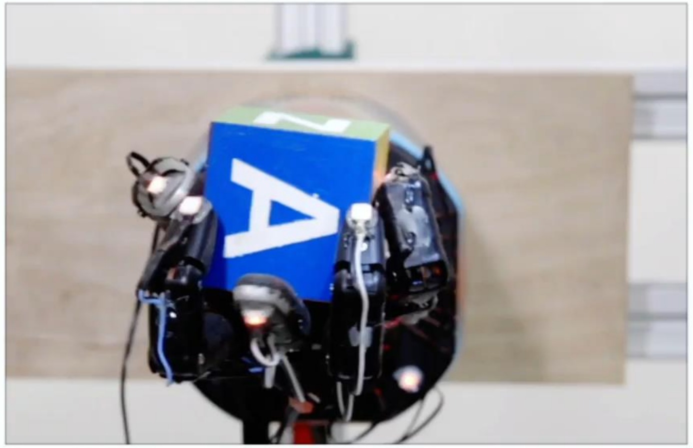
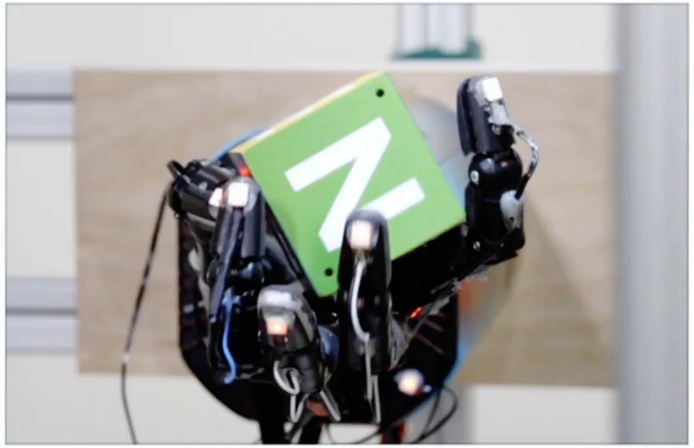
PODS: Policy Optimization via Differentiable Simulation

Miguel Zamora*, Momchil Peychev, Sehoon Ha,
Martin Vechev, Stelian Coros



FINGER PIVOTING



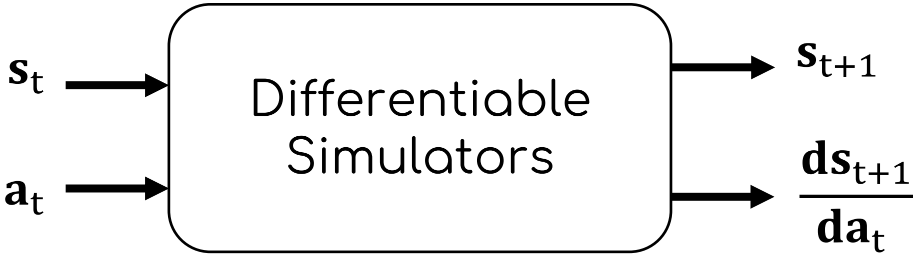
SLIDING

FINGER GAITING

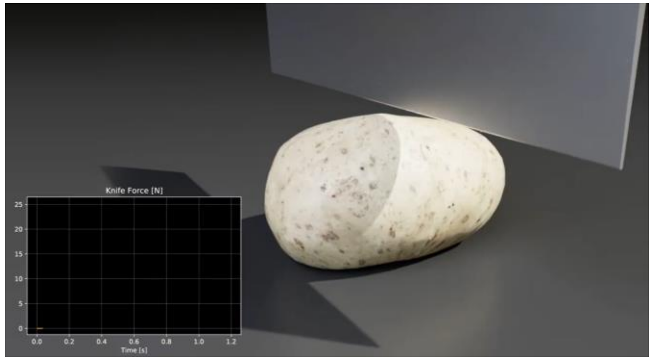
*OpenAI: Learning Dexterous In-Hand Manipulation

Goal:

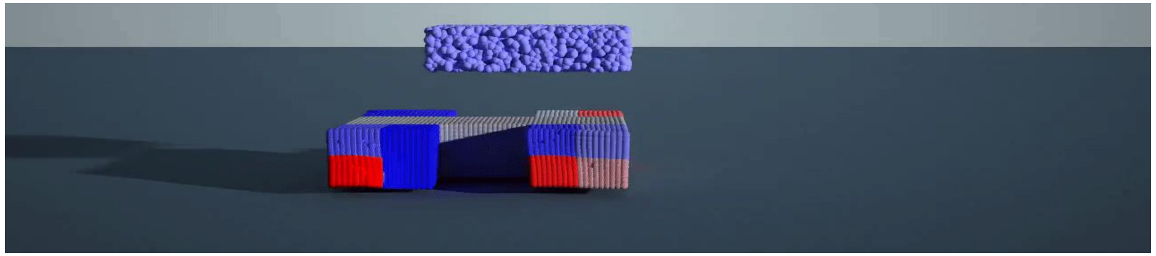
How can we best use differentiable simulators to learn policies?



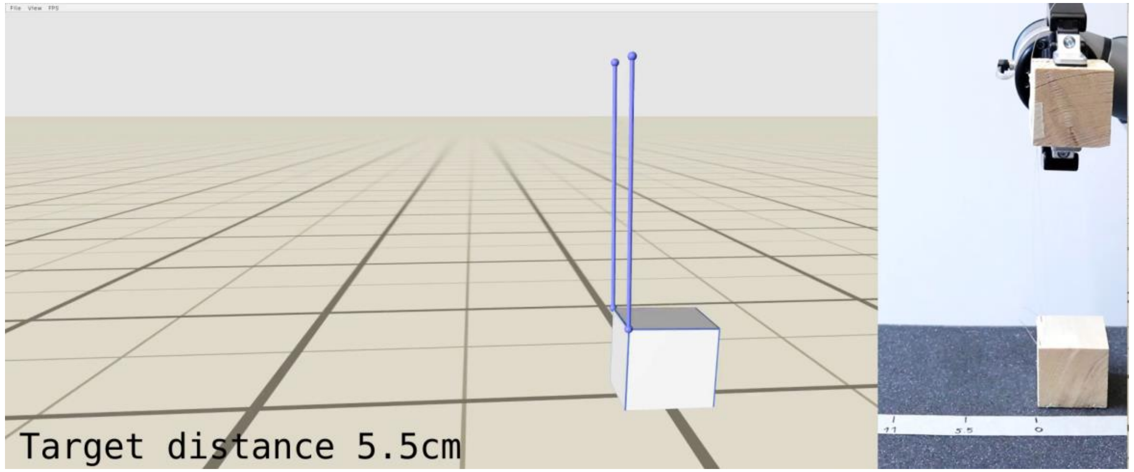
[Differentiable Cloth Simulation for Inverse Problems]



[DiSECt: A Differentiable Simulation Engine for Autonomous Robotic Cutting]

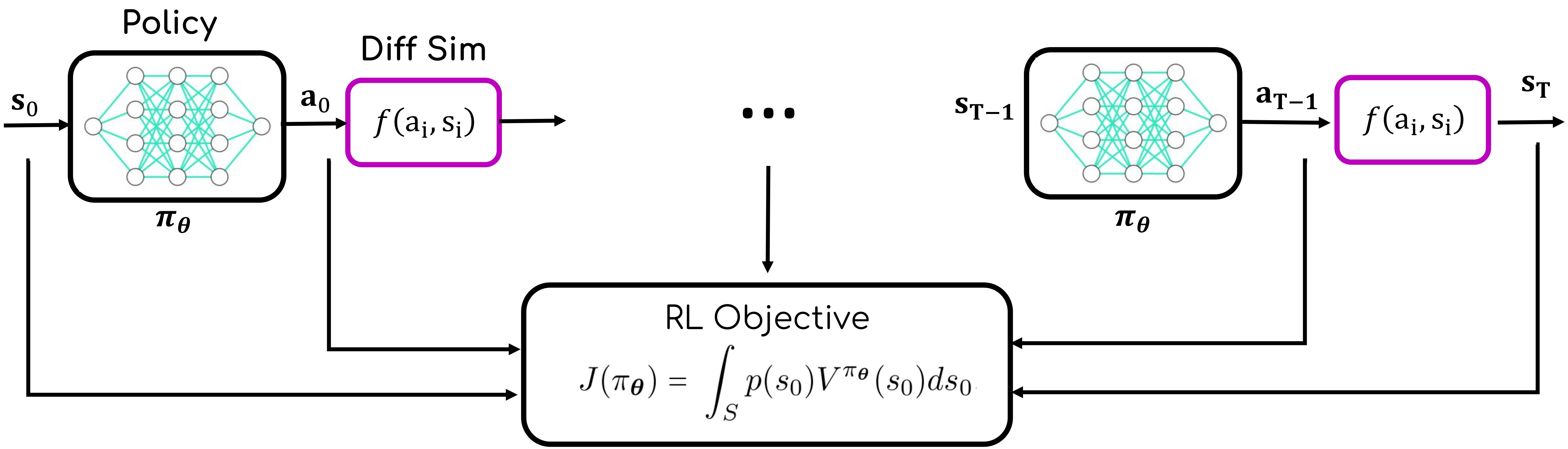


[DiffTaichi: Differentiable Programming for Physical Simulation]



[ADD: Analytically Differentiable Dynamics for Multi-Body Systems with Frictional Contact]

Differentiable simulation as a layer?



Policy Gradient:

$$\nabla_{\boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) = \int_S p(s_0) \nabla_{\boldsymbol{\theta}} V^{\pi_{\boldsymbol{\theta}}}(s_0) ds_0.$$

$$\approx \frac{1}{k} \sum_i^k \nabla_{\boldsymbol{\theta}} V^{\pi_{\boldsymbol{\theta}}}(s_{0,i}).$$

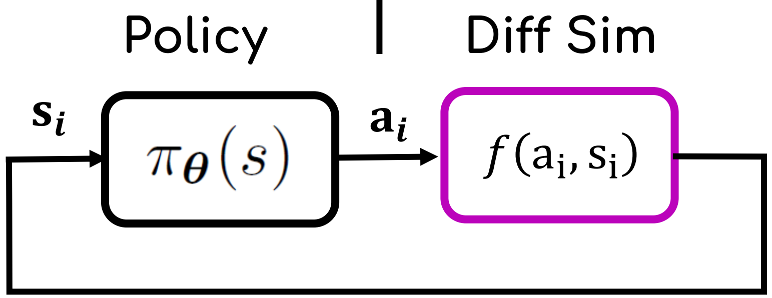
Effectively BPTT:

- Only first order
- Exploding / Vanishing gradients

PODS

Policy Improvement: $V^{\pi_\theta}(s_0) = V^{\bar{a}}(s_0)$

$$\bar{\boldsymbol{a}} = [a_0, a_1, \ldots, a_{N-1}]$$



Improve $\bar{a}$ to get $V^{\pi_\theta}(s_0) < V^{\bar{a}}(s_0)$

$$\bar{\boldsymbol{a}} = \boldsymbol{\pi_\theta} + \alpha_a \frac{\mathrm{d}V^{\bar{\boldsymbol{a}}}(s_0)}{\mathrm{d}\bar{\boldsymbol{a}}}$$

$$\frac{\mathrm{d}V^{\bar{\boldsymbol{a}}}(s_0)}{\mathrm{d}\bar{\boldsymbol{a}}} = \frac{\partial V^{\bar{\boldsymbol{a}}}}{\partial \bar{\boldsymbol{a}}} + \frac{\partial V^{\bar{\boldsymbol{a}}}}{\partial \boldsymbol{s}} \frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}\bar{\boldsymbol{a}}}$$

Info provided by diff sim

Second order improvement

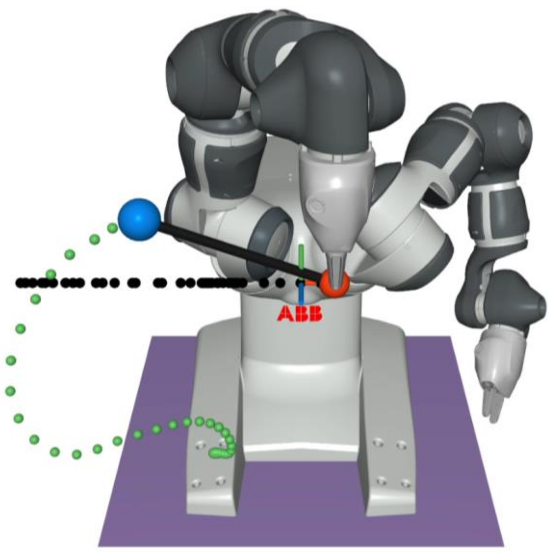
$$\bar{\boldsymbol{a}} = \boldsymbol{\pi_\theta} + \alpha_a \hat{\mathbf{H}}^{-1} \frac{\mathrm{d}V^{\bar{\boldsymbol{a}}}(s_0)}{\mathrm{d}\bar{\boldsymbol{a}}}$$

Check the paper for more details!

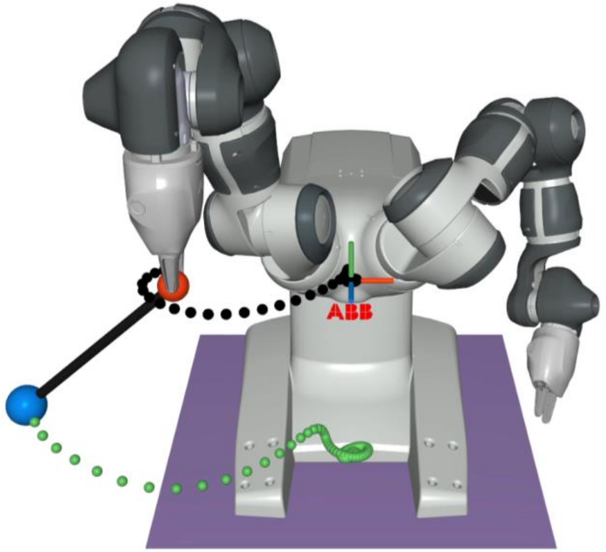
Policy update

$$L_\theta = \frac{1}{k} \sum_{i}^{k} \sum_{t}^{N} \frac{1}{2} \left\| \pi_\theta(s_{t,i}) - a_{t,i} \right\|^2$$

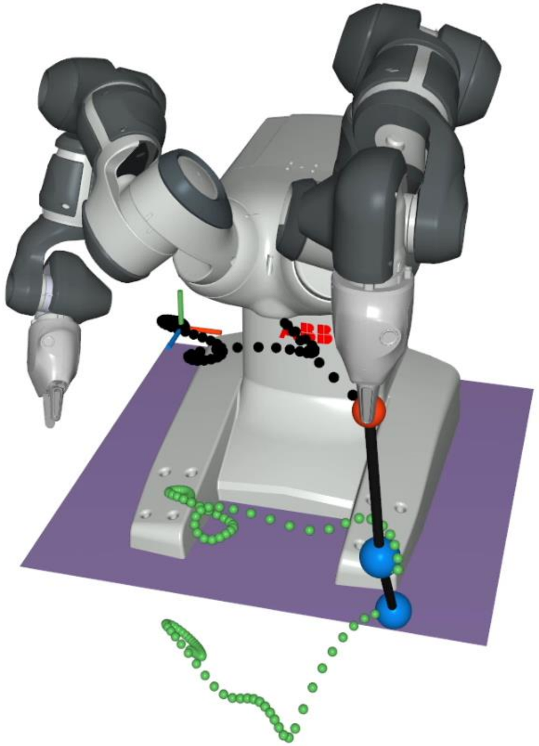
Fine manipulation tasks



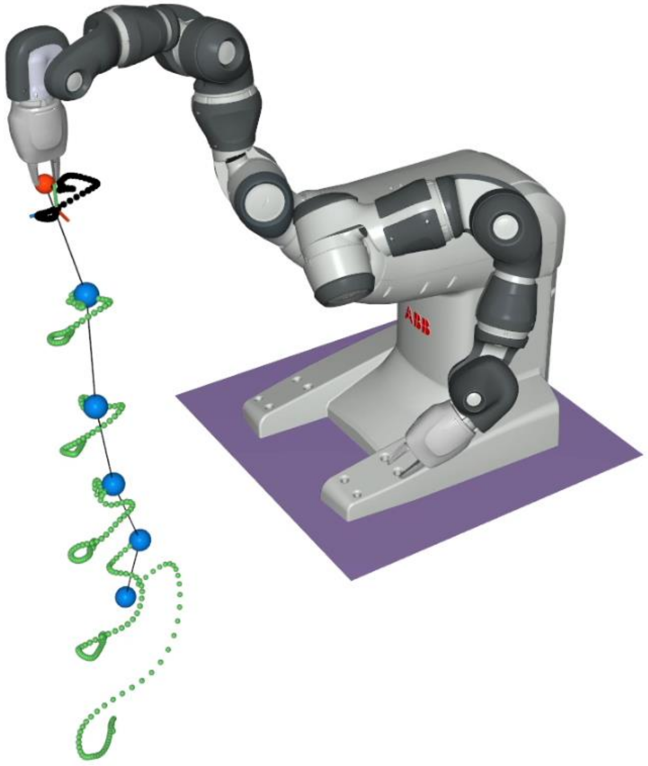
2D Pendulum
Stop as fast as possible



3D Pendulum
Stop at origin



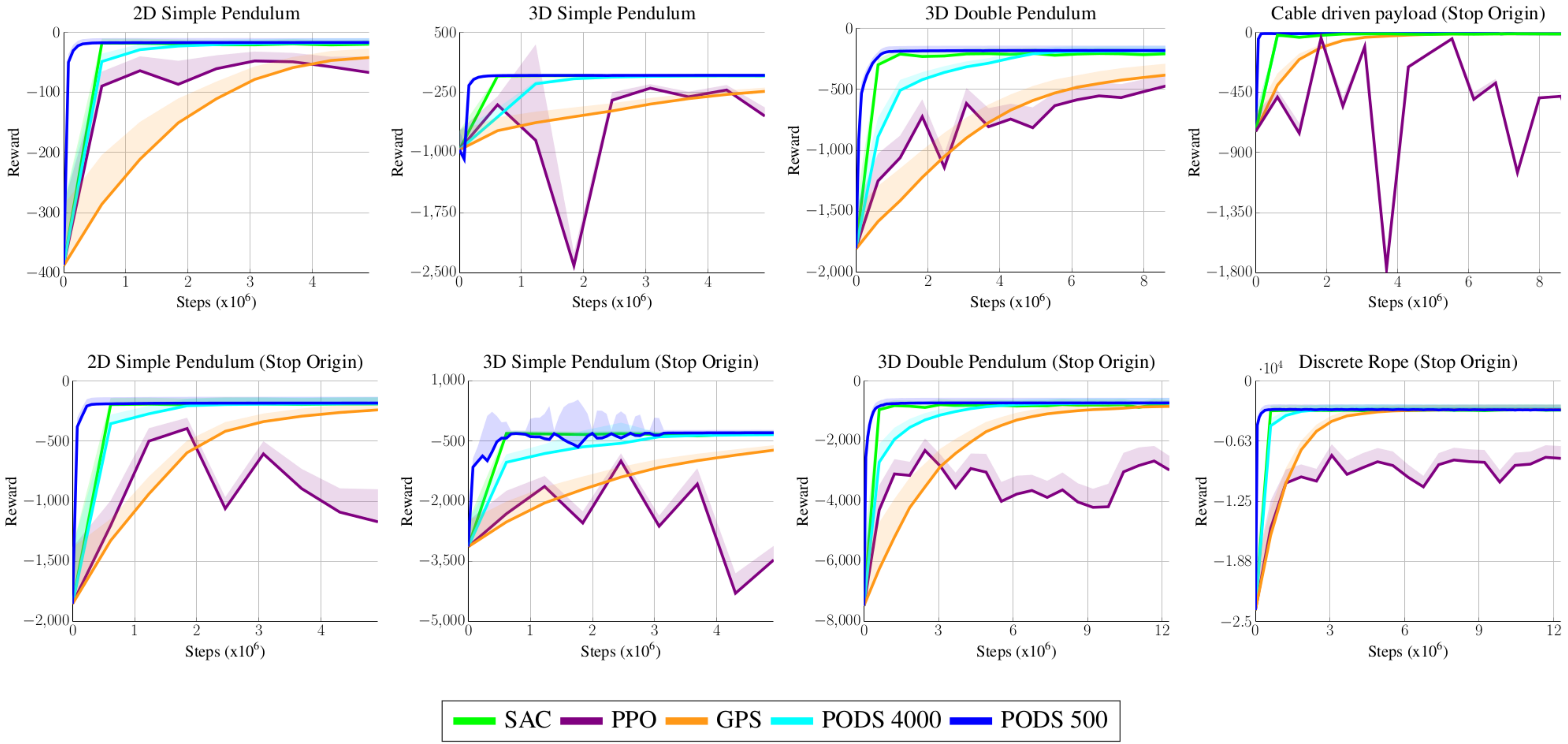
3D Double Pendulum
Stop at origin



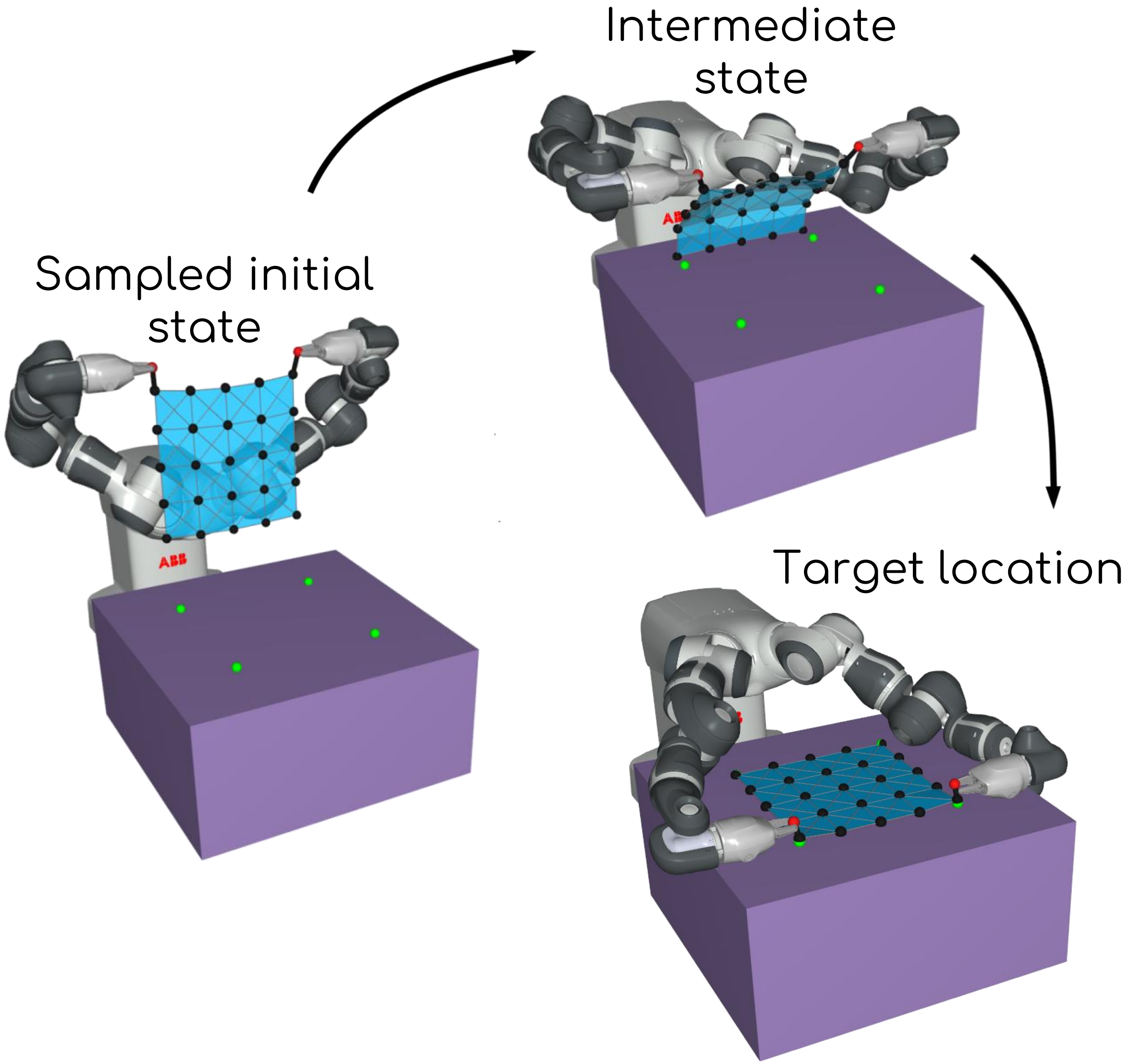
Discretized Rope
Stop at origin

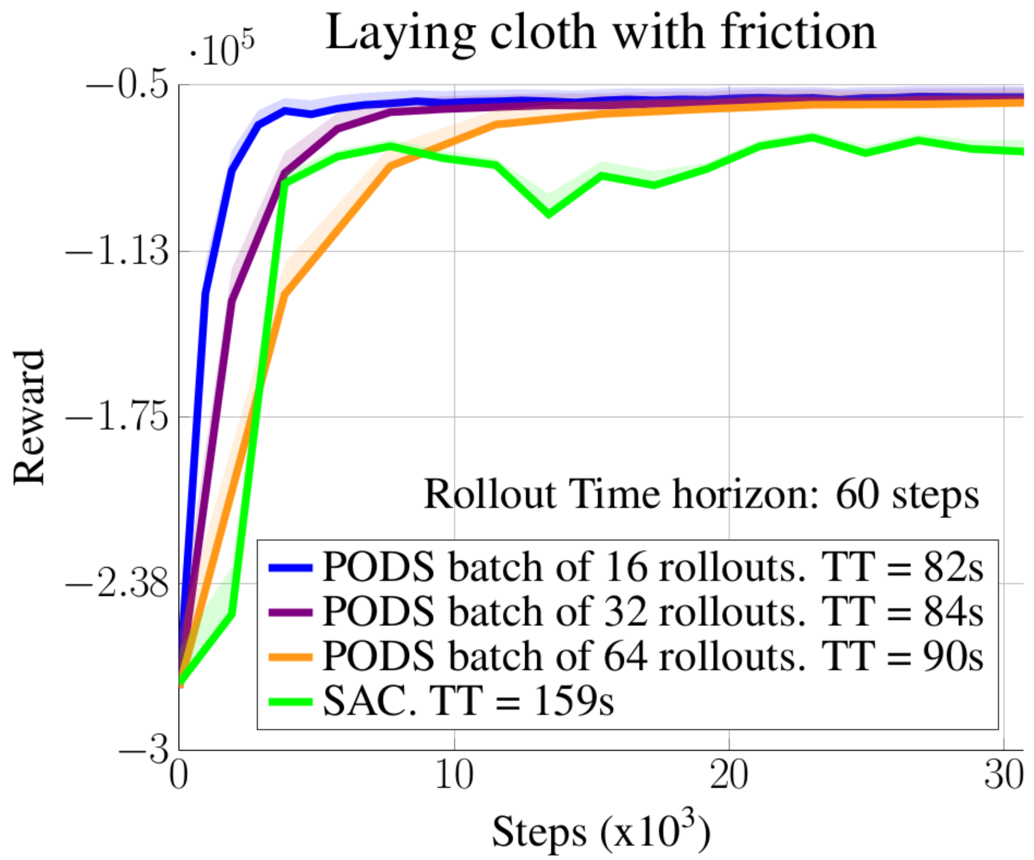
PODS – Performance

Despite additional computations
PODS is 10x to 30x faster



PODS – Complexity



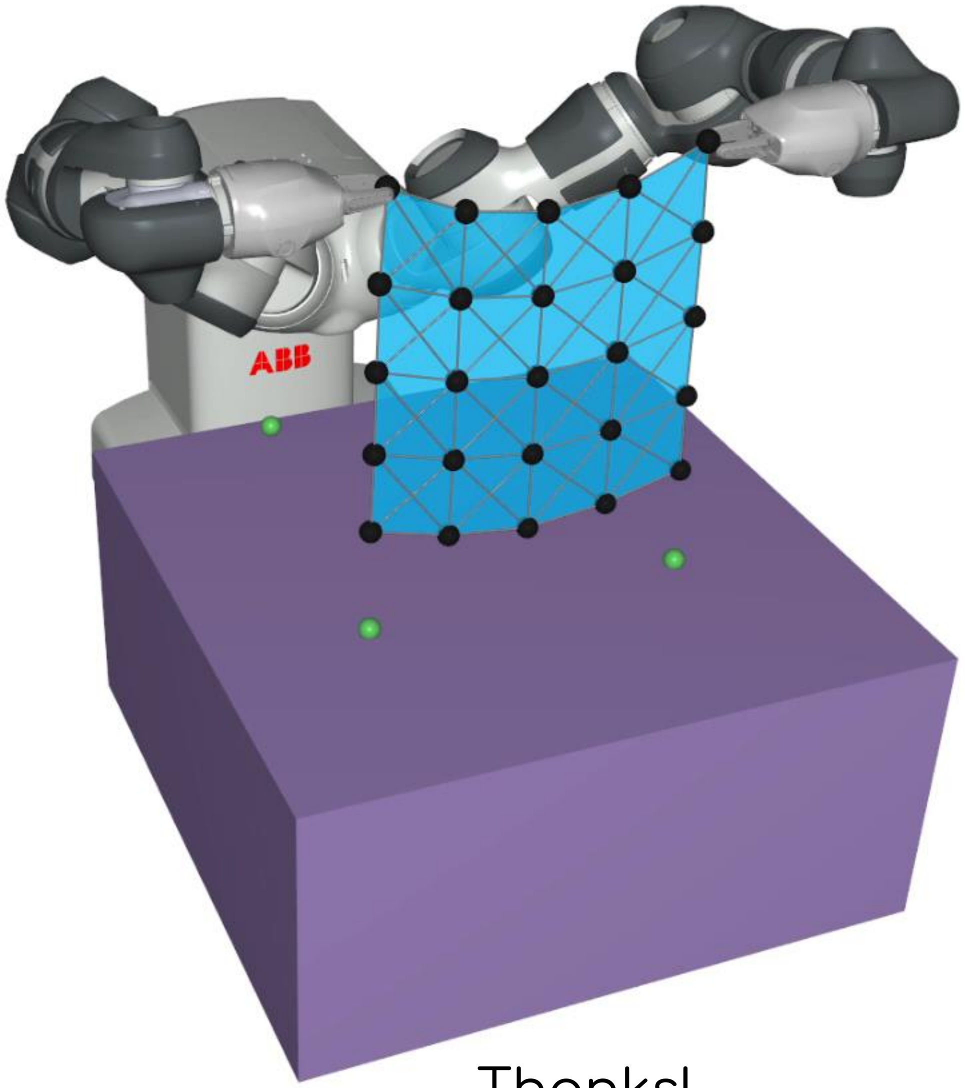


Conclusion

- Simple, fast and principled method
- Better exploit differentiable sims
- Outperformed baselines w.r.t. sample efficiency and compute time.

Future work

- Interleave with existing RL methods (exploration).
- Leverage Inverse-RL to obtain surrogate reward function for non-smooth rewards.
- Find ceiling of complexity!



Thanks!