Policy optimization with Demonstrations

Bingyi Kang, Zequn Jie, Jiashi Feng

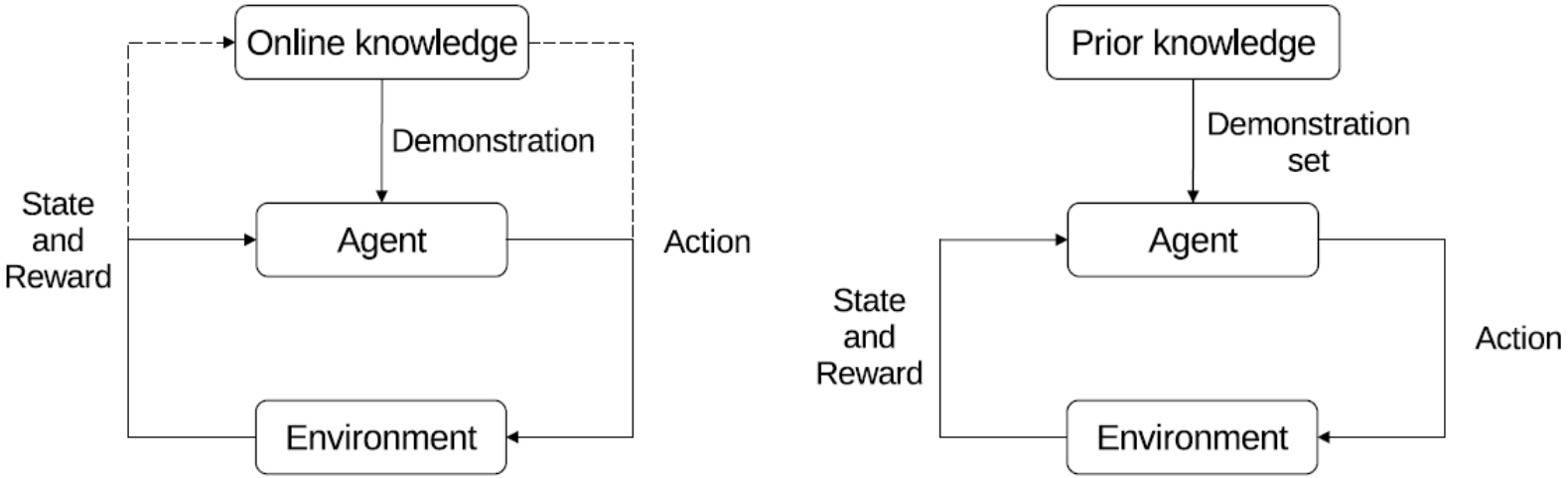
International conference on machine learning. PMLR, 2018

Speaker : Min Joon Kim

Oct 30th, 2023

Background

- **Learning from demonstrations**
  - One of the transfer learning approach to reinforcement learning.
  - LfD is a technique to assist RL by utilizing external demonstrations for more efficient exploration. (Zhu, 2023)

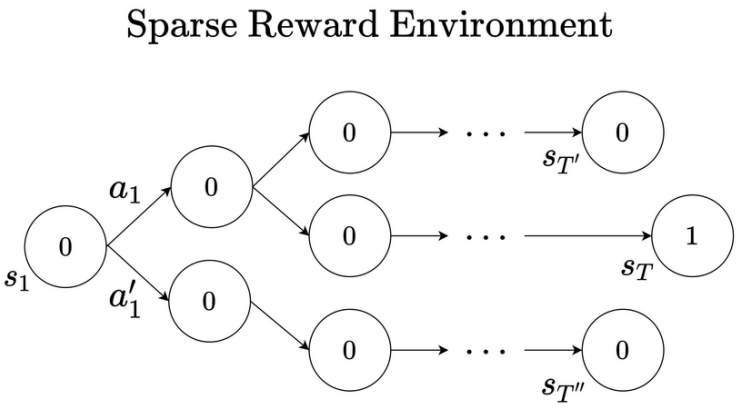


[Reinforcement Learning from Expert Demonstrations]

'Model-free reinforcement learning from expert demonstrations: a survey', Ramírez, 2022.

Problem definition

- **Policy optimization with Demonstrations**
  - Develop a policy optimization algorithm with a well-designed reward function in sparse reward environment.



[Sparse reward environment]

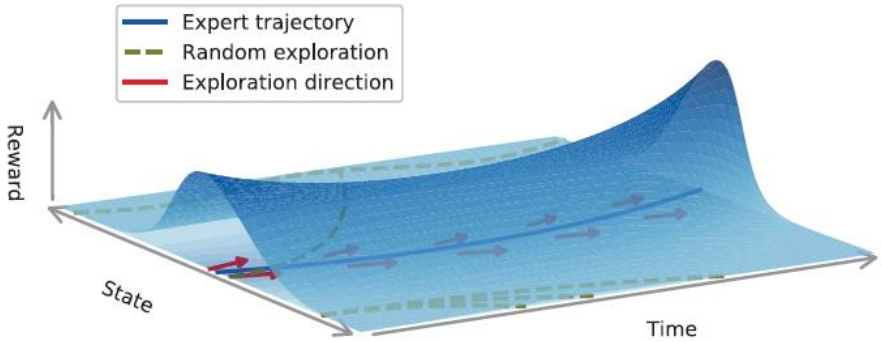


Figure 1. Our proposed POfD explores in the high-reward regions (red arrows), with the aid of demonstrations (the blue curve). It thus performs better than random explorations (olive green dashed curves) in sparse-reward environments.

[Exploration in proposed method]

Related works

Author	Paper	Keywords
Ho et al. (2016)	Generative adversarial imitation learning	GAIL, Imitation learning
Hester, Todd, et al. (2018)	Deep q-learning from demonstrations	DQfD, Replay buffer
Vecerik, Mel, et al. (2017)	Leveraging Demonstrations for Deep Reinforcement Learning on Robotics Problems with Sparse Rewards	DDPGfD, Replay buffer
Rengarajan et al. (2022)	Reinforcement learning with sparse rewards using guidance from offline demonstration	LOGO(Learning Online with Guidance Offline)

- **Limitation of previous work**
  - Slow convergence, Discrete action space, Poor performance in sparse reward environment

- **In this work**
  - Reward shaping with expert demonstrations
  - Can be applied in PPO, TRPO (continuous action space)

Method

- **Loss function for POfD**

$$\mathcal{L}(\pi_\theta) = -\eta(\pi_\theta) + \lambda_1 D_{JS}(\pi_\theta, \pi_E),$$

expected return Distance between two polices(Jensen-Shannon divergence)

Entropy loss

$$\min_\theta \max_w \mathcal{L} = -\eta(\pi_\theta) - \lambda_2 H(\pi_\theta) + \lambda_1 \left(\mathbb{E}_{\pi_\theta}[\log(D_w(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D_w(s,a))]\right). \quad (6)$$

Similar to GAN loss function

- Labeling expert state-action pairs as true("1") and policy state-action pairs as false("0")

$$\min_\theta \max_w -\mathbb{E}_{\pi_\theta}[r'(s,a)] - \lambda_2 H(\pi_\theta) + \lambda_1 \mathbb{E}_{\pi_E}[\log(1 - D_w(s,a))], \quad (7)$$

where $r'(s,a) = r(a,b) - \lambda_1 \log(D_w(s,a))$ Reward shaping

Method(algorithm)

Algorithm 1 Policy optimization with demonstrations

Input: Expert demonstrations $\mathcal{D}_E = \{\tau_1^E, \ldots, \tau_N^E\}$, initial policy and discriminator parameters θ_0 and w_0, regularization weights λ_1, λ_2, maximal iterations I.

for $i = 1$ to I **do**

Sample trajectories $\mathcal{D}_i = \{\tau\}, \tau \sim \pi_{\theta_i}$.

Sample expert trajectories $\mathcal{D}_i^E \subset \mathcal{D}^E$.

Update discriminator parameters from w_i to w_{i+1} with the gradient

$$\hat{\mathbb{E}}_{\mathcal{D}_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\mathcal{D}_i^E}[\nabla_w \log(1 - D_w(s,a))]$$

Update the rewards in $\mathcal{D}_i$ with

$$r'(s,a) = r(a,b) - \lambda_1 \log(D_{w_i}(s,a)), \forall (s,a,r) \in \mathcal{D}_i$$

Update the policy with policy gradient method (*e.g.*, TRPO, PPO) using the following gradient

$$\hat{\mathbb{E}}_{\mathcal{D}_i}[\nabla_\theta \log \pi_\theta(a|s) Q'(s,a)] - \lambda_2 \nabla_\theta H(\pi_{\theta_i})$$

end for

Method(Learning procedure)

- **The procedure of POfD**

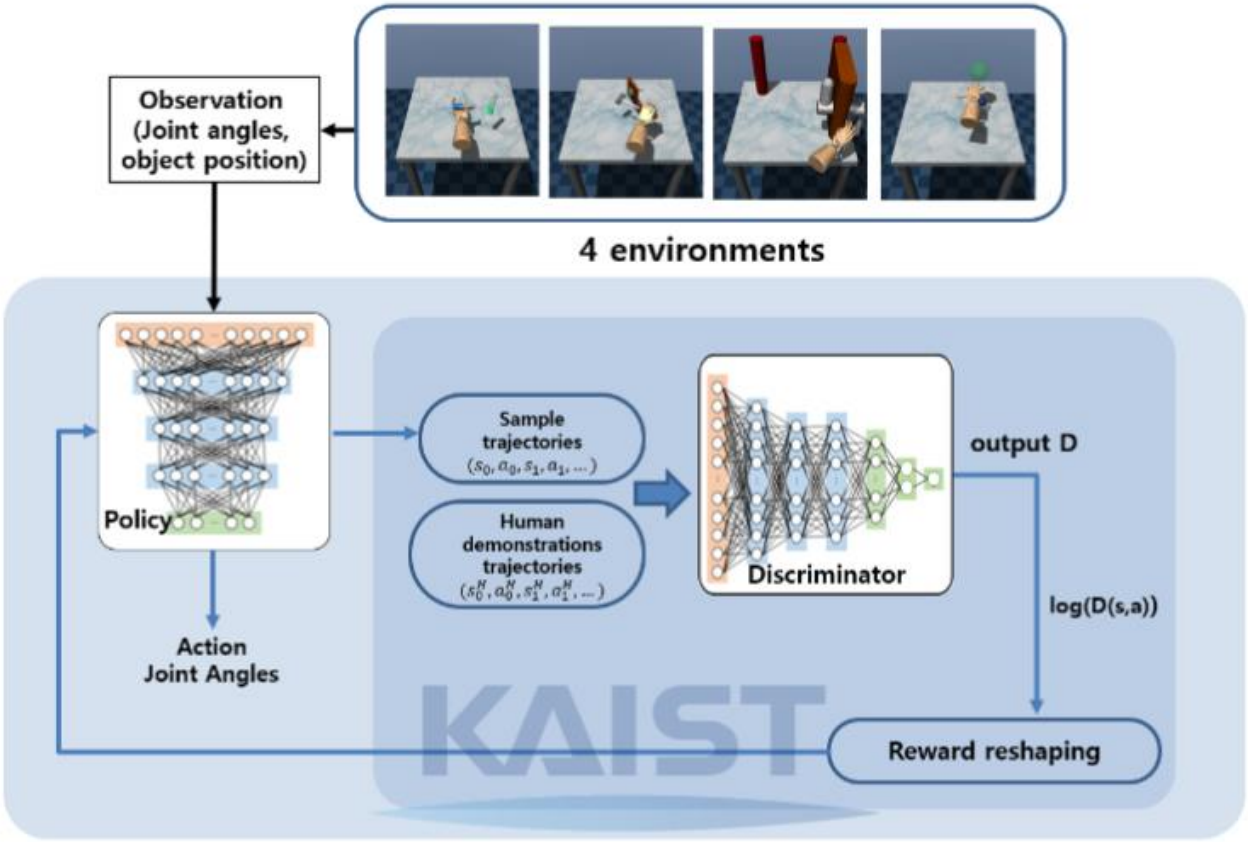
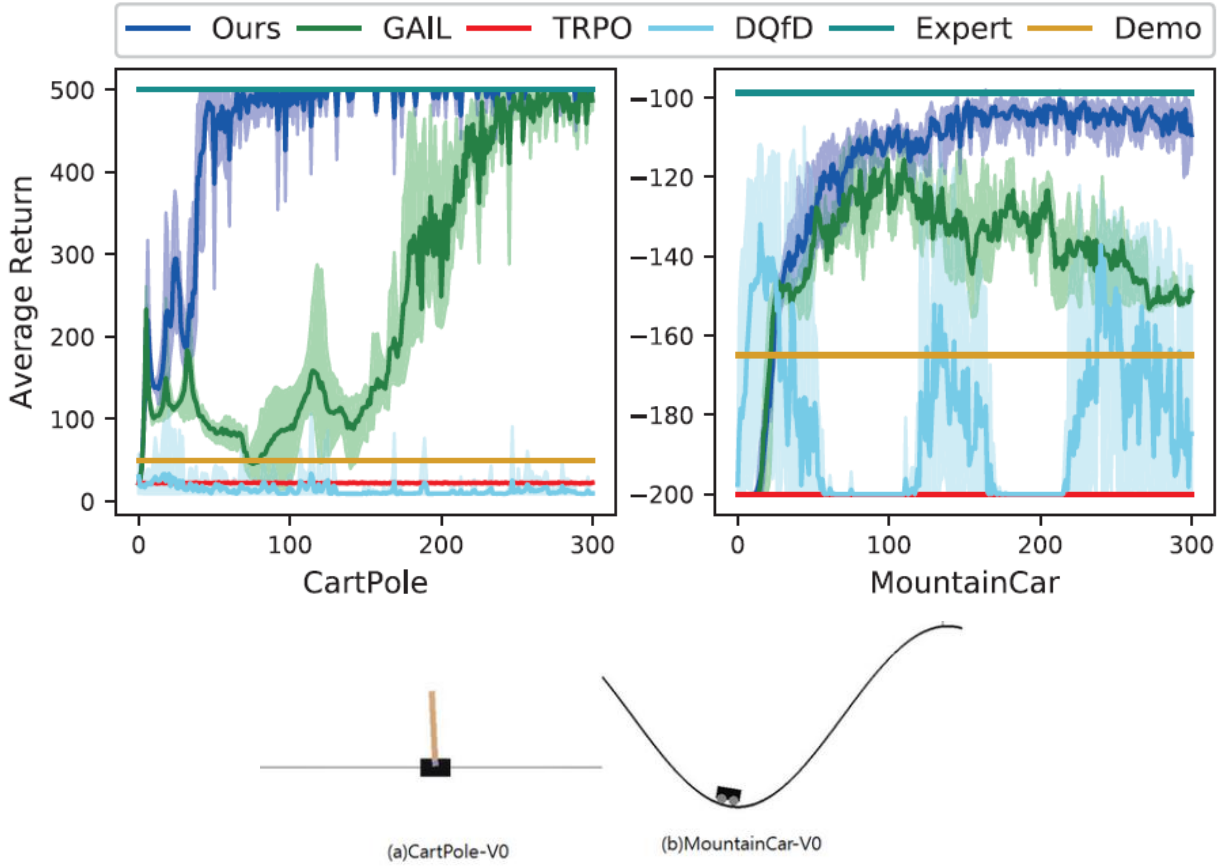


Figure 3.1: The learning procedure of Policy Optimization from Demonstration

Source : 'Policy Optimization from Demonstrations with Behavior Cloning for Robot Hand Manipulation' Choi, 2020

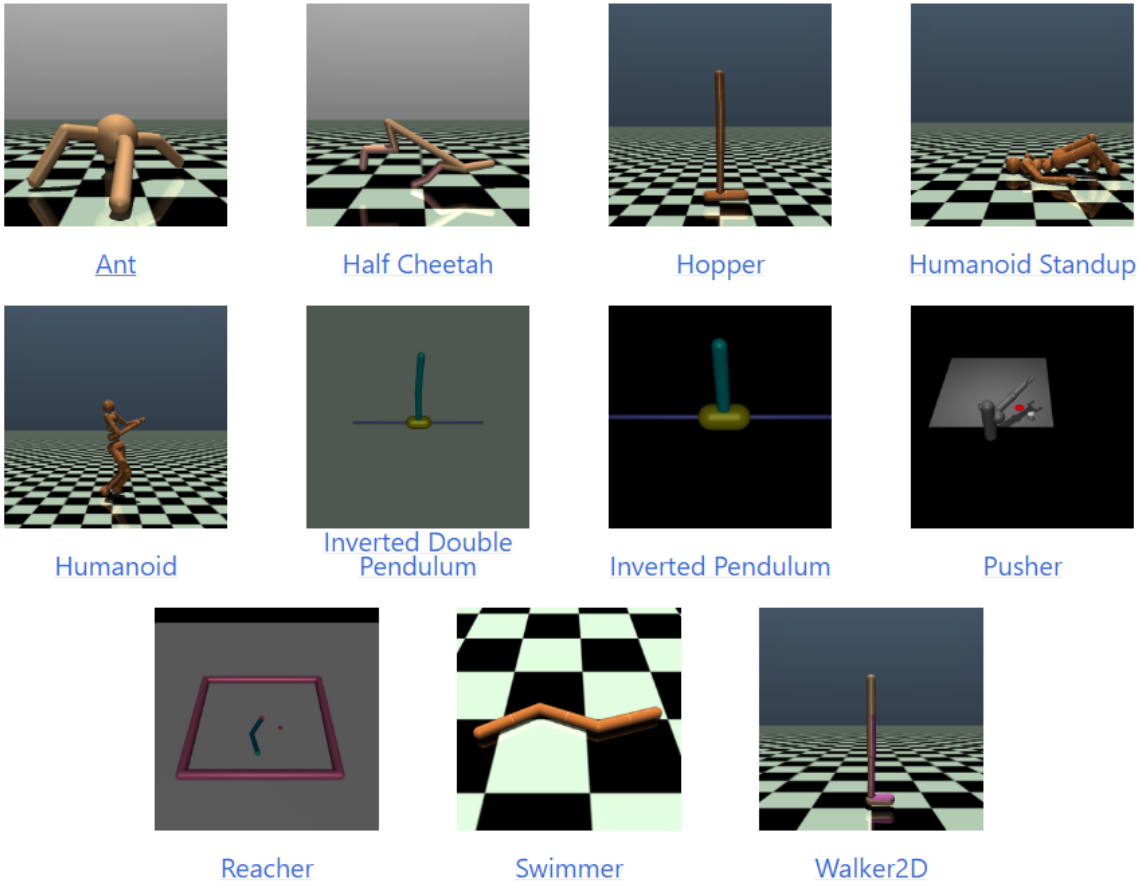
Experiments

- Compare with the baselines (Classic control)
  - Proposed method can acheive the highest score in both games.
  - Get faster convergence than other methods.



Experiments

- Compare with the baselines (MuJoCo)
  - More complicated environments(continuous action)



Experiments

- Compare with the baselines (MuJoCo)
  - More complicated environments(continuous action)

Environment	$\mathcal{S}$	$\mathcal{A}$	Sparsification	Empirical Return		
				Demonstration	Expert	Ours
MountainCar-v1	$\mathbb{R}^2$	$\{0, 1, 2\}$	TYPE1	-165.0	-98.75	$\mathbf{-98.35 \pm 9.35}$
CartPole-v0	$\mathbb{R}^4$	$\{0, 1\}$	TYPE2	49	**500**	$\mathbf{500 \pm 0}$
Hopper-v1	$\mathbb{R}^{11}$	$\mathbb{R}^3$	TYPE3 (1 unit)	793.86	3571.38	$\mathbf{3652.23 \pm 263.62}$
HalfCheetah-v1	$\mathbb{R}^{17}$	$\mathbb{R}^6$	TYPE3 (15 unit)	1827.77	4463.46	$\mathbf{4771.15 \pm 646.96}$
Walker2d-v1	$\mathbb{R}^{17}$	$\mathbb{R}^6$	TYPE3 (1 unit)	1701.13	6717.08	$\mathbf{7687.47 \pm 394.97}$
DoublePendulum-v1	$\mathbb{R}^{11}$	$\mathbb{R}$	TYPE4	520.23	8399.86	$\mathbf{9116.08 \pm 1290.74}$
Humanoid-v1	$\mathbb{R}^{376}$	$\mathbb{R}^{17}$	TYPE3 (1 unit)	2800.05	9575.40	$\mathbf{9823.43 \pm 2015.48}$
Reacher-v1	$\mathbb{R}^{11}$	$\mathbb{R}^2$	TYPE1	0.73	0.75	$\mathbf{0.86 \pm 0.34}$

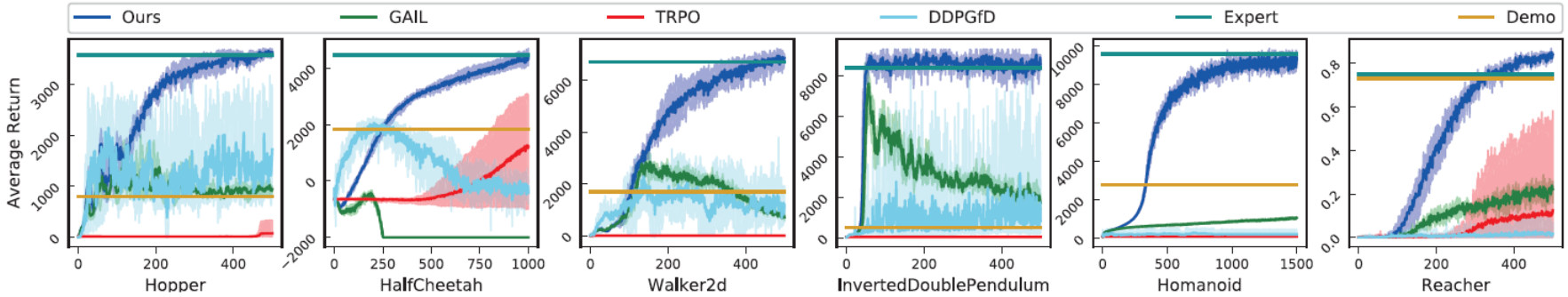


Figure 3. Learning curves of our POfD versus baselines under sparse environments with continuous action space.

Conclusion

Conclusion

- They suggested demonstration method to overcome the sparse reward problem.
- Proposed method is compatible with any other PG methods(PPO, TRPO)
- Achieve the better performance with few and imperfection demonstration data.

Contribution

- Reward reshaping with generative adversarial training.
- Theoritical derivation of loss function.

Further research

- How to apply in dual stocker scheduling problem
  - Useful in unknown reward function environment?
  - Using a makespan reward when schedule is done.

End of document