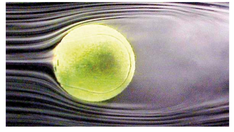
NEURAL DYNAMIC POLICIES FOR END-TO-END SENSORIMOTOR LEARNING

Shikhar Bahl Mustafa Mukadam Abhinav Gupta Deepak Pathak
CMU & FAIR

Code Released!
https://shikharbahl.github.io/neural-dynamic-policy/

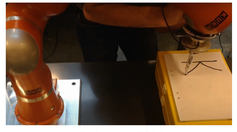
Naturally occurring phenomena are *dynamical systems.*

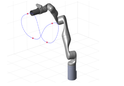
Can we leverage ***dynamical systems*** reason in trajectory space?




Dynamical systems in robotics literature have been used to control robots (e.g. **DMPs** [Schaal., 2002], etc).

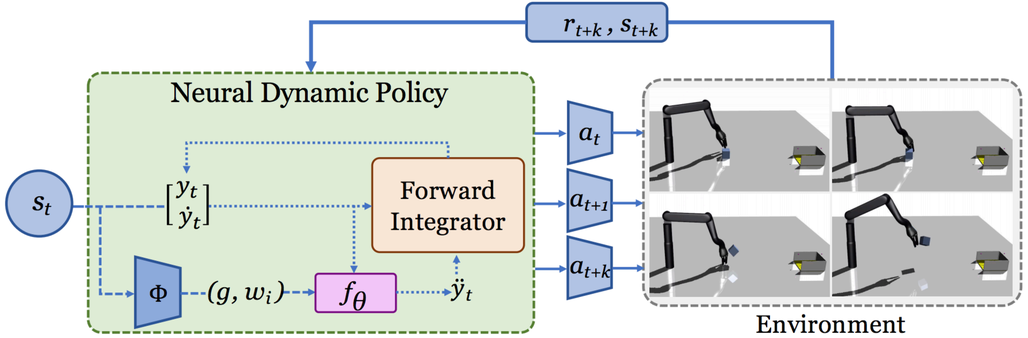

[Muelling et. al, 2013]


[Steinmetz, 2014]

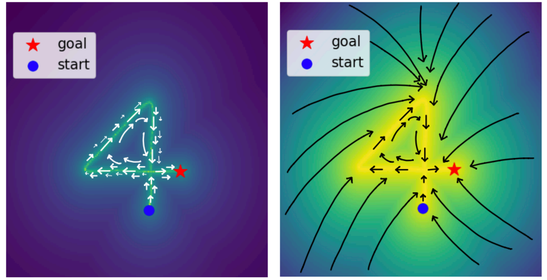
	Deep RL	DMPs
	✓ Learning from weak supervision (reward)	✗ Can only learn from demonstrations
	✗ Reason at each timestep	✓ Reason in trajectory space
 [Kormushev et. al, 2010]	✗ Struggle with dynamic tasks	✓ Good with dynamic tasks
	✓ Good with high dimensional inputs	✗ Cannot handle unstructured inputs

Can we bridge the gap between these two paradigms?

Ours: Neural Dynamic Policies



- Fully **differentiable dynamical system** embedded as a layer
- Can operate with high dimensional inputs, i.e. **images**



Vanilla Policy NDP (Ours)

NDP can operate a trajectory level while maintaining the ***benefits of end-to-end methods***

NDP Structure

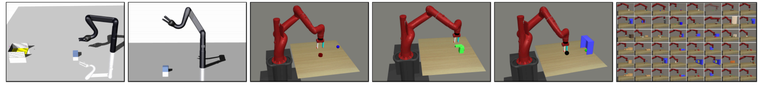
We embed a second order dynamical system inside a network

$$\ddot{y} = \alpha(\beta(g - y) - \dot{y}) + f(x)$$

Acceleration is determined by a forcing function of weighted RBFs

$$f(x, g) = \frac{\sum \psi_i w_i}{\sum \psi_i} x(g - y_0), \quad \psi_i = e^{(-h_i(x - c_i)^2)}$$

Imitation Learning


(a) Throwing (b) Picking (c) Pushing (d) Faucet Open (e) Soccer (f) 50 Tasks

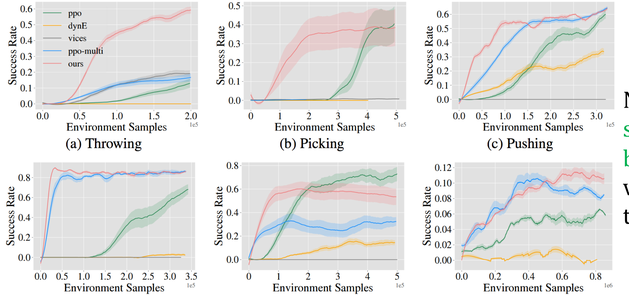
Method	NN	NDP (ours)
Throw	0.528 ± 0.262	$\mathbf{0.642 \pm 0.246}$
Push	0.002 ± 0.004	$\mathbf{0.208 \pm 0.049}$
Soccer	0.885 ± 0.016	$\mathbf{0.890 \pm 0.010}$
Faucet	0.532 ± 0.231	$\mathbf{0.790 \pm 0.059}$

NDPs outperform baseline in ***imitation learning*** for various tasks


Input Ours CNN DMP

We show strong success in digit tracing with a robot

Reinforcement Learning



NDPs outperforms state-of-the-art baselines, across various control tasks.