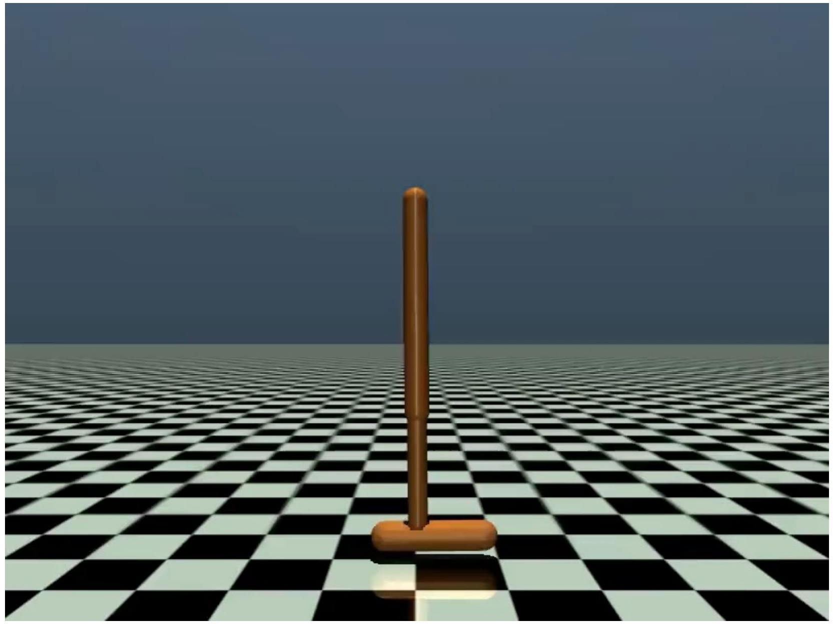
State Regularized Policy Optimization on Data with Dynamics Shift

NeurIPS 2023

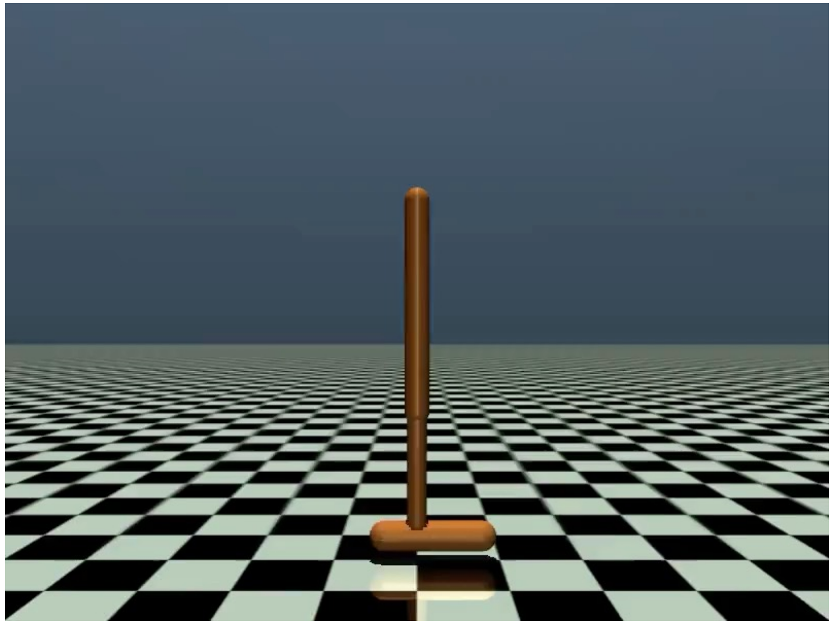
Zhenghai Xue[1], Qingpeng Cai[2], Shuchang Liu[2], Dong Zheng[2], Peng Jiang[2] , Kun Gai[3] , Bo An[1]

[1]Nanyang Technological University, Singapore,
[2]Kuaishou Technology, [3]Unaffliated

Traditional RL algorithms cannot cope with environments with different dynamics.



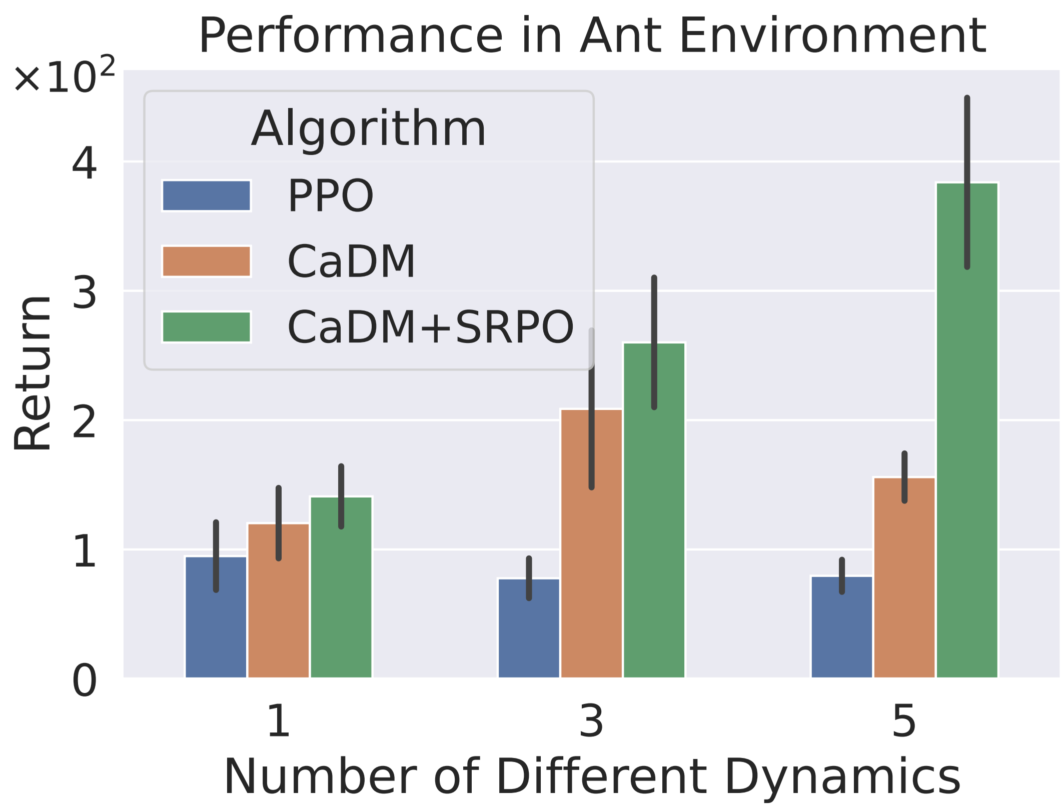
Original Hopper-v2 environment
SAC Policy; Return 3820



Hopper-v2 environment with 0.7x body mass
SAC Policy; Return 2274

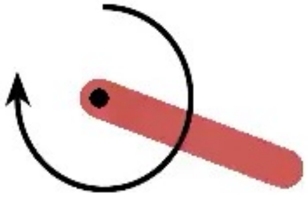
Context-based algorithms use context encoders to detect dynamic changes.

Problems with context-based algorithms: policies can not learn from data with dynamics shift

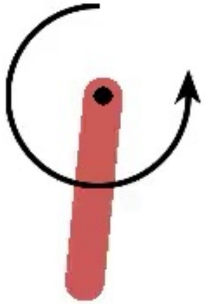


We propose the SRPO algorithm that can leverage data with different dynamics.

The key intuition: optimal policies in environments with *different* dynamics can generate a *similar* stationary state distribution.

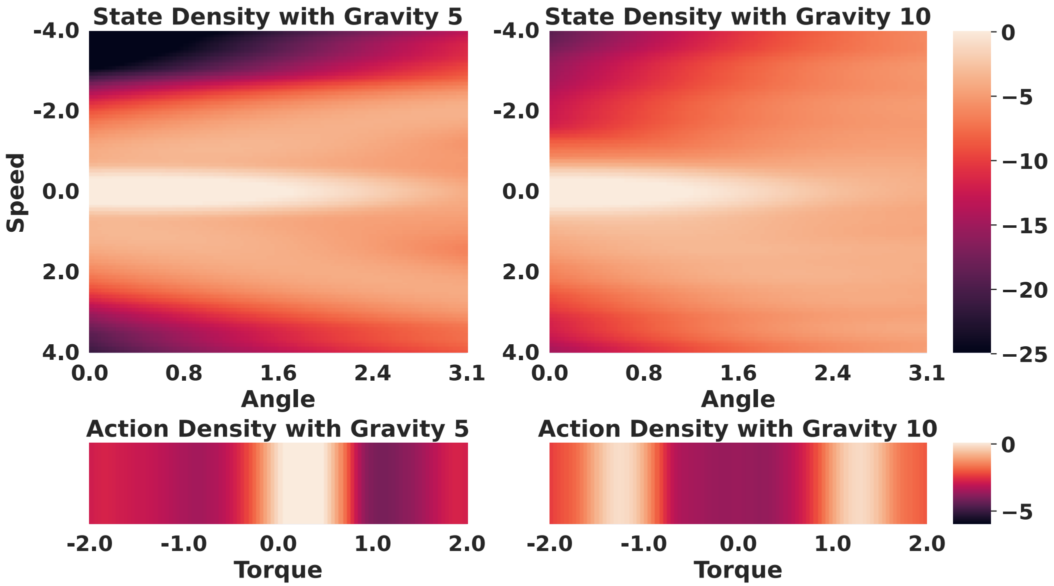


Pendulum-v1 with gravity 5



Pendulum-v1 with gravity 10

State and action density estimated by KDE:



Incorporate into policy optimization:

$$\max_{\pi} \mathbb{E}_{s_t,a_t\sim\tau_\pi}\left[\sum_{t=0}^{\infty}\gamma^t r\left(s_t,a_t\right)\right] \qquad \text{s.t.} \quad D_{\mathrm{KL}}\left(d_\pi(\cdot)\|\zeta(\cdot)\right) < \varepsilon$$

d_π : the stationary state distribution of the *current* policy
ζ : the stationary state distribution of the *optimal* policy

Lagrangian: $$L = -\mathbb{E}_{s_t,a_t\sim\tau}\left[\sum_{t=0}^{\infty}\gamma^t\left(r(s_t,a_t)+\lambda\log\frac{\zeta(s_t)}{d_\pi(s_t)}\right)\right]-\frac{\lambda\varepsilon}{1-\gamma}$$

Challenges: How to obtain the state probability under the optimal policy?
How to compute the probability ratio?

Lagrangian: $$L = -\mathbb{E}_{s_t,a_t\sim\tau}\left[\sum_{t=0}^{\infty}\gamma^t\left(r(s_t,a_t)+\lambda\log\frac{\zeta(s_t)}{d_\pi(s_t)}\right)\right]-\frac{\lambda\varepsilon}{1-\gamma}$$

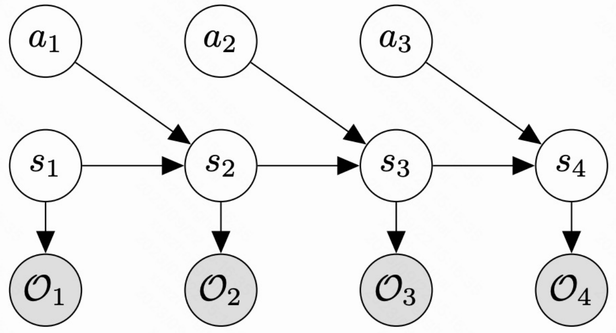
Challenges: How to obtain the state probability under the optimal policy?
How to compute the probability ratio?

Proposition 3.1. *In a GAN, when the real data distribution is $\zeta(s)$ and the generated data distribution is $d_\pi(s)$, the output of the discriminator $D(s)$ follows*

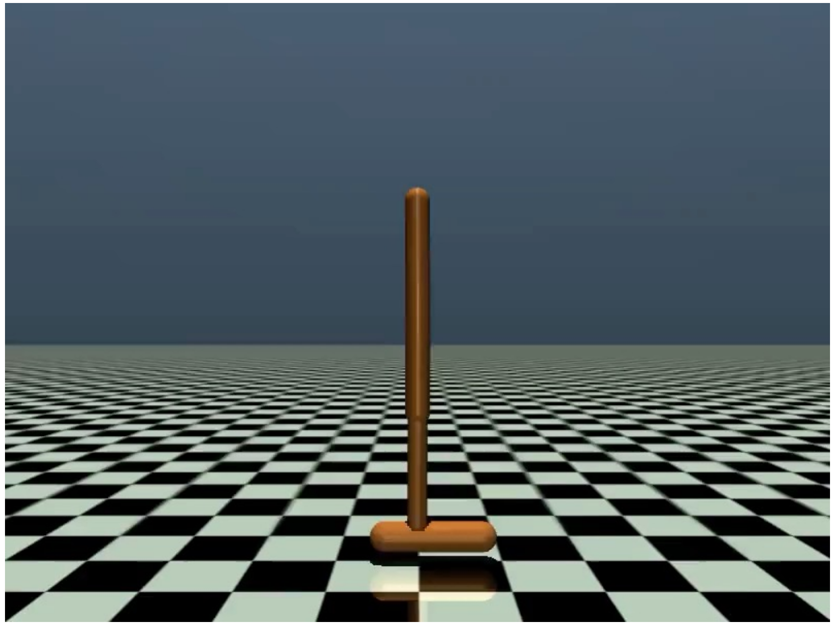
$$\frac{D(s)}{1-D(s)}=\frac{\zeta(s)}{d_\pi(s)}. \tag{4}$$

Measure the state optimality: an HMM-based approach

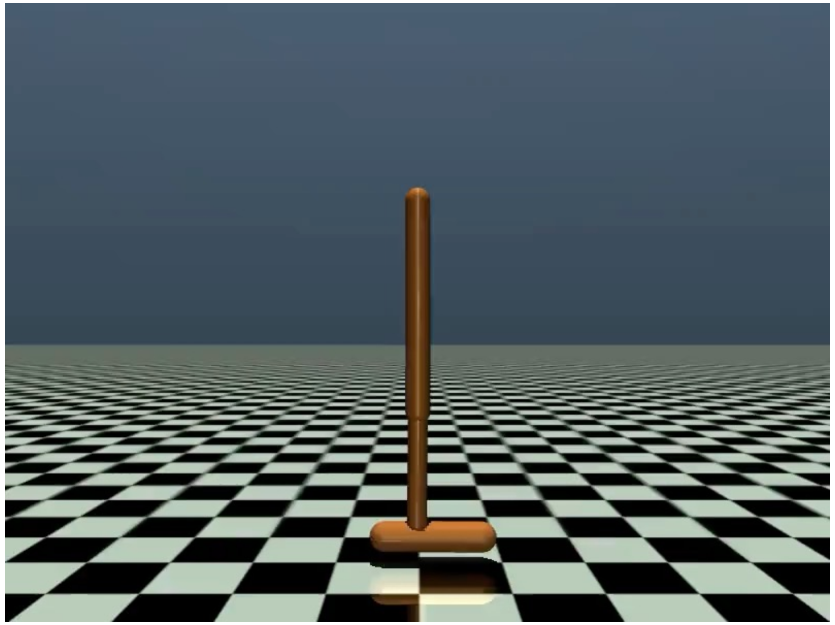
$$p(\mathcal{O}_t|s_t)=\max_{a_t}\exp[\gamma^t(r(s_t,a_t)-R_{\max})]$$



In Online RL tasks, we propose the CaDM+SRPO algorithm that can efficiently train a robust policy.

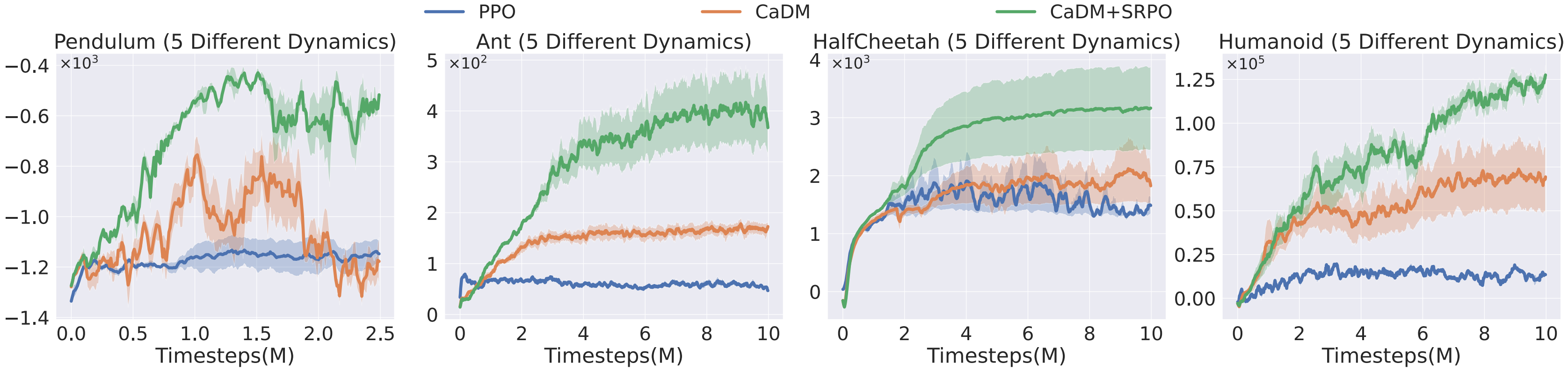


Original Hopper-v2 environment
Return 3167



Hopper-v2 environment with 10x medium density
Return 3628

Comparative results:



In Offline RL tasks, we propose the MAPLE+SRPO algorithm that reaches the highest performance in 8 of 12 tasks.

	CQL Single	CQL	MOPO	MAPLE	MAPLE +DARA	MAPLE +SRPO(Ours)
Walker2d-medium-expert	**1.11**	1.03±0.10	0.25±0.18	0.55±0.21	0.80±0.02	0.66±0.08
Walker2d-medium	0.79	0.78±0.01	0.23±0.34	0.82±0.01	0.83±0.03	**0.84**±0.03
Walker2d-medium-replay	**0.27**	0.07±0.00	0.00±0.00	0.16±0.02	0.17±0.01	0.17±0.02
Walker2d-random	0.07	0.03±0.01	0.00±0.00	**0.22**±0.00	**0.22**±0.00	**0.22±0.00**
Hopper-medium-expert	**0.98**	0.32±0.14	0.01±0.00	0.96±0.14	0.96±0.06	**0.98**±0.02
Hopper-medium	0.58	0.57±0.16	0.01±0.00	0.78±0.28	0.40±0.05	**1.03**±0.09
Hopper-medium-replay	0.46	0.14±0.02	0.01±0.01	0.91±0.11	**1.02**±0.01	**1.02**±0.01
Hopper-random	0.11	0.11±0.00	0.01±0.00	0.13±0.00	0.13±0.01	**0.32**±0.02
HalfCheetah-medium-expert	0.62	0.03±0.04	-0.03±0.00	0.50±0.06	0.50±0.00	**0.63**±0.01
HalfCheetah-medium	0.44	0.43±0.03	0.38±0.28	0.62±0.01	**0.67**±0.03	0.63±0.01
HalfCheetah-medium-replay	0.46	0.46±0.00	-0.03±0.00	0.52±0.00	0.53±0.01	**0.55**±0.00
HalfCheetah-random	**0.35**	0.01±0.02	-0.03±0.00	0.22±0.03	0.21±0.00	0.24±0.01
Average	0.52	0.33	0.068	0.53	0.54	**0.61**

Code:

https://github.com/AIDefender/SRPO

Poster Page:

https://neurips.cc/virtual/2023/poster/72138