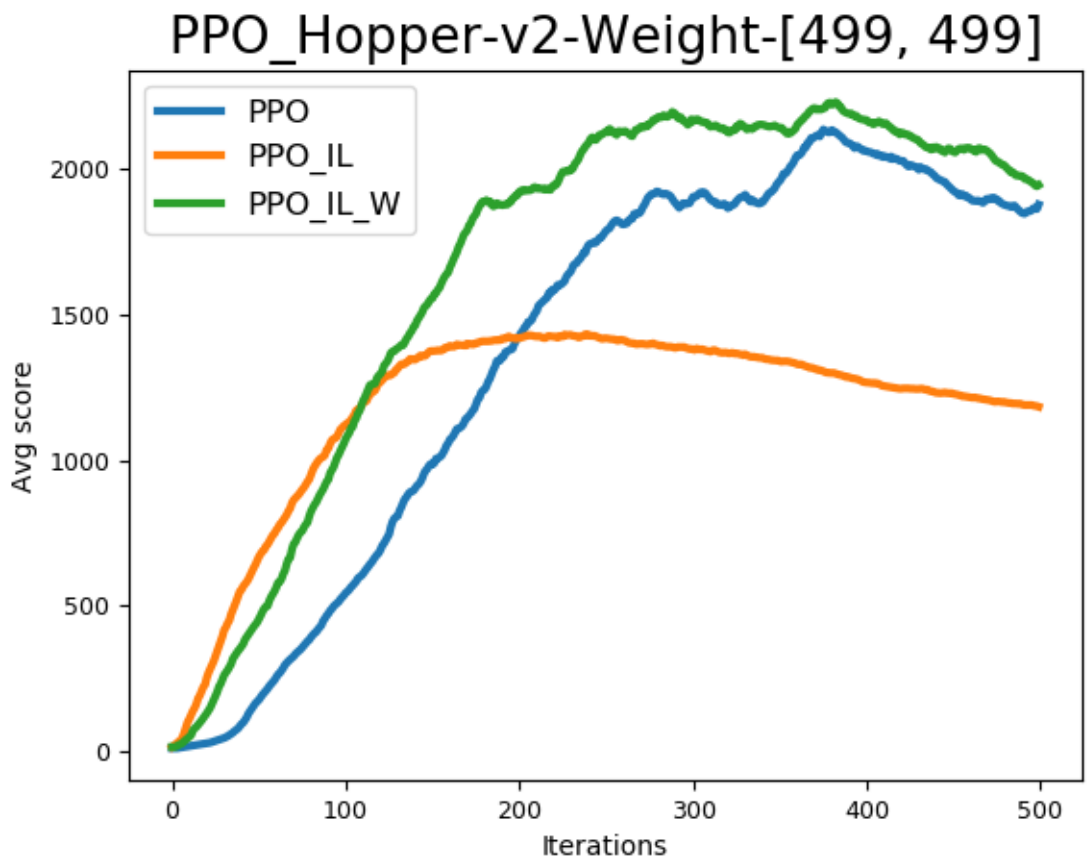
Experiments - PPO_Walker2d

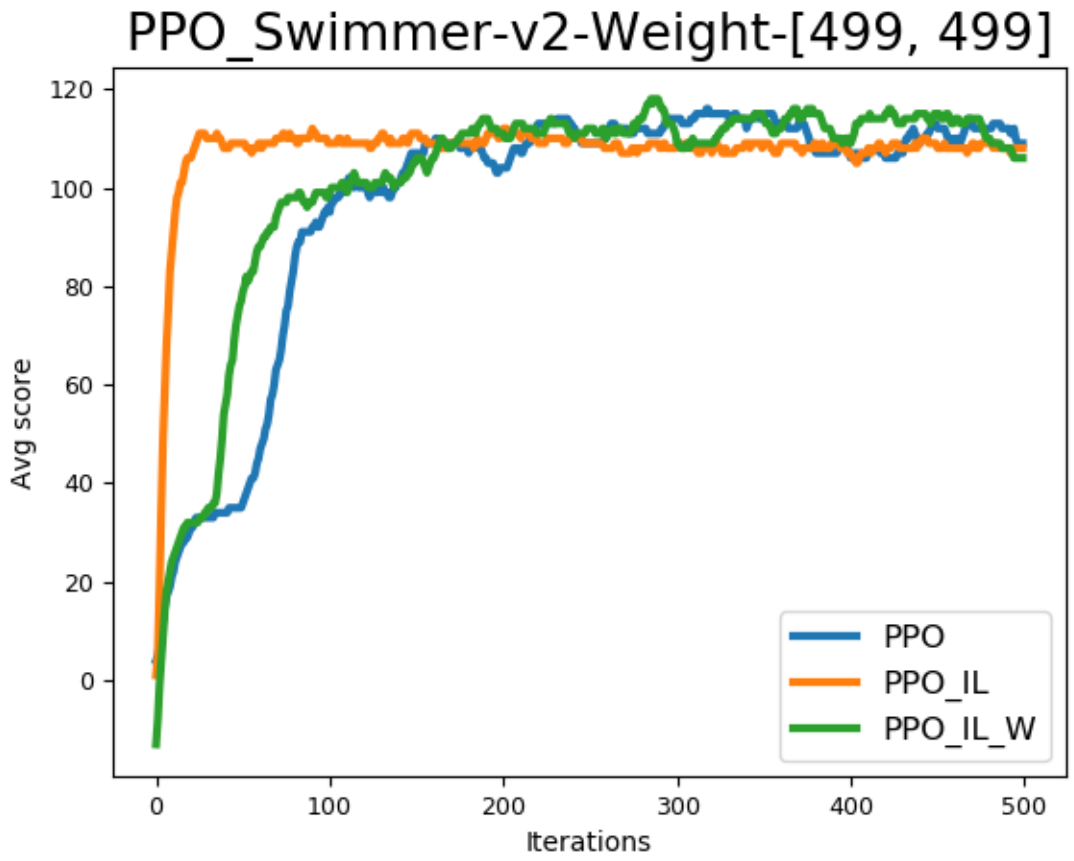




$$\lambda_i = \log(\max\{Q(s_i, a_i) - R_i, 1\})$$

Experiments - PPO_Hopper&Swimmer





$$\lambda_i = \log(\max\{Q(s_i, a_i) - R_i, 1\})$$

Experiments

- Mujoco Environment
  - RLfD + Re-Weighting
  - Baseline (PPO, TRPO, ATRPO, PG) without any demonstrations
  - Imitation Learning
  - RLfD

- Classic control
  - 验证使用demonstrations学策略网络的前几层，RL学策略网络的后几层，且将前几层迁移到新的算法上能快速收敛。