Efficient Wasserstein Natural Gradients for Reinforcement Learning

Ted Moskovitz[1], Michael Arbel[1], Ferenc Huszar[1,2], and Arthur Gretton[1]

[1]Gatsby Computational Neuroscience Unit, University College London
[2]University of Cambridge

Overview

Problem

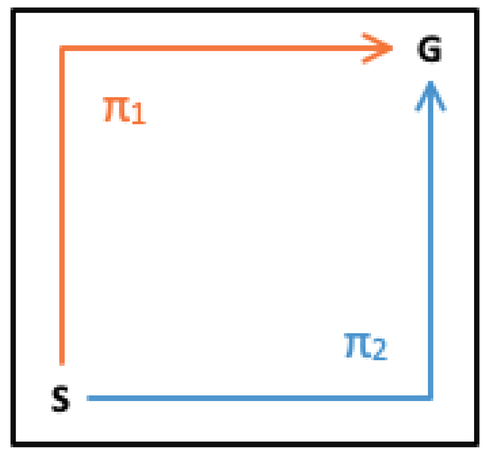
- ✓ Regularized policy optimization is at the heart of many SOTA algorithms for on-policy continuous control. The choice of penalty induces a geometry on the loss surface which is often under-exploited.
- ✓ Example: $\text{argmax}_\theta \mathbb{E}_\pi[\sum_t r_t] - \beta D_{KL}(\pi_{\theta_k}(\cdot|s)||\pi_\theta(\cdot|s)) \to$ approx. FNG
- ✓ Goal: take advantage of the geometry induced by regularizing with the WD

Contributions

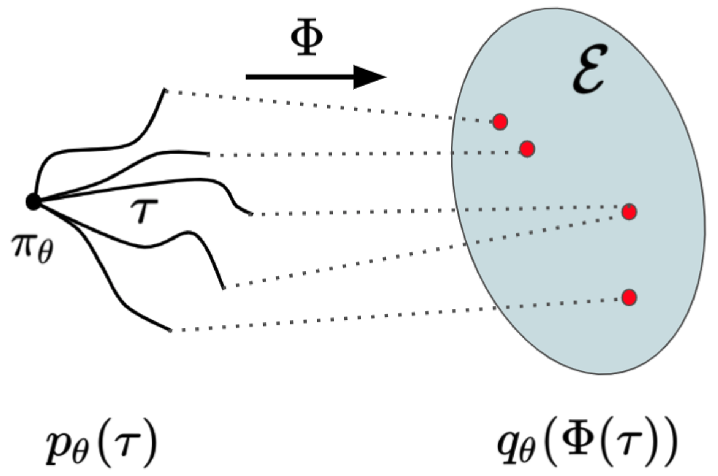
- ✓ Use WIM to define a *local* similarity measure between behavioral distributions
- ✓ WIM $\to$ WNG on behavioral distributions, even w/o re-param. trick
- ✓ Introduce *Wasserstein Natural* PG and ES (WNPG and WNES)
- ✓ Show WNG > FNG on problems with deterministic solutions

Behavioral Geometry

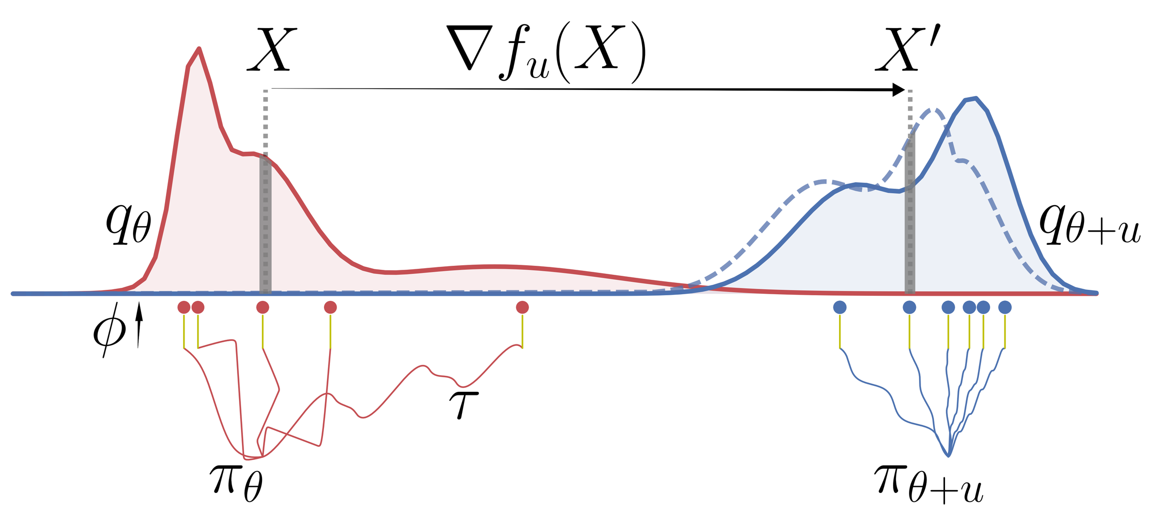
- Local action distributions don't always reflect global *behavior*:



- How can we capture behavioral similarity? *Embed* trajectories and compare [1]:



- How to compare? Measure WD between embedding distributions
- $\text{WD}_2(q_\theta, q_{\theta+u})$: average cost of transporting samples from q_θ to $q_{\theta+u}$ using $\nabla_x f_u(X)$

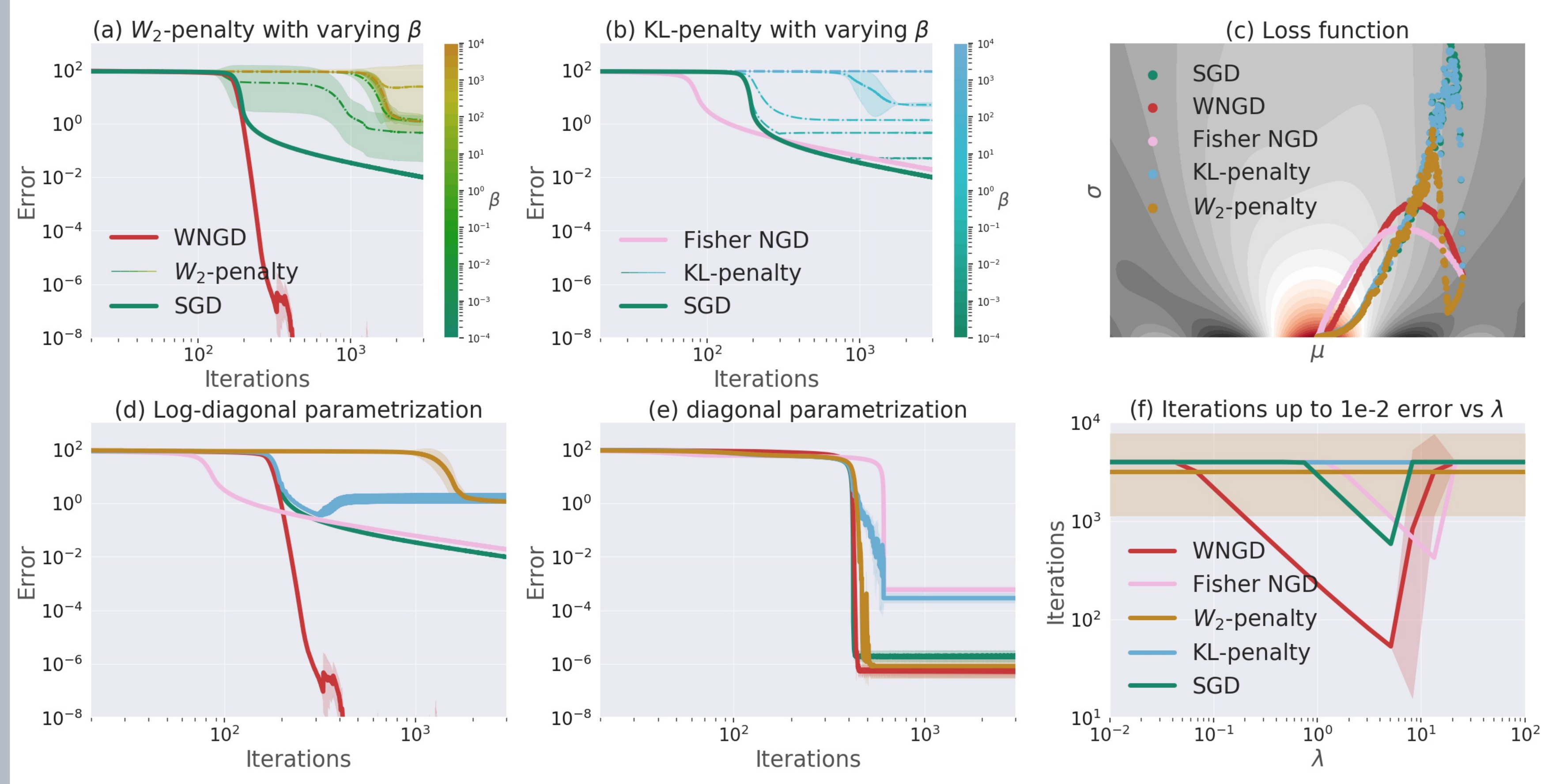


- Optimality of $\nabla_x f_u$ is defined via

$$\sup_{f_u} \underbrace{\nabla_\theta \mathbb{E}_{q_\theta}[f_u(X)]^\top u}_{\text{accurate alignment}} - \underbrace{\frac{1}{2}\mathbb{E}_{q_\theta}\left[\|\nabla_x f_u(X)\|^2\right]}_{\text{transport cost}}$$

Behavioral Geometry via the Wasserstein Natural Gradient

- When the optimal solution is deterministic, WNG outperforms FNG:

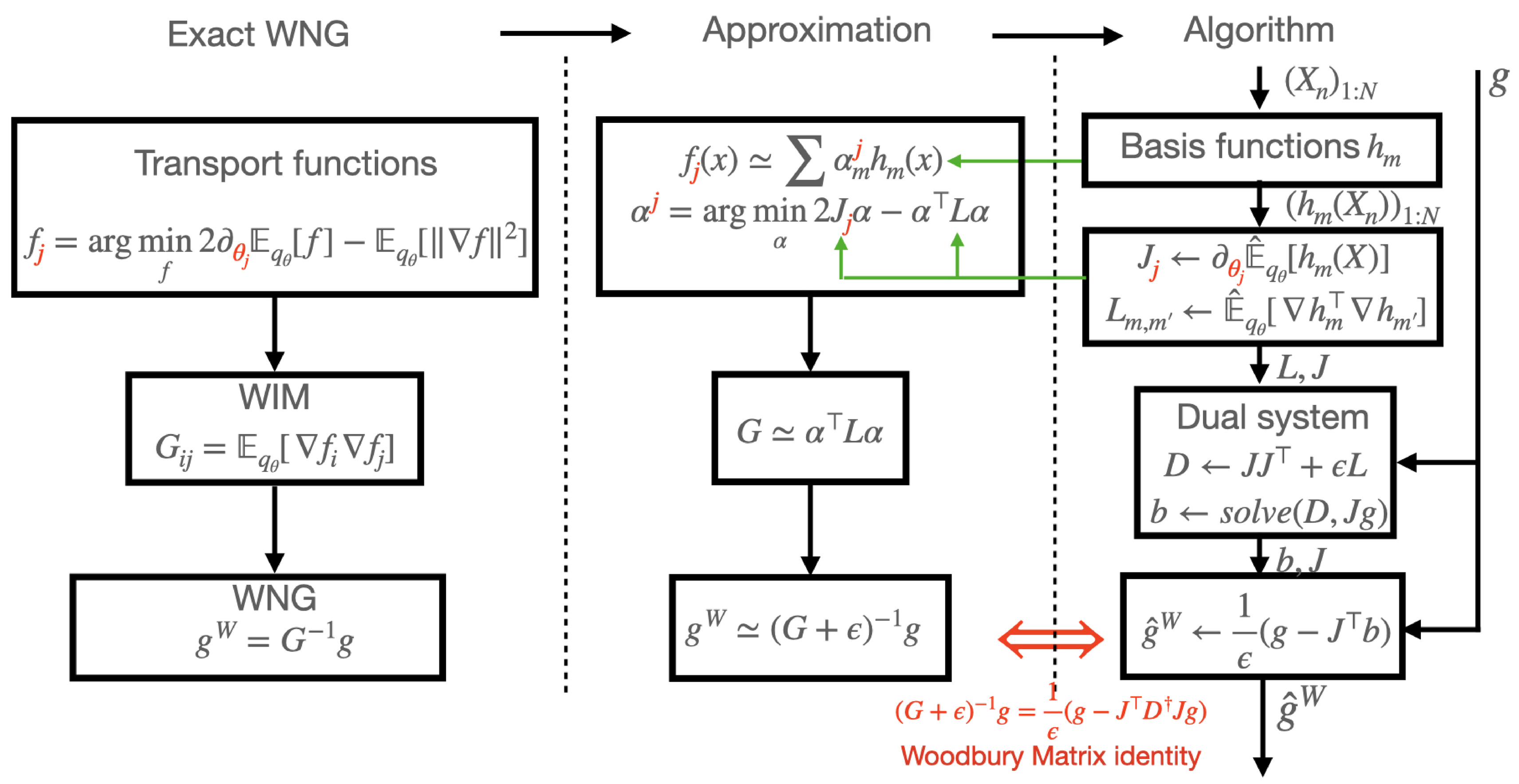


Policy Optimization using Behavioral Geometry

- Compute the behavioral embedding X:
  - Re-parameterization $B_\theta(Z)$ available:

$$X = \Phi(\tau) = [a_0, \dots, a_T] = B_\theta(Z), \qquad Z = \{[s_0, \dots, s_T], \ \epsilon \sim \mathcal{N}(0, \sigma^2 I)\}$$

  - Score $\nabla \log q_\theta(X)$ available: ex. $X = \Phi(\tau) = \sum_t r_t$



- Apply g^W instead of standard gradient g for PG (WNPG) or ES (WNES).
  - WNPG: Jacobian J computed using the *score trick* or the *re-parameterization trick*:

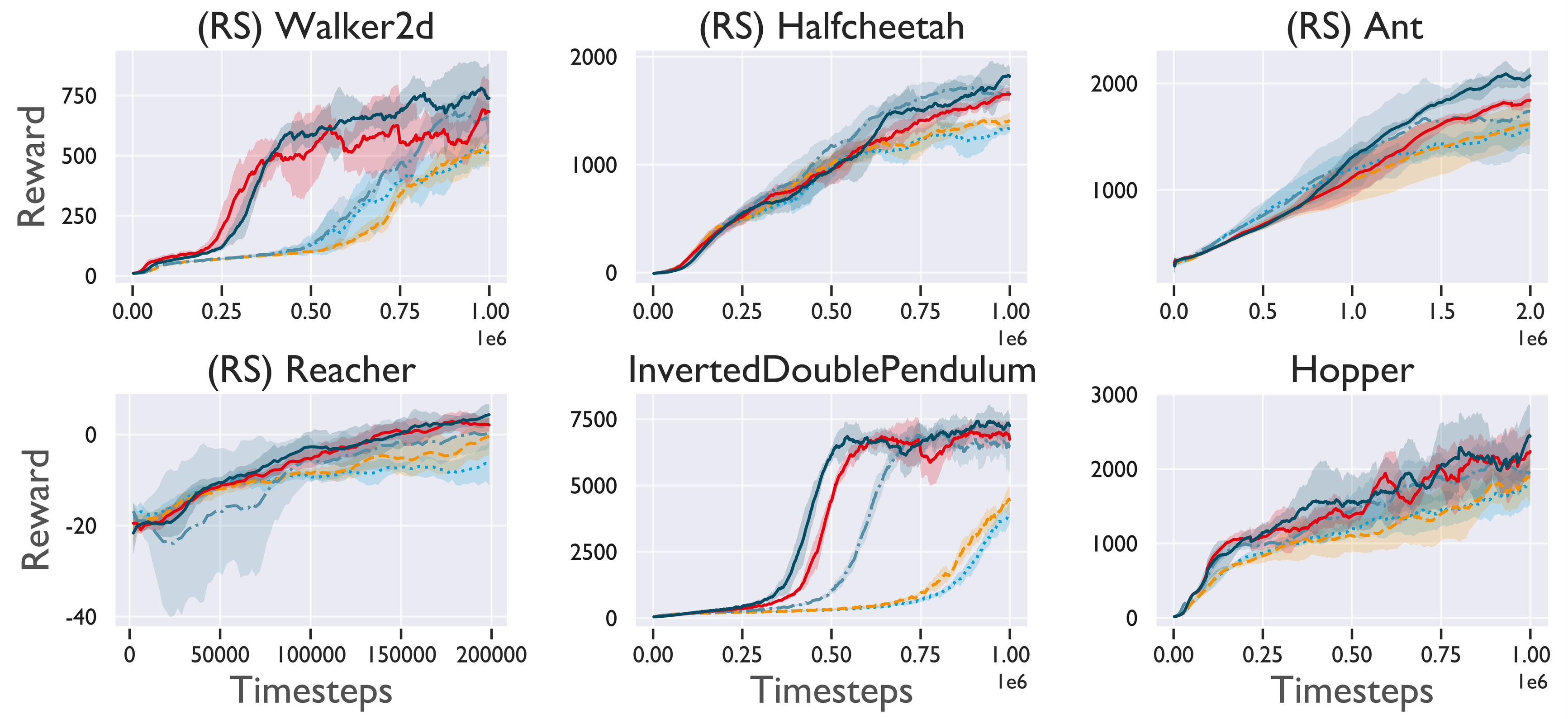
$$J_{m,\cdot} = \underbrace{\hat{\mathbb{E}}_{q_\theta}[\nabla_x h_m(X) \nabla_\theta B_\theta(Z)]}_{\text{reparam. available}} \quad \text{or} \quad J_{m,\cdot} == \underbrace{\hat{\mathbb{E}}_{q_\theta}[\nabla_\theta \log q_\theta(X) h_m(X)]}_{\text{score available}}$$

  - WNES: Jacobian J computed using:

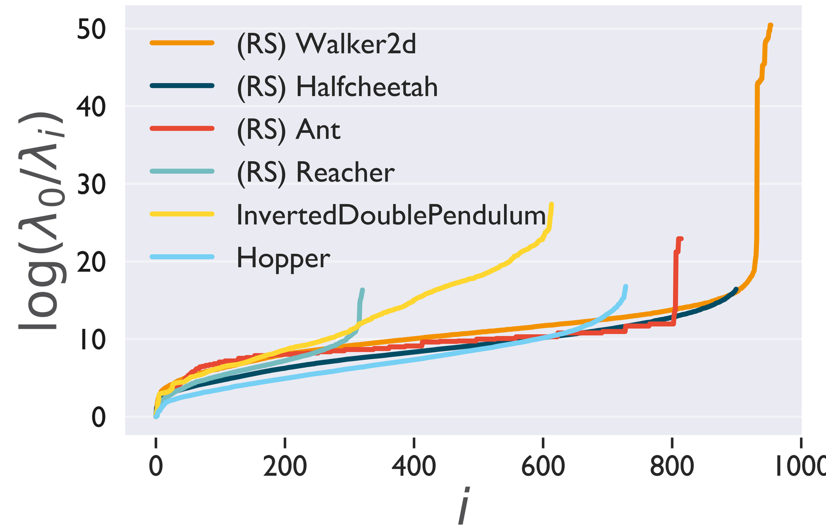
$$J_{m,\cdot} = \frac{1}{N\sigma}\sum_{n=1}^{N} h_m(X_n)(\underbrace{\tilde{\theta}^n - \theta_k}_{\epsilon_n}).$$

Numerical results

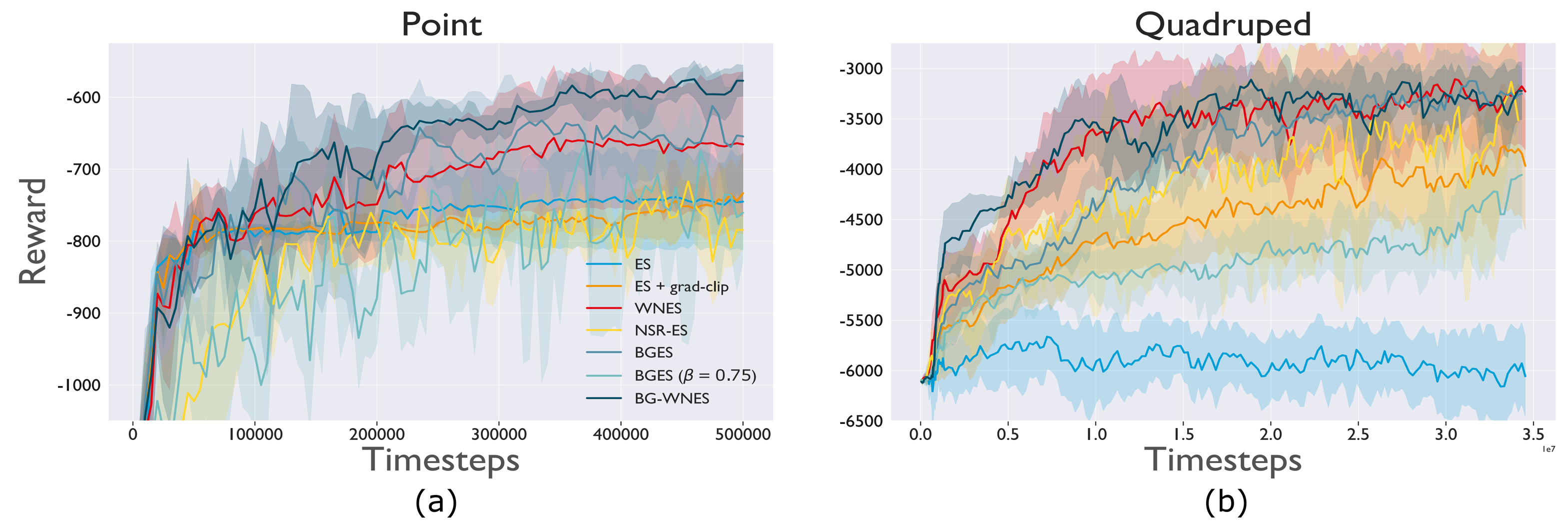
- WNPG matches or beats WD-regularized PG; BG-WNPG does even better



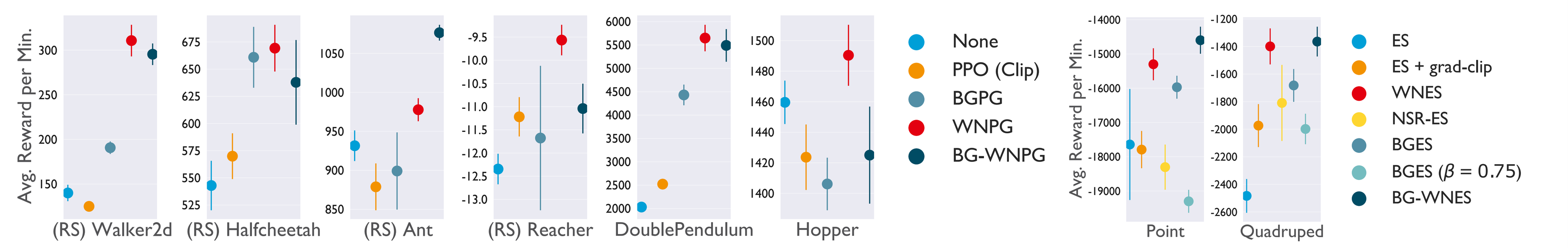
- WNPG does especially well on problems with poor conditioning:



- WNES and BG-WNES reliably navigate around local maxima:



- WNG-based methods are more computationally efficient:



Bibliography

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).

This work was funded by the Gatsby Charitable Foundation.

Contact: ted@gatsby.ucl.ac.uk