Learning Robust Policy against Disturbance in Transition Dynamics via State-Conservative Policy Optimization

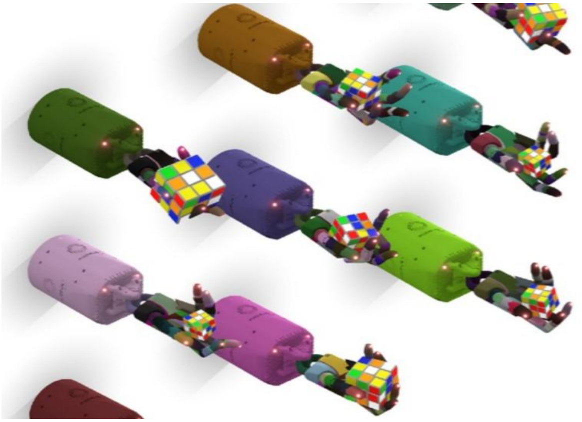
     

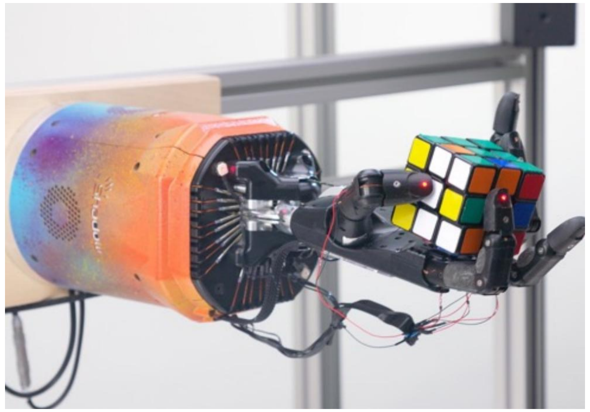
Yufei Kuang, Miao Lu, Jie Wang*, Qi Zhou, Bin Li, Houqiang Li
University of Science and Technology of China, China
* Corresponding author

Content

Part 1 Introduction

Sim-to-Real Transfer





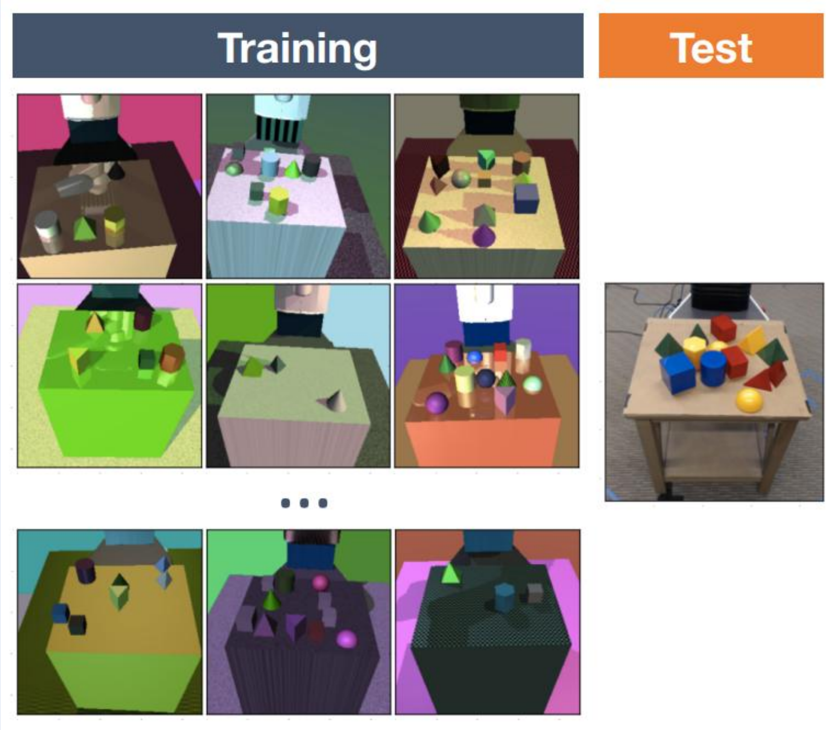
- DRL algorithms can perform poorly in real-world tasks due to
  - test-generalization
  - simulation-transfer
- Example:
  - policies trained to control robots in simulators can fail in environments with different mass or friction

[1] OpenAI et al. Solving Rubik’s cube with a robot hand. arXiv preprint 2019.

Existing Algorithms

domain randomization

$$J_{\mathcal{P}}(\pi) \triangleq \mathbb{E}_{p\sim\mathcal{P},\pi}\left[\sum_{t=0}^{+\infty}\gamma^{t} r\left(s_t, a_t\right)\right]$$



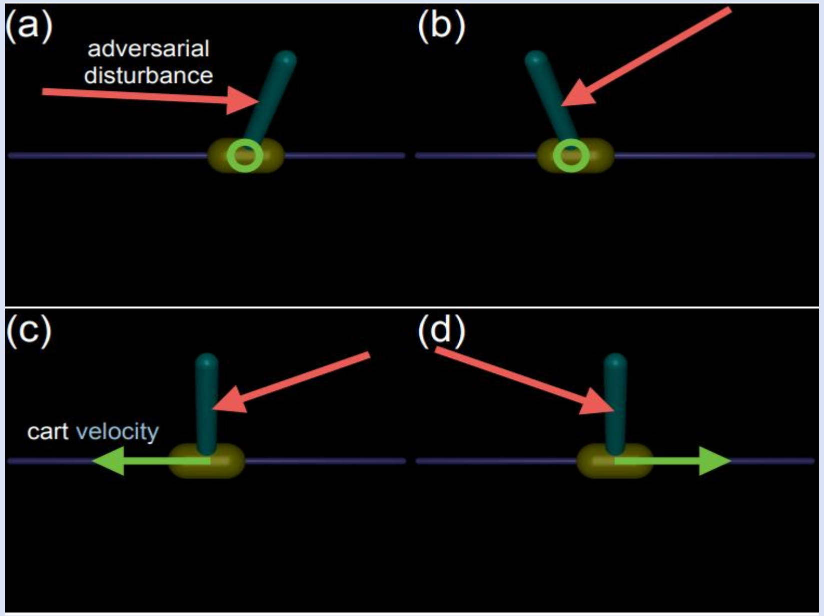
- Domain randomization:
    - uses different dynamics to approximate disturbance in target environments.
- Limitation:
    - requires pre-given uncertainty sets

[1] Tobin, J. et al. Domain randomization for transferring deep neural networks from simulation to the real world. IROS 2017.

Existing Algorithms

robust adversarial RL

$$J_{\mathcal{P}}(\pi) \triangleq \inf_{p \in \mathcal{P}} \mathbb{E}_{p,\pi}\left[\sum_{t=0}^{+\infty} \gamma^t r\left(s_t, a_t\right)\right]$$



- Robust adversarial reinforcement learning:
  - models the disturbance as trainable forces and learns by adversarial training.
- Limitation:
  - requires extra forces manually designed for each task

[1] Pinto, L. et al. Robust adversarial reinforcement learning. ICML 2017.

Challenges and Goals

Challenges:

1. require task-specific prior knowledge to model the disturbance
2. assume control of specially designed simulators

Our goals:

learn robust policies *without modeling the disturbance in advance*

Part 2 Method

Motivation: State Disturbance

Intuition: any disturbance in transition dynamics eventually influences the process via the change of future states

$$\mathcal{T}_{\mathcal{P}}^{\pi} Q \triangleq r(s,a) + \gamma \inf_{p(\cdot|s,a)\in\mathcal{P}(s,a)} \mathbb{E}_{s'\sim p(\cdot|s,a),a'\sim\pi(\cdot|s')} \left[Q\left(s',a'\right)\right] \tag{1}$$

If $\mathcal{P}$ is bounded by Wasserstein distance, then the dual problem of (1) is (2)

$$\mathcal{T}_{\mathcal{P}_\epsilon^\pi} Q(s,a) = r(s,a) + \gamma \sup_{\lambda\geq 0} \mathbb{E}_{\tilde{s}\sim p_0(\cdot|s,a)} \left\{ \inf_{s'\in\mathcal{S}} \mathbb{E}_{a'\sim\pi(\cdot|s')} \left[Q\left(s',a'\right)\right] + \lambda\left(d\left(s',\tilde{s}\right)^p - \epsilon\right) \right\}. \tag{2}$$

If $V(s)$ is convex and the environment dynamics are deterministic, then (2) is equal to (3)

$$\widehat{\mathcal{T}}_{\mathcal{P}_\epsilon}^{\pi} Q\left(s_t,a_t\right) = r\left(s_t,a_t\right) + \gamma \inf_{s'\in B_\epsilon(s_{t+1})} \mathbb{E}_{a'\sim\pi(\cdot|s')} \left[Q\left(s',a'\right)\right]. \tag{3}$$

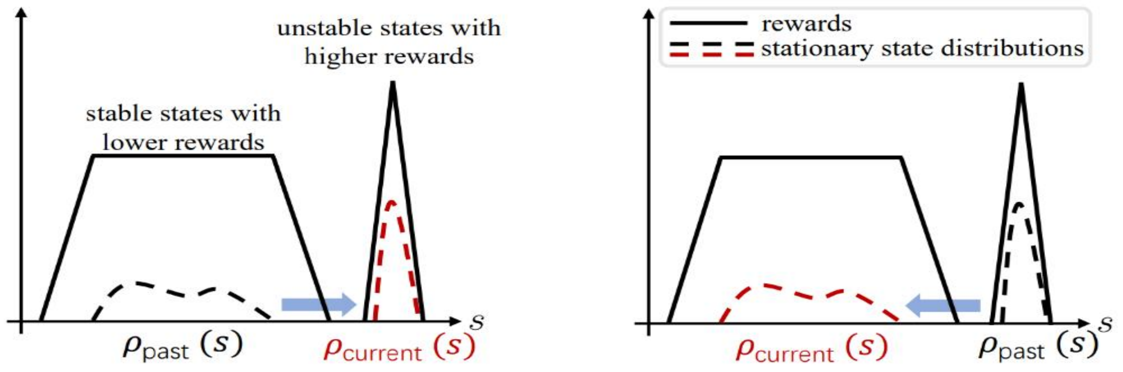
reduces the constrained optimization problem in transition dynamics to a simpler one in state space

State-Conservative Markov Decision Process

State-Conservative Markov Decision Process (SC-MDP):

Objective:

$$J_{\epsilon\text{-}\mathcal{S}}(\pi) \triangleq \mathbb{E}_{s_0\sim d_0}\Big[\inf_{s_0'\in B_\epsilon(s)} \mathbb{E}_{a_0'\sim\pi(\cdot|s_0')}\big[r(s_0',a_0')$$
$$+\gamma\mathbb{E}_{s_1\sim p_0(\cdot|s_0',a_0')}\Big[\inf_{s_1'\in B_\epsilon(s_1)} \mathbb{E}_{a_1'\sim\pi(\cdot|s_1')}\big[r(s_1',a_1')$$
$$+\gamma\mathbb{E}_{s_2\sim p_0(\cdot|s_1',a_1')}\Big[\inf_{s_2'\in B_\epsilon(s_2)} \mathbb{E}_{a_2'\sim\pi(\cdot|s_2')}\big[r(s_2',a_2')+\cdots\big]\cdots\Big]$$



(a) An example MDP. (b) Illustrate the SC-MDP.

An illustration of the state-conservative MDP

Theoretical Analysis

Based on the above definition of SC-MDP, we can define the corresponding Q function, which is exactly the fixed point of $\mathbb{T}_\epsilon^\pi$

$$\mathbb{T}_\epsilon^\pi Q(s,a) = r(s,a) + \gamma \mathbb{E}_{\tilde{s}\sim p_0(\cdot|s,a)}\left[\inf_{s'\in B_\epsilon(\tilde{s})} \mathbb{E}_{a'\sim\pi(\cdot|s')}\left[Q\left(s',a'\right)\right]\right]$$

Thus, the traditional strategy iterative algorithm is still convergent in the conservative state Markov decision-making process（Algorithm1）：

Algorithm 1 State-Conservative Policy Iteration

1: **Input:** A given MDP with transition p_0, an initial policy π^0, a non-negative real ϵ and iteration step K.
2: **for** $k = 0, 1, \cdots, K$ **do**
3: Perform **State-Conservative Policy Evaluation** till convergence to obtain $Q^{\pi^k}_{\epsilon\text{-}\mathcal{S}}$.
4: Perform **Policy Improvement** step by $\pi^{k+1}(\cdot|s) \leftarrow \arg\max_{\pi\in\Delta_\mathcal{A}} \inf_{s'\in B_\epsilon(s)} \mathbb{E}_{a'\sim\pi}[Q^{\pi^k}_{\epsilon\text{-}\mathcal{S}}(s',a')]$.
5: **end for**
6: **Output:** an estimation π^{K+1} of the optimal policy.

We extend the SC-PI algorithm to a model-free actor-critic algorithm: state-conservative policy optimization (SCPO).

Policy Evaluation (Train Critic)

$$J_{\epsilon-Q}(\theta) \triangleq \mathbb{E}_{(s_t,a_t)\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\theta\left(s_t,a_t\right)-\hat{Q}\left(s_t,a_t\right)\right)^2\right]$$

where:

$$\hat{Q}\left(s_t,a_t\right) \triangleq r\left(s_t,a_t\right)+\gamma\mathbb{E}_{s_{t+1}\sim p_0}\left[\hat{V}\left(s_{t+1}\right)\right]$$

$$\hat{V}\left(s_{t+1}\right) \triangleq \inf_{s'\in B_f(s_{t+1})} \mathbb{E}_{a'\sim\pi_\phi(\cdot|s')}\left[Q_\theta\left(s',a'\right)\right]$$

Policy Improvement (Train Actor)

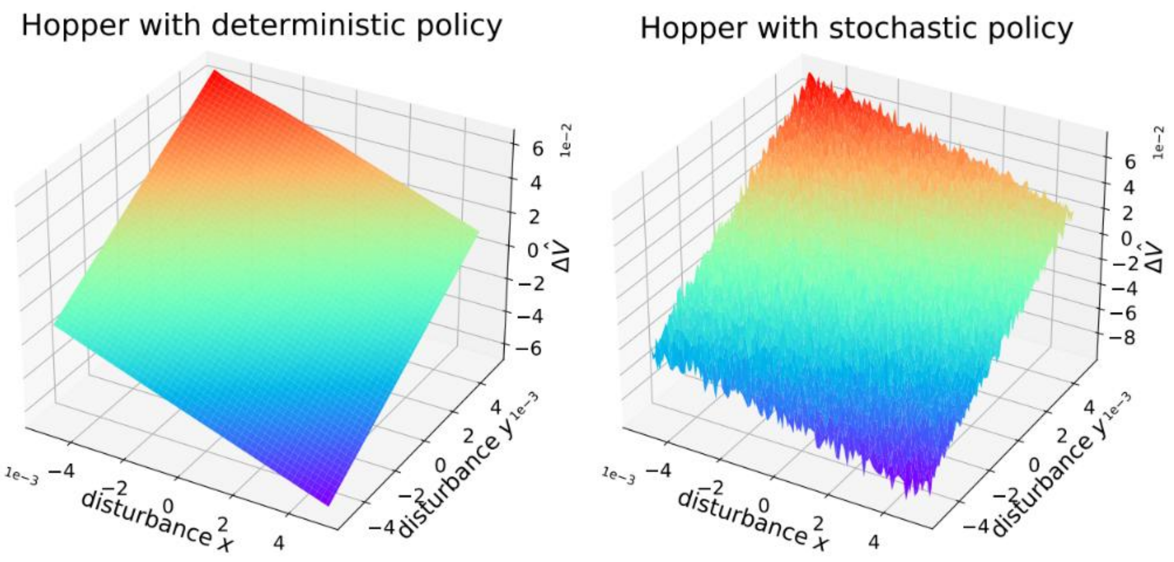
$$J_{\epsilon-\pi}(\phi) \triangleq \mathbb{E}_{S_t\sim D}\left[\inf_{s\in B_\epsilon(s_t)} \mathbb{E}_{a\sim\pi_\phi(\cdot|s)}\left[Q_\theta(s,a)\right]\right].$$

How to solve the following problem

$$\inf_{s'\in B_\epsilon(s)} \mathbb{E}_{a'\sim\pi_\phi(\cdot|s')}\left[Q_\theta\left(s',a'\right)\right]$$

State-Conservative Policy Optimization

We use a gradient based method that approximately solves the constrained optimization problem efficiently.



(a) Deterministic policy. (b) Stochastic policy.

$$\inf_{s' \in B_\epsilon(s)} \bar{U}_{\theta,\phi,s}\left(s'\right) = U_{\theta,\phi}(s) - \epsilon \left\|\nabla_s U_{\theta,\phi}(s)\right\|_1$$

where $U_{\theta,\phi}(s) \triangleq \mathbb{E}_{a \sim \pi_\phi(\cdot|s)}\left[Q_\theta(s,a)\right]$

State-Conservative Policy Optimization

Algorithm 2: State-Conservative Soft Actor-Critic

Input: Critic $Q_{\theta_1}, Q_{\theta_2}$. Actor π_ϕ. Initial temperature parameter α. Step size $\beta_Q, \beta_\pi, \beta_\alpha$. Target smoothing coefficient τ.
$\bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2, \mathcal{D} \leftarrow \emptyset$.
for each iteration **do**
  for each environment step **do**
    $a_t \sim \pi(\cdot|s_t),\ s_{t+1} \sim p_0(\cdot|s_t, a_t)$.
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{s_t, a_t, r(s_t, a_t), s_{t+1}\}$.
  end for
  for each training step **do**
    $\theta_i \leftarrow \theta_i - \beta_Q \nabla J_{\epsilon\text{-}Q}(\theta)$ for $i = 1, 2$.
    $\phi \leftarrow \phi - \beta_\pi \nabla J_{\epsilon\text{-}\pi}(\phi)$.
    $\alpha \leftarrow \alpha - \beta_\alpha \nabla J_\alpha(\alpha)$.
    $\bar{\theta}_i \leftarrow \tau\theta_i + (1-\tau)\bar{\theta}_i$ for $i = 1, 2$.
  end for
end for
Output: θ_1, θ_2, ϕ

State-Conservative Soft Actor-

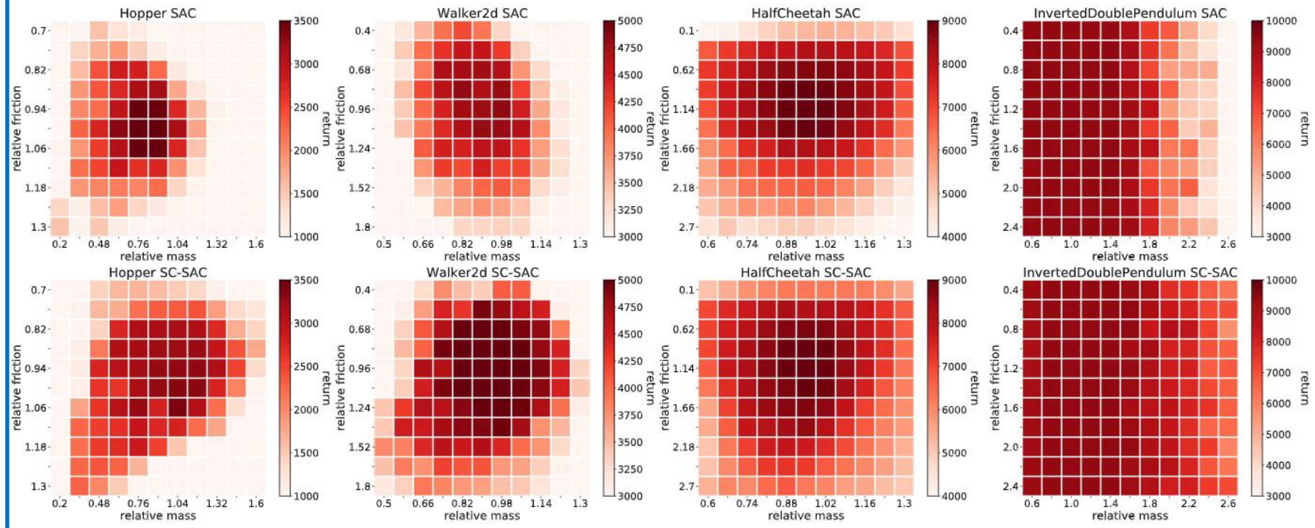
strengths:

- simple to implement
- does not require task-specific prior knowledge to model the disturbance.

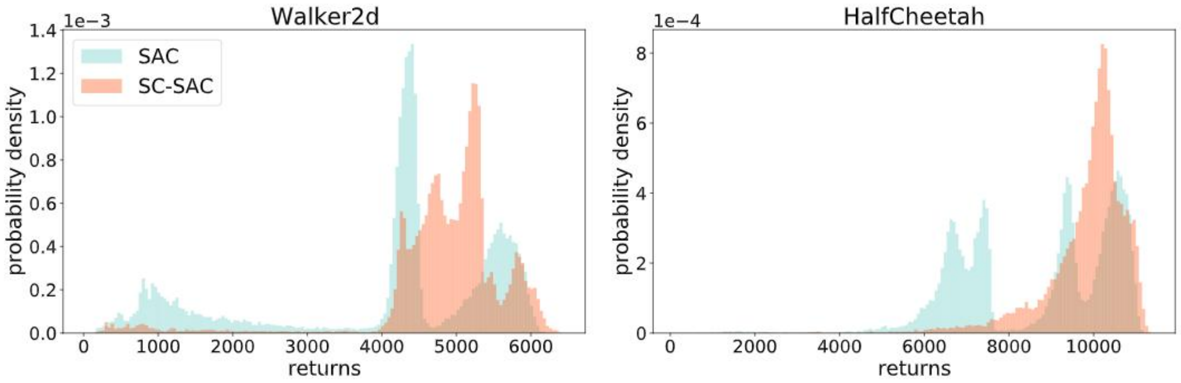
Part 3 Experiments

Experimental Results

Results (1)： comparison with the original SAC algorithm



Compare SAC and SC-SAC in environments with perturbed mass and friction

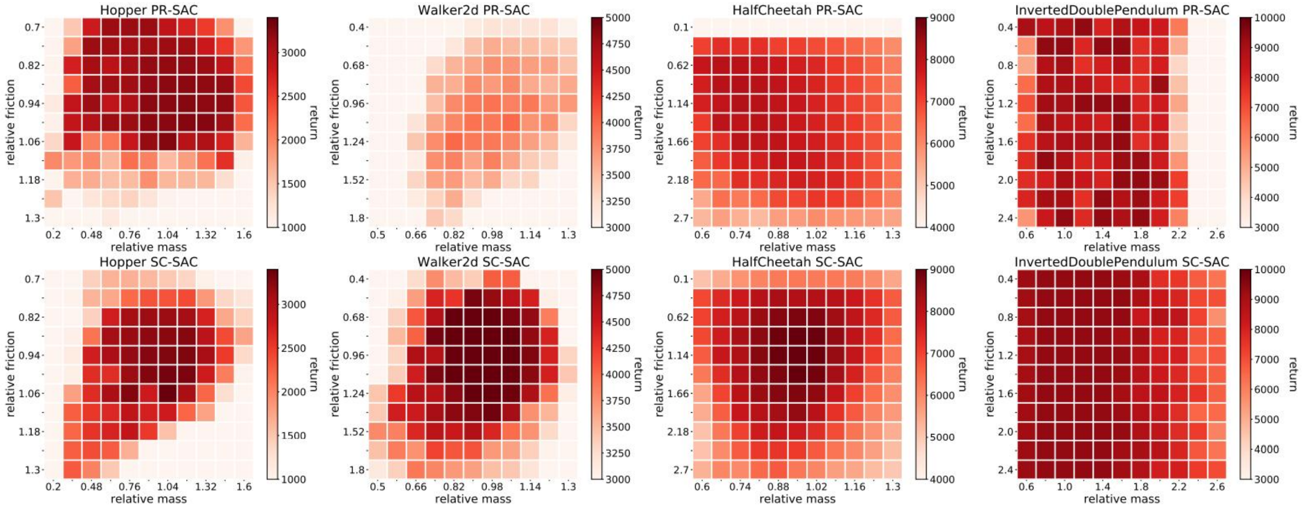


Compare the return distribution between SAC and SC-SAC with truncated Gaussian parameters

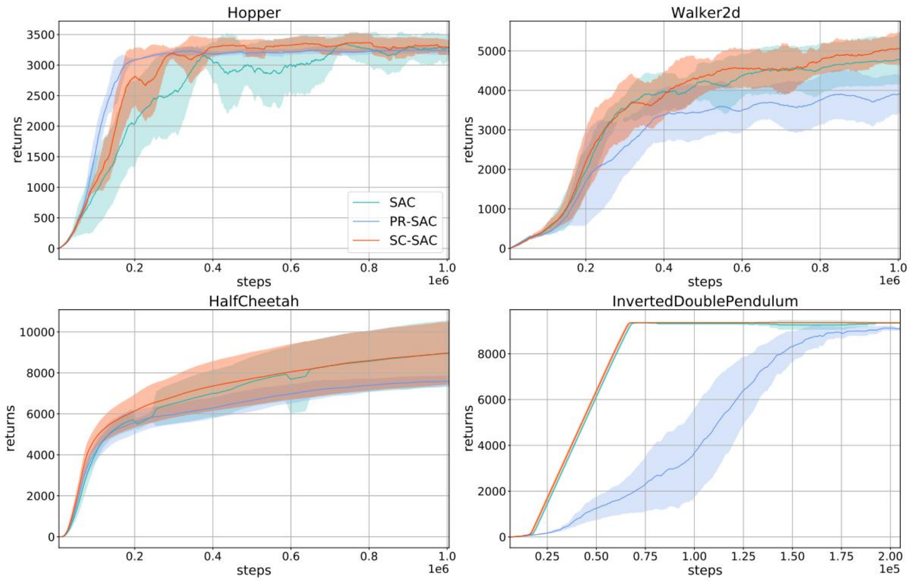
Conclusion: SC-SAC is more robust than SAC when generalize to target environments

Experimental Results

Results (2)： comparison with the action robust algorithm PR-MDP



Compare PR-SAC and SC-SAC in environments with perturbed mass and friction

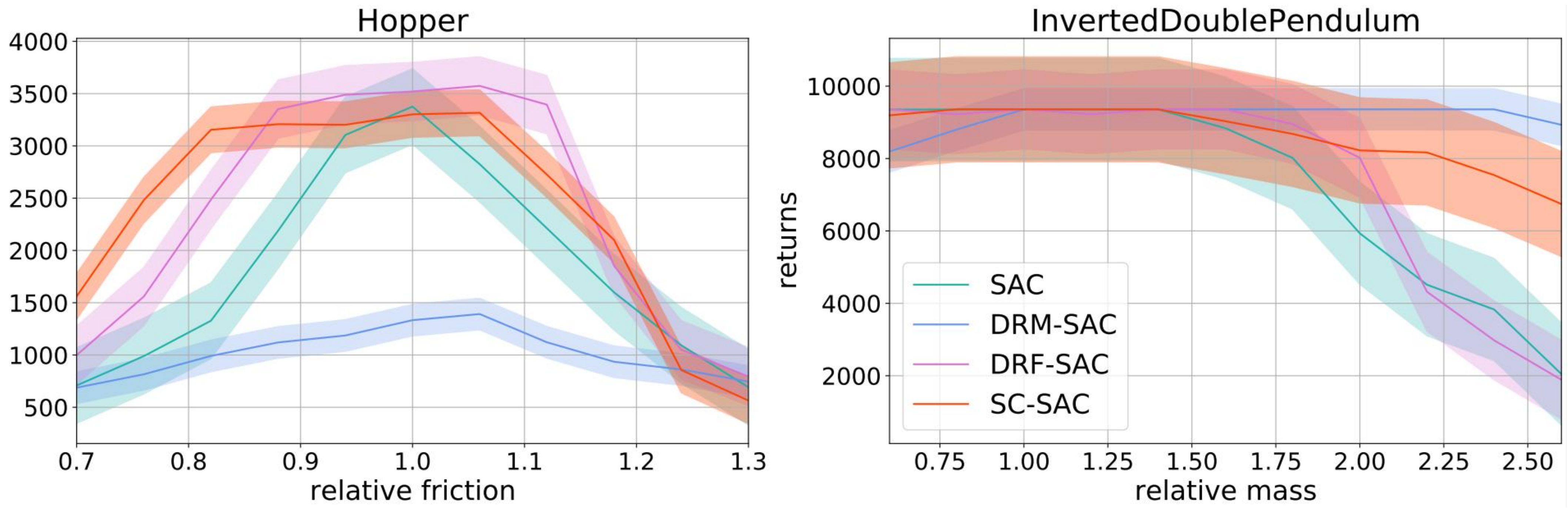


Training curves for different algorithms.

Conclusion: SC-SAC achieves higher average returns than PR-SAC in most tasks

Experimental Results

Results (3)： comparison with domain randomization (DR)



Compare SCPO with DR in target environments with one parameter perturbed.

Conclusion: though the perturbations on both mass and friction are unmodeled during training, SC-SAC trained policies are robust to them consistently

Part 4 Conclusion

Conclusion

- A novel algorithm SCPO to learn robust policies
  - ***simple to implement*** and apply to existing actor-critic algorithms
  - does not require ***task-specific knowledge*** to model the disturbance
  - ***learns robust policies*** against disturbance in practice

See our Homepage for More Details

https://miralab.ai/

https://miralab.ai/people/yufei-kuang/

MANY THANKS !