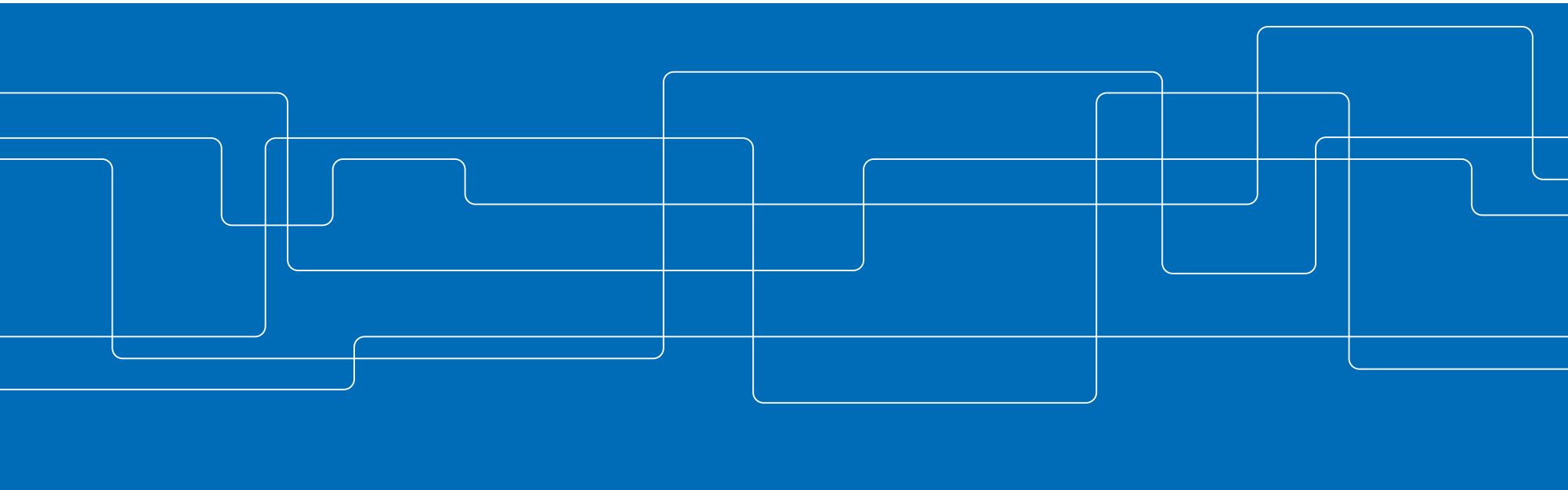
KTH ROYAL INSTITUTE
OF TECHNOLOGY

Robust Adversarial Reinforcement Learning

Authors: Lerrel Pinto, James Davison, Raul Sukthankar and Abhinav Gupta

Venue: International Conference on Machine Learning (ICML), 2017.

Presenter: Ezgi Korkmaz



Introduction

- Most current RL-based approaches fail to generalize
  - The gap between simulation and real world is large
  - Data scarcity leads to failed generalization from training to test scenarios
- H_∞ control methods
- Modeling errors and differences in training and test scenarios can be viewed as disturbances in the system
- Authors propose robust adversarial reinforcement learning
- Training agent in the presence of destabilizing adversary
- The jointly trained adversary is reinforced
- Where it learns an optimal destabilization policy
  - In games data collection is easy
  - However, data collection in real world physical tasks challenging

Introduction

- Many policy learning algorithms stochastic in nature
  - The test-generalization
  - Simulation transfer issues
- Authors claim that an approach is needed to make stable robust in learning policies across different runs and initializations while requiring less data
- High friction at test time can be modeled as extra forces at contact points
- Authors propose the idea of modelling uncertainties via an adversarial agent that applies disturbance forces to the system
- The adversary is also reinforced where it learns an optimal policy to thwart the original agent's goal.

RARL

- Proposed RARL jointly trains a pair of agents adversary and a protagonist
- Authors demonstrate the proposed the proposed approach robust to
  - Model initializations
  - Modeling errors and uncertainties
- Learn a policy to walk on carpet (training)
- Generalizes to walk on ice (test scenario)
- Authors core idea is to model the differences during training and test scenario via extra forces/disturbances in the system.
- Authors hypothesis if a policy can be learned robust to all disturbances, the this policy will be robust to changes in training and test.
- Will generalize well

RARL

- How to sample all trajectories under all possible disturbances?
- The possible disturbance space could be larger than action space.
- Adversarial agents for modeling disturbances
- Jointly train a second agent (adversary) where the goal is to impede the original agent
- Adversary is only rewarded only for the failure of the original agent
- Learn a policy to walk on carpet (training)
- Generalizes to walk on ice (test scenario)
- Authors core idea is to model the differences during training and test scenario via extra forces/disturbances in the system.

MDP

- Continuous space represented by a tuple: $(S, A, \mathcal{P}, r, \gamma, s_{\mathrm{o}})$ where,
  - S is a continuous states
  - A is continuous actions
  - $\mathcal{P} : S \times A \times S \to R$ transition probabilities
  - $r : S \times A \to R$ is the reward function
  - γ is the discount factor

Two-player zero-sum discounted games

- The adversarial setting the authors propose can be expressed as, two player discounted zero-sum Markov game,
- Can be represented as a tuple: $(S, A_1, A_2, \mathcal{P}, r, \gamma, s_0)$ where,
  - S is a continuous states
  - A_1 and A_2 is continuous set of actions
  - $\mathcal{P} : S \times A_1 \times A_2 \times S \to R$ transition probability density
  - $r : S \times A \times A_2 \to R$ is the reward of both players
  - γ is the discount factor
  - Player 1 playing strategy 1 μ
  - Player 2 playing strategy 2 v
  - Reward $r_{\mu,v} = \mathbb{E}_{a^1 \sim \mu(.|s), a^2 \sim v(.|s)}[r(s, a^1, a^2)]$
- A zero sum game can be seen Player 1 maximizing the γ discounted reward while player 2 minimizing it.

Robust Adversarial RL

- Authors goal is to learn the policy of the Player 1 which is better and robust.

$$\rho(\mu; \theta^{\mu}, \mathcal{P}) = \mathbb{E}[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) | s_0, \mu, \mathcal{P}] \tag{1}$$

- In this formulation expected reward is conditioned on the transition function.

$$\rho(\mu; \theta^{\mu}) = \mathbb{E}_{\mathcal{P}} \mathbb{E}[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) | s_0, \mu, \mathcal{P}] \tag{2}$$

- In authors setting the policy parameters θ^{μ} will be estimated such that the expected reward will be maximized over different possible transition functions

Formulating Adversarial Reinforcement Learning

- At each time step t both players observe s_t and take actions $a_t^1 \sim \mu(s_t)$ and $a_t^2 \sim v(s_t)$
- The state transitions $s_{t+1} = \mathcal{P}(s_t, a_t^1, a_t^2)$
- Rewards $r_t = r(s_t, a_t^1, a_t^2)$
- Player 1 gets $r_t^1 = r_t$
- Player 2 gets $r_t^2 = -r_t$

$$R^1 = \mathbb{E}_{s_0 \sim \rho, a^1 \sim \mu(s), a^2 \sim v(s)}[\sum_{t=0}^{T-1} r^1(s, a^1, a^2)] \tag{3}$$

RARL

Algorithm 1 RARL

input: Environment ϵ; Stochastic policies μ and v
Initialize: Learnable parameters θ^{μ}_{0} for μ and θ^{v}_{0} for v
for $i = 1, 2 \ldots N_{\text{iter}}$ **do**
  $\theta^{\mu}_{i} \leftarrow \theta^{\mu}_{i-1}$
  for $j = 1, 2 \ldots N_{\mu}$ **do**
    $\{(s^{i}_{t}, a^{1i}_{t}, a^{2i}_{t}, r^{1i}_{t}, r^{2i}_{t})\} \leftarrow roll(\epsilon, \mu_{\theta^{\mu}_{i}}, v_{\theta^{v}_{i-1}}, N_{\text{traj}})$
    $\theta^{\mu}_{i} \leftarrow policyOptimizer(\{(s^{i}_{t}, a^{1i}_{t}, r^{1i}_{t})\}, \mu, \theta^{\mu}_{i})$
  end for
  $\theta^{v}_{i} \leftarrow \theta^{v}_{i-1}$
  for $j = 1, 2 \ldots N_{v}$ **do**
    $\{(s^{i}_{t}, a^{1i}_{t}, a^{2i}_{t}, r^{1i}_{t}, r^{2i}_{t})\} \leftarrow roll(\epsilon, \mu_{\theta^{\mu}_{i}}, v_{\theta^{v}_{i-1}}, N_{\text{traj}})$
    $\theta^{v}_{i} \leftarrow policyOptimizer(\{(s^{i}_{t}, a^{1i}_{t}, r^{1i}_{t}\}, v, \theta^{v}_{i})$
  end for
end for
return $\theta^{\mu}_{N_{\text{iter}}}, \theta^{v}_{N_{\text{iter}}}$

Results

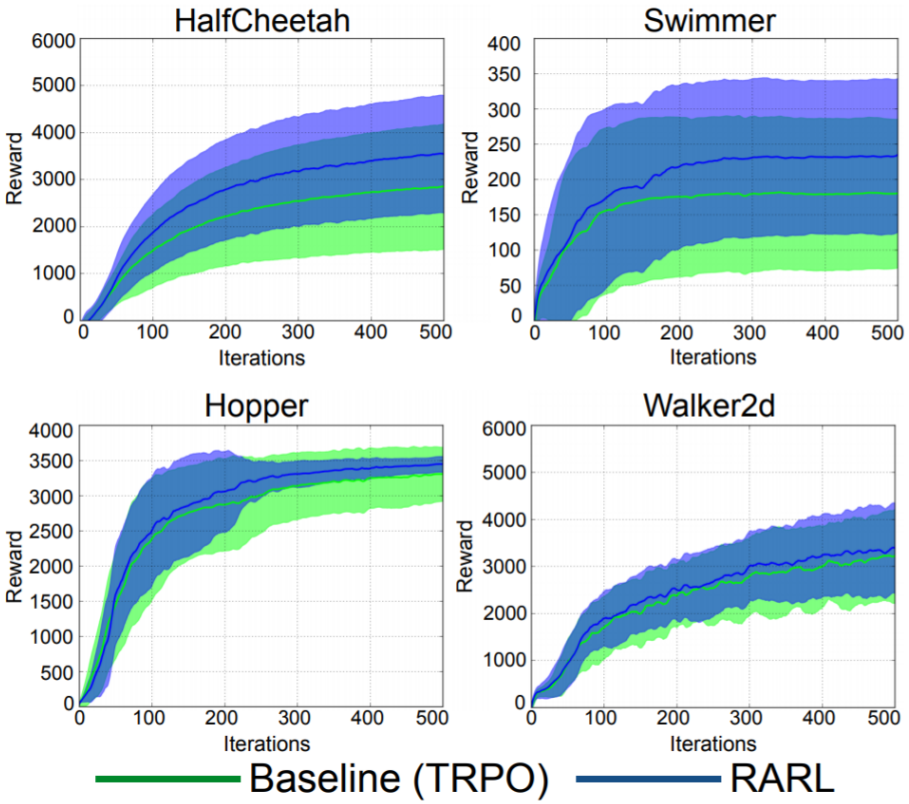


Figure: Comparison of TRPO and RARL when tested without disturbance.

Percentile Results

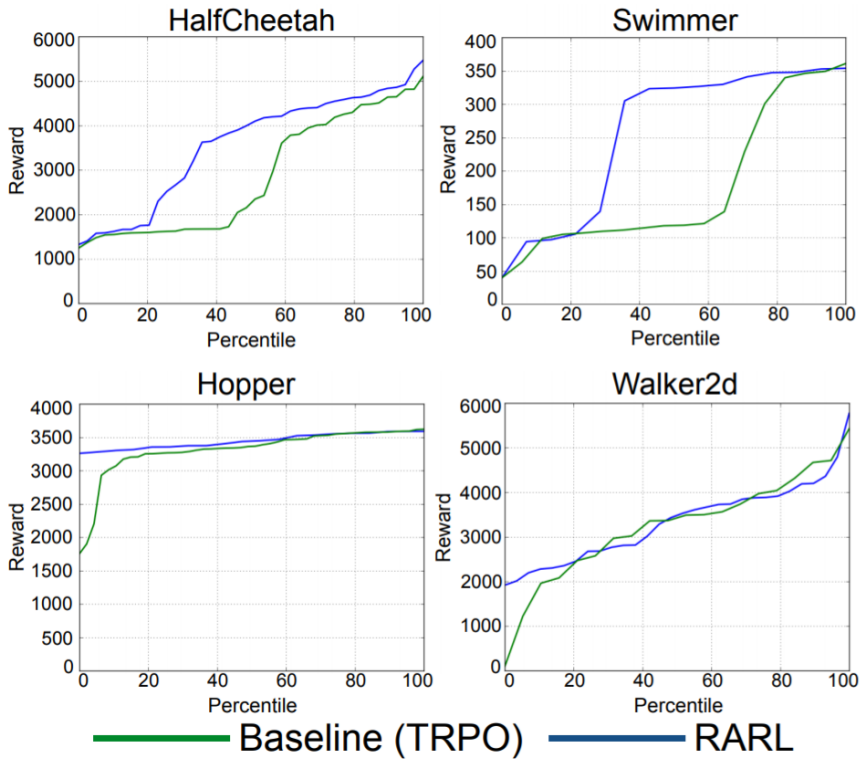


Figure: Percentile Plots without any disturbance the robustness of RARL compared to the baseline. Run on multiple initializations and sorted to show n^{th} percentile of cumulative reward.

Percentile Results

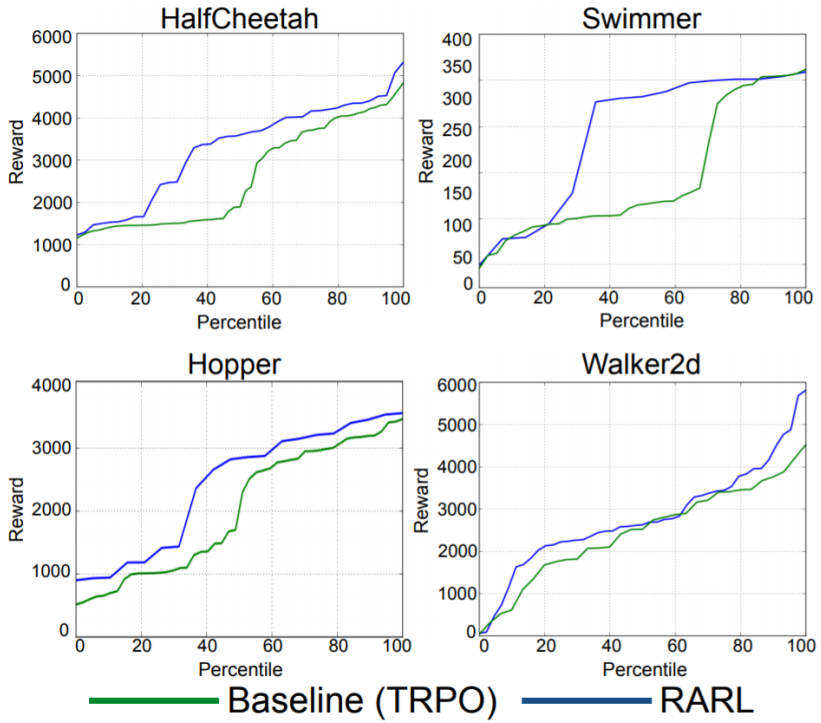


Figure: Percentile plots with learned adversarial disturbance show the robustness of RARL compared to the baseline. The algorithms are run on multiple initializations followed by learning an adversarial disturbance that is applied at test time.