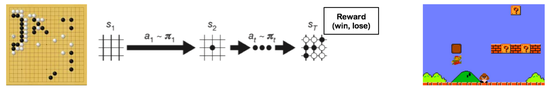
Learning Self-Imitating Diverse Policies

Tanmay Gangwani [1] Qiang Liu [2] Jian Peng [1]

[1]University of Illinois, Urbana-Champaign [2]The University of Texas at Austin

Introduction

Policy-gradient RL algorithms struggle in environments with delayed or episodic rewards.



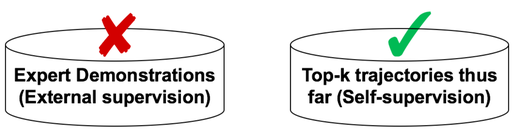
Our contributions:

- Self-imitation (SI): Exploiting useful agent behavior from the past, to improve temporal credit assignment.
- Exploration via a diverse ensemble of Self-imitation agents, using Stein Variational Gradient Descent (SVGD).

SI with GAIL

- GAIL (Ho & Ermon, 2016) frames imitation learning as matching the state-action visitation distributions of the expert and the policy.
- **Exploiting the past:** Maintain a priority buffer $\mathcal{B}$ of high return trajectories from the previous policy rollouts, and optimize the policy to match the state-action visitation distribution in the buffer.

$$\min_{\pi} \mathbb{D}(\rho_\pi, \{s_i, a_i\}_{\mathcal{B}}) \qquad \mathbb{D} : \text{distance metric}$$



- Similar to GAIL, using Jenson-Shanon divergence as the distance metric gives a min-max objective,

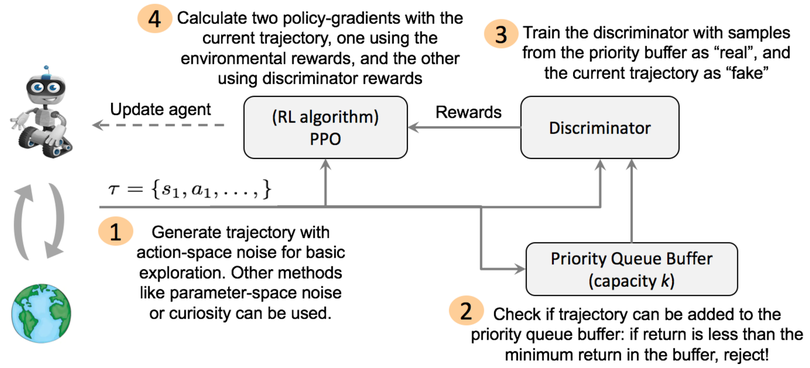
$$\min_{\pi} \max_{D} \mathbb{E}_{(s,a)\sim\rho_{\mathcal{B}}}[\log D(s,a)] + \mathbb{E}_{(s,a)\sim\rho_\pi}[\log(1 - D(s,a))]$$

- **Combining RL and SI:**

$$\nabla_\theta \eta(\pi_\theta) = \nabla_\theta \mathbb{E}_{(s,a)\sim\rho_\pi}[r(s,a)] + \beta \nabla_\theta \mathbb{E}_{(s,a)\sim\rho_\pi}[-\log(1 - D(s,a))]$$

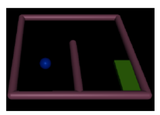
  - Environmental rewards r(s,a) can be sparse or delayed.
  - Synthetic rewards $-\log(1 - D(s,a))$ are available for each (s,a), helping with temporal credit assignment.

Algorithm Schema



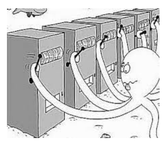
Diverse Policy Ensemble with SVGD

Motivation: A key limitation of this Self-imitation approach is the reliance on the presence of high return trajectories in the priority buffer. A few failure cases are illustrated.



Buffer has random trajectories, and SI is not useful until the reward is obtained atleast once.

Buffer has undesirable trajectories, and SI hastens convergence to sub-optimal reward.



2-armed bandit with reward distributions Bernoulli (p) and (p+ϵ) makes it difficult to converge to the optimal policy with SI.

Proposed Solution: Train an ensemble of interacting Self-imitation agents, with an explicit requirement for diversity.

- Max-entropy objective: $\max_q \mathbb{E}_{\theta\sim q}[\eta(\theta)] + \alpha H(q)$
- SVGD as the Bayesian inference algorithm to sample from the resulting energy-based distribution. It represents q with an ensemble of particles (policies) $\{\theta_i\}_{i=1}^n$, which are iteratively updated as:

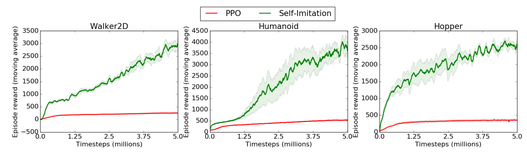
$$\theta_i \leftarrow \theta_i + \epsilon \Delta\theta_i, \qquad \Delta\theta_i = \frac{1}{n}\sum_{j=1}^{n}\left[\nabla_{\theta_j}\eta(\pi_{\theta_j})k(\theta_j,\theta_i) + \alpha\nabla_{\theta_j}k(\theta_j,\theta_i)\right]$$

- Kernel based on JS divergence: $k(\theta_j,\theta_i) = \exp(-D_{JS}(\rho_{\pi_{\theta_j}}, \rho_{\pi_{\theta_i}})/T)$
  - $\Delta\theta_i$ includes a term of the form: $\nabla_{\theta_i} D_{JS}(\rho_{\pi_{\theta_j}}, \rho_{\pi_{\theta_i}})$, which is the **repulsion gradient**, pushing policies i and j apart in the state-action space.
  - Repulsion gradients are calculated using the policy-gradient theorem, with rewards obtained from trained discriminators (one for each policy pair i,j).

Results

Experiment I: Compare single-agent Self-imitation to standard policy-gradient RL in MuJoCo environments with episodic rewards.

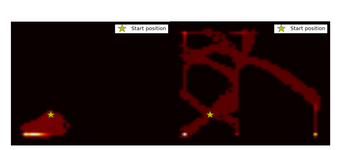
	SI	PPO
Walker	2996	252
Humanoid	3602	532
H-Standup ($\times 10^4$)	18.1	4.4
Hopper	2618	354
Swimmer	173	21
Invd.Pendulum	8668	344

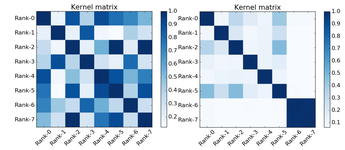


- *More in paper:* Evaluation on noisy environments (each reward r_t masked to zero with some probability); comparison to CEM and ES.

Experiment II: Elucidate failure of Self-imitation in harder tasks, and measure the benefit of training an ensemble of SI agents with explicit D_{JS} repulsion between policies (labeled SI-interact-JS). The SI-independent baseline trains isolated SI policies and selects the best among them.

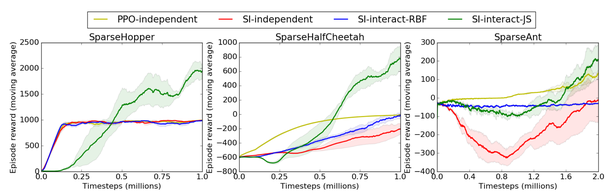
- 2D-Maze with deceptive rewards.





Left: State-density plots for SI-independent and SI-interact-JS, respectively.
Right: Final kernel matrix for SI-independent and SI-interact-JS, respectively.

- MuJoCo locomotion tasks: Forward velocity reward is only provided if the center-of-mass is beyond a certain threshold distance.
  - SI-interact-RBF uses RBF kernel for SVGD



Takeaways

a. Diversification with SVGD helps in *discovery* of sparse rewards
b. Self-imitation then helps to efficiently *exploit* the discovered rewards

Code - `github.com/tgangwani/selfImitationDiverse`
Paper Link - `https://arxiv.org/abs/1805.10309`
Correspondence - gangwan2@illinois.edu

Scan me