Efficient Wasserstein Natural Gradients for Reinforcement Learning

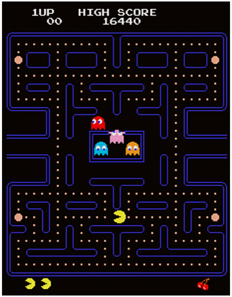
Ted Moskovitz*, Michael Arbel*, Ferenc Huszár, Arthur Gretton

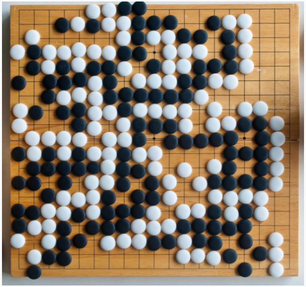
ICLR 2021

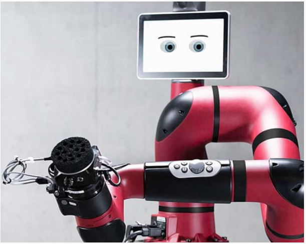
Gatsby Computational Neuroscience Unit

UNIVERSITY OF CAMBRIDGE

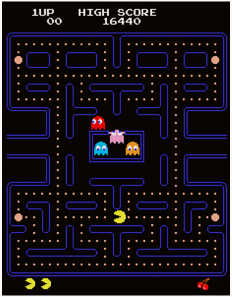
Motivation

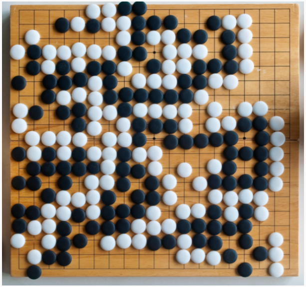


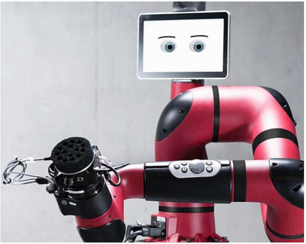




Motivation

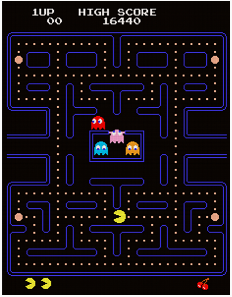


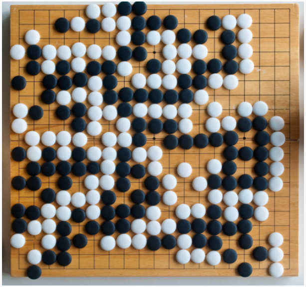


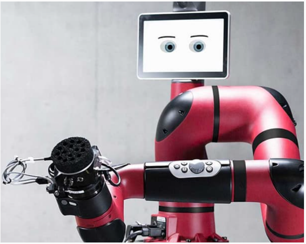


- Deep RL algorithms require $\sim 10^6$-10^9 samples for optimization

Motivation

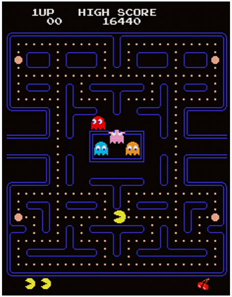


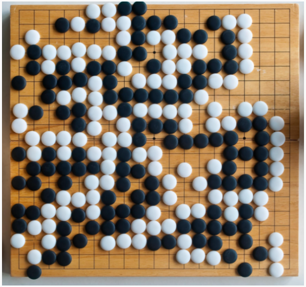


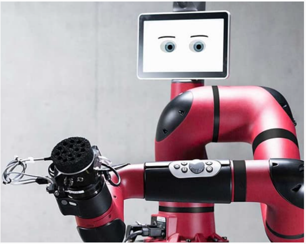


- Deep RL algorithms require $\sim 10^6$-10^9 samples for optimization
- Regularized policy optimization can improve sample efficiency

Motivation







- Deep RL algorithms require $\sim 10^6$-10^9 samples for optimization
- Regularized policy optimization can improve sample efficiency
- The method of regularization can induce a geometry on the loss surface that is often underexploited

Contributions

Contributions

- Introduce a local similarity measure between behavioral distributions based on the Wasserstein Information Matrix (WIM)

Contributions

- Introduce a local similarity measure between behavioral distributions based on the Wasserstein Information Matrix (WIM)
- Use a low rank approximation of the WIM to derive an efficient Wasserstein natural gradient (WNG) for RL

Contributions

- Introduce a local similarity measure between behavioral distributions based on the Wasserstein Information Matrix (WIM)
- Use a low rank approximation of the WIM to derive an efficient Wasserstein natural gradient (WNG) for RL
- Advantage over KL-based methods on problems with deterministic solutions.

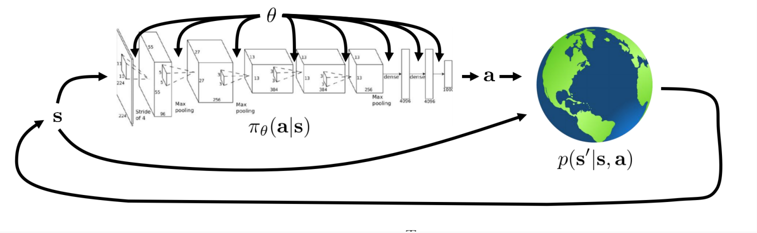
Contributions

- Introduce a local similarity measure between behavioral distributions based on the Wasserstein Information Matrix (WIM)
- Use a low rank approximation of the WIM to derive an efficient Wasserstein natural gradient (WNG) for RL
- Advantage over KL-based methods on problems with deterministic solutions.
- WNG + policy gradients $\rightarrow$ WNPG
  WNG + evolution strategies $\rightarrow$ WNES [1]

Contributions

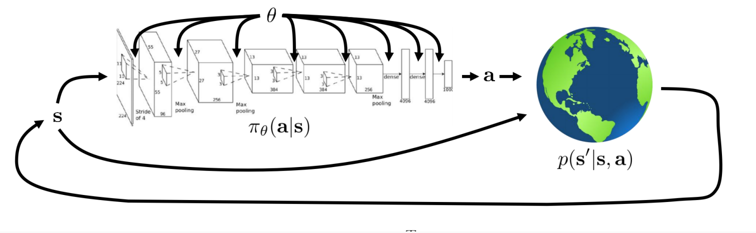
- Introduce a local similarity measure between behavioral distributions based on the Wasserstein Information Matrix (WIM)
- Use a low rank approximation of the WIM to derive an efficient Wasserstein natural gradient (WNG) for RL
- Advantage over KL-based methods on problems with deterministic solutions.
- WNG + policy gradients $\rightarrow$ WNPG
  WNG + evolution strategies $\rightarrow$ WNES [1]
- Improved results on challenging continuous control tasks.

RL Background



* figure adapted from http://rail.eecs.berkeley.edu/deeprlcourse/

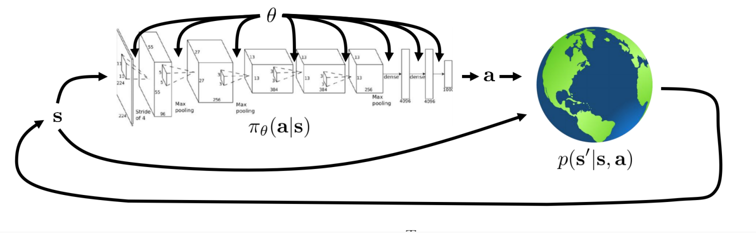
RL Background



- Assume an agent is acting in an MDP $(\mathcal{S}, \mathcal{A}, r, p, \gamma)$

* figure adapted from http://rail.eecs.berkeley.edu/deeprlcourse/

RL Background

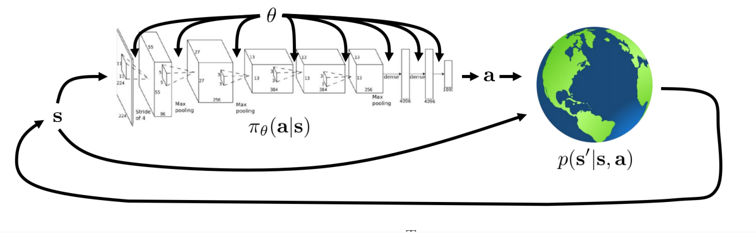


- Assume an agent is acting in an MDP $(\mathcal{S}, \mathcal{A}, r, p, \gamma)$

- Running the policy in an episodic/finite horizon task of length T produces trajectories $\tau = (s_1, a_1, r_1, \ldots, s_T, a_T, r_T)$

* figure adapted from http://rail.eecs.berkeley.edu/deeprlcourse/

RL Background



- Assume an agent is acting in an MDP $(\mathcal{S}, \mathcal{A}, r, p, \gamma)$
- Running the policy in an episodic/finite horizon task of length T produces trajectories $\tau = (s_1, a_1, r_1, \ldots, s_T, a_T, r_T)$
- Maximize $\mathbb{E}_\pi [Z(\tau)] = \mathbb{E}_\pi \left[\sum_t \gamma^t r_t\right]$

* figure adapted from http://rail.eecs.berkeley.edu/deeprlcourse/

Regularized Policy Optimization

- Consider policy gradients on some parametric policy $\pi_\theta(a_t|s_t)$:

$$F(\theta) = \mathbb{E}_\pi \left[\sum_t \gamma^t r_t \right] \implies \nabla_\theta F = \mathbb{E}_\pi \left[Z(\tau) \sum_t \nabla_\theta \log \pi_\theta(a_t|s_t) \right]$$

Regularized Policy Optimization

- Consider policy gradients on some parametric policy $\pi_\theta(a_t|s_t)$:

$$F(\theta) = \mathbb{E}_\pi\left[\sum_t \gamma^t r_t\right] \quad \implies \quad \nabla_\theta F = \mathbb{E}_\pi\left[Z(\tau)\sum_t \nabla_\theta \log \pi_\theta(a_t|s_t)\right]$$

- Problem: *expensive* to sample new trajectories, but re-using for multiple updates degrades optimization...

Regularized Policy Optimization

- Consider policy gradients on some parametric policy $\pi_\theta(a_t|s_t)$:

$$F(\theta) = \mathbb{E}_\pi \left[\sum_t \gamma^t r_t \right] \quad \Longrightarrow \quad \nabla_\theta F = \mathbb{E}_\pi \left[Z(\tau) \sum_t \nabla_\theta \log \pi_\theta(a_t|s_t) \right]$$

- Problem: *expensive* to sample new trajectories, but re-using for multiple updates degrades optimization...

- Solution: Only take large steps that change *behavior* the least:

$$\text{maximize}_\theta \; F(\theta) - \beta D(\pi_{\theta_k}(\cdot|s_t) || \pi_\theta(\cdot|s_t))$$

Regularized Policy Optimization

- Consider policy gradients on some parametric policy $\pi_\theta(a_t|s_t)$:

$$F(\theta) = \mathbb{E}_\pi\left[\sum_t \gamma^t r_t\right] \quad \Longrightarrow \quad \nabla_\theta F = \mathbb{E}_\pi\left[Z(\tau)\sum_t \nabla_\theta \log \pi_\theta(a_t|s_t)\right]$$

- Problem: *expensive* to sample new trajectories, but re-using for multiple updates degrades optimization...

- Solution: Only take large steps that change *behavior* the least:

$$\text{maximize}_\theta \; F(\theta) - \beta D(\pi_{\theta_k}(\cdot|s_t)||\pi_\theta(\cdot|s_t))$$

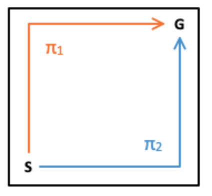
- $\Rightarrow$ We can make multiple updates with the same trajectories

Behavioral Geometry

- Local action distributions don't always reflect global *behavior*:

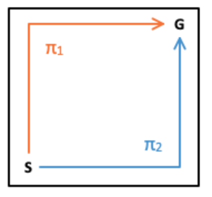
Behavioral Geometry

- Local action distributions don't always reflect global *behavior*:

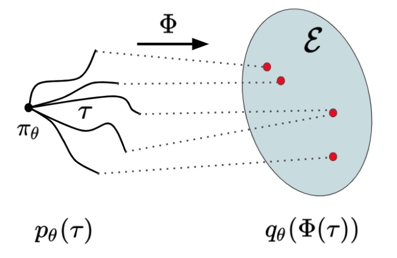


Behavioral Geometry

- Local action distributions don't always reflect global *behavior*:

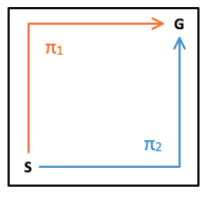


- How can we capture behavioral similarity? *Embed* trajectories [2]

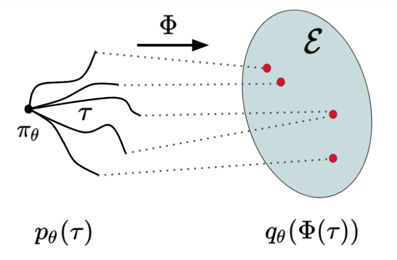


Behavioral Geometry

- Local action distributions don't always reflect global *behavior*:



- How can we capture behavioral similarity? *Embed* trajectories [2]



- How to compare? Measure WD between embedding distributions

Divergences Measures $\Rightarrow$ Natural Gradients

Divergences Measures $\Rightarrow$ Natural Gradients

- Euclidean gradient: Steepest direction wrt the L_2 distance

$$g^E = \underset{u}{\text{argmax}}\, F(\theta) + \nabla F^\top u - \frac{1}{2}\|u\|^2 \approx \underset{u}{\text{argmax}}\, F(\theta + u) - \frac{1}{2}\|u\|^2$$

Divergences Measures ⇒ Natural Gradients

- Euclidean gradient: Steepest direction wrt the L_2 distance

$$g^E = \underset{u}{\operatorname{argmax}} F(\theta) + \nabla F^\top u - \frac{1}{2}\|u\|^2 \approx \underset{u}{\operatorname{argmax}} F(\theta + u) - \frac{1}{2}\|u\|^2$$

- Fisher natural gradient: Steepest direction wrt the KL

$$g^F = \underset{u}{\operatorname{argmax}} F(\theta) + \nabla F^\top u - \frac{1}{2} u^\top G_F u \approx \underset{u}{\operatorname{argmax}} F(\theta + u) - \frac{1}{2}\mathrm{KL}[\pi_{\theta+u} \| \pi_\theta]$$

Divergences Measures $\Rightarrow$ Natural Gradients

- Euclidean gradient: Steepest direction wrt the L_2 distance

$$g^E = \operatorname*{argmax}_u F(\theta) + \nabla F^\mathsf{T} u - \frac{1}{2}\|u\|^2 \approx \operatorname*{argmax}_u F(\theta + u) - \frac{1}{2}\|u\|^2$$

- Fisher natural gradient: Steepest direction wrt the KL

$$g^F = \operatorname*{argmax}_u F(\theta) + \nabla F^\mathsf{T} u - \frac{1}{2} u^\mathsf{T} G_F u \approx \operatorname*{argmax}_u F(\theta + u) - \frac{1}{2}\mathrm{KL}[\pi_{\theta+u} \| \pi_\theta]$$

- Wasserstein natural gradient: Steepest direction wrt W_2

$$g^W = \operatorname*{argmax}_u F(\theta) + \nabla F^\mathsf{T} u - \frac{1}{2} u^\mathsf{T} G_W u \approx \operatorname*{argmax}_u F(\theta + u) - \frac{1}{2} W_2(\pi_{\theta+u}, \pi_\theta)$$

$$\Rightarrow g^W = {G_W}^{-1} \nabla F$$

Policy Optimization Using Behavioral Geometry

- The WD penalty captures a policy's global behavioral geometry [1]

$$\underset{\theta}{\operatorname{argmax}} F(\theta + u) + \frac{1}{2}\beta W_2^2(q_{\theta+u}, q_\theta) \tag{0.1}$$

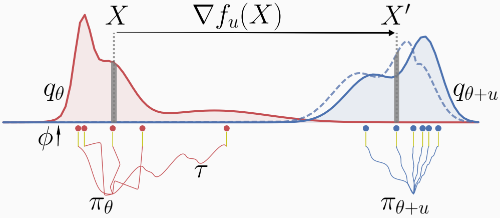
Policy Optimization Using Behavioral Geometry

- The WD penalty captures a policy's global behavioral geometry [1]

$$\underset{\theta}{\operatorname{argmax}} F(\theta + u) + \frac{1}{2}\beta W_2^2(q_{\theta+u}, q_\theta) \tag{0.1}$$

- The WNG captures the local behavior of a policy:

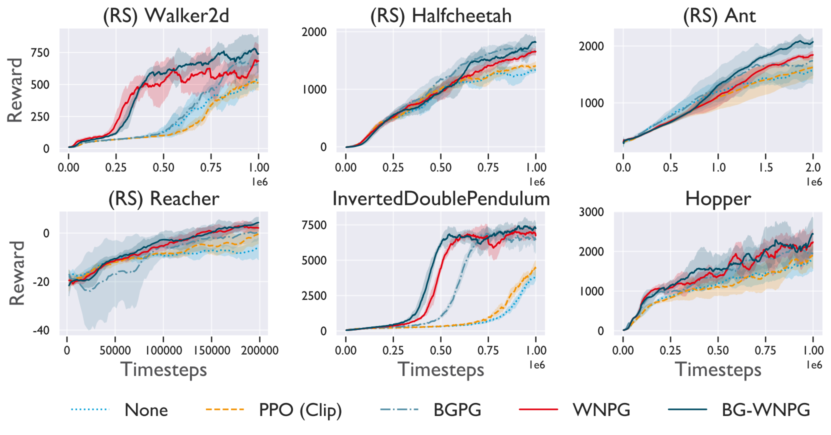
$$W_2^2(q_{\theta+u}, q_\theta) \simeq \mathbb{E}_{q_\theta}[\|\nabla f_u(X)\|^2] = u^\top G_W u \Rightarrow g^W = G_W^{-1}\nabla F \tag{0.2}$$



Wasserstein Natural Policy Gradients

- Apply behavioral WNG to policy gradients (PG):
  (1) $F(\theta)$ (WNPG) (2) $F(\theta) - \beta W_2(q_\theta, q_{\theta+u})$ (BG-WNPG)

- Apply behavioral WNG to policy gradients (PG):
  (1) $F(\theta)$ (WNPG) (2) $F(\theta) - \beta W_2(q_\theta, q_{\theta+u})$ (BG-WNPG)

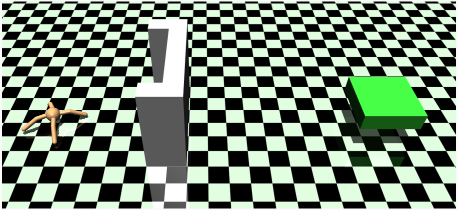


Wasserstein Natural Evolution Strategies

- Apply behavioral WNG to evolution strategies (ES):
  (1) $F(\theta)$ (WNES)   (2) $F(\theta)+\beta W_2(q_\theta, q_{\theta+u})$ (BG-WNES)

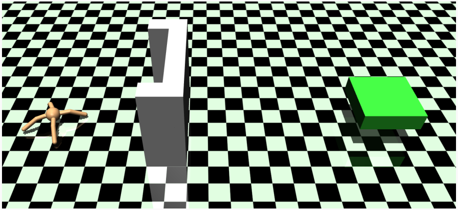
Wasserstein Natural Evolution Strategies

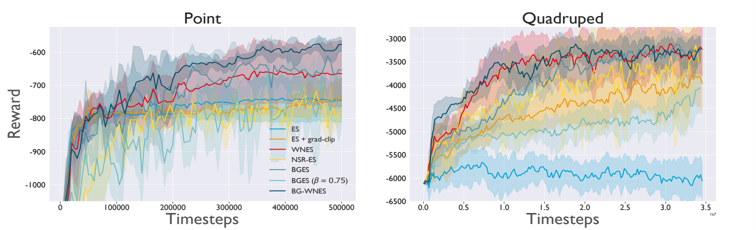
- Apply behavioral WNG to evolution strategies (ES):
  (1) $F(\theta)$ (WNES) (2) $F(\theta) + \beta W_2(q_\theta, q_{\theta+u})$ (BG-WNES)



Wasserstein Natural Evolution Strategies

- Apply behavioral WNG to evolution strategies (ES):
  (1) $F(\theta)$ (WNES) (2) $F(\theta)+\beta W_2(q_\theta, q_{\theta+u})$ (BG-WNES)





(a) (b)

[1] M. Arbel, A. Gretton, W. Li, and G. Montufar. Kernelized wasserstein natural gradient. In *International Conference on Learning Representations*, 2020.

[2] A. Pacchiano, J. Parker-Holder, Y. Tang, A. Choromanska, K. Choromanski, and M. I. Jordan. Learning to score behaviors for guided policy optimization. *arXiv preprint arXiv:1906.04349*, 2019.