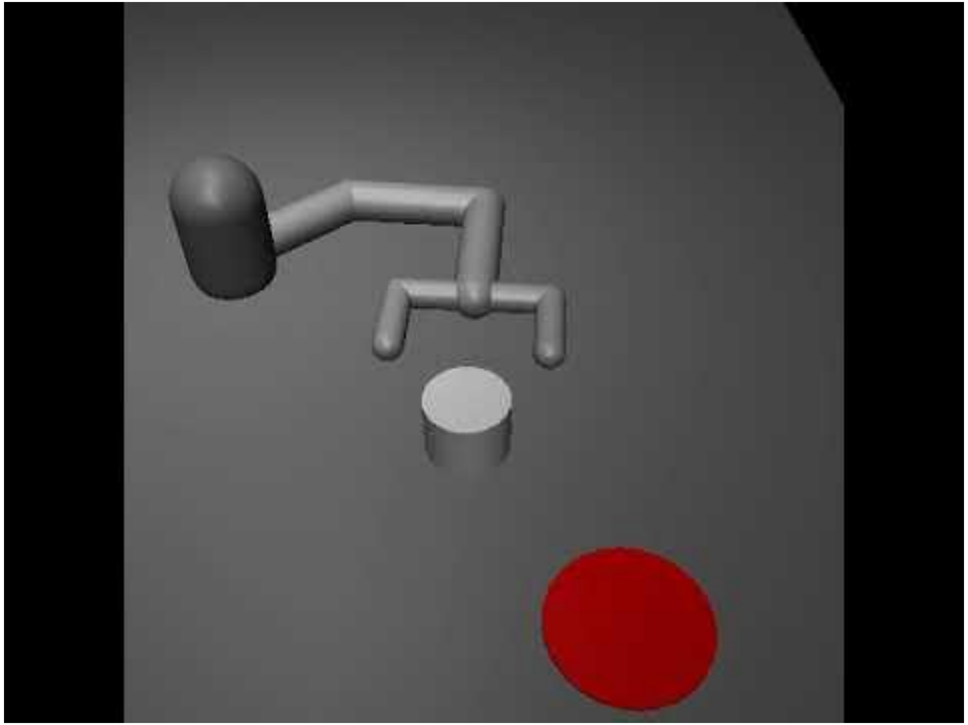
Sim 2 Real Zero-Shot Policy Transfer

w/ INRIA Bordeaux, MILA Montreal, and McGill Montreal

Problem: Sim 2 Real is hard.

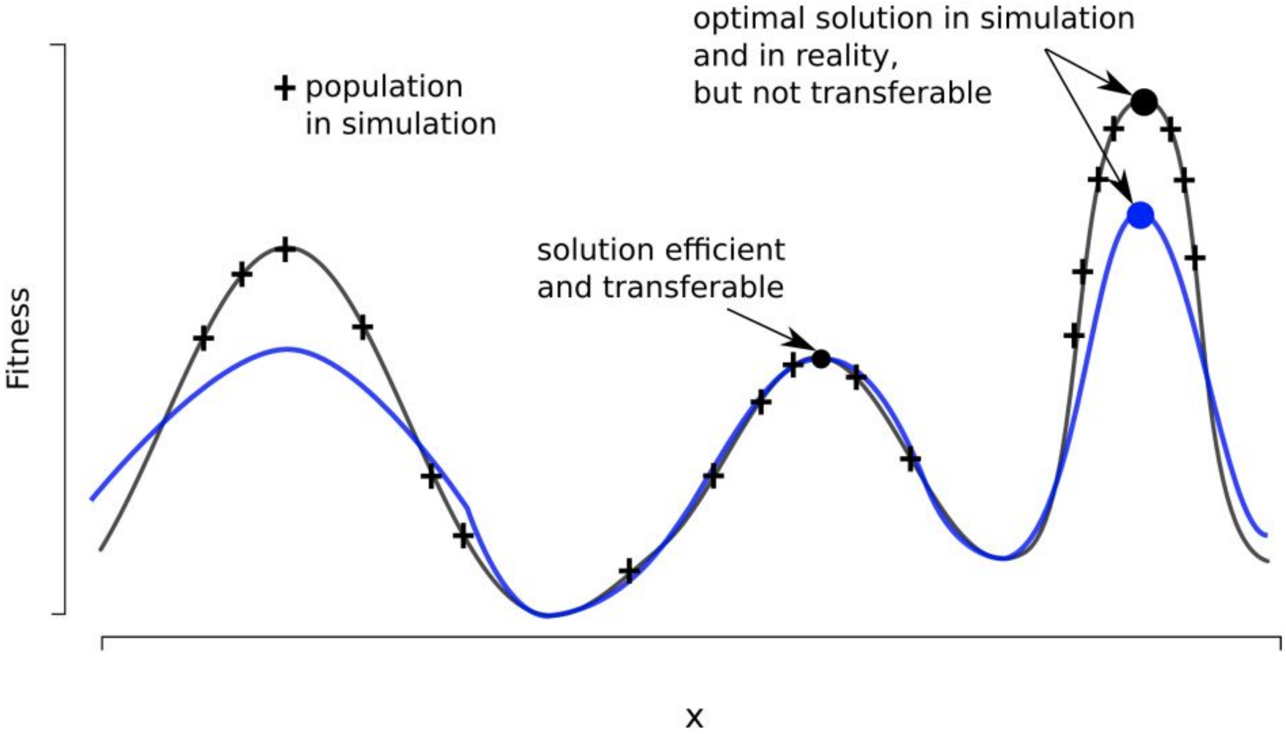


Background

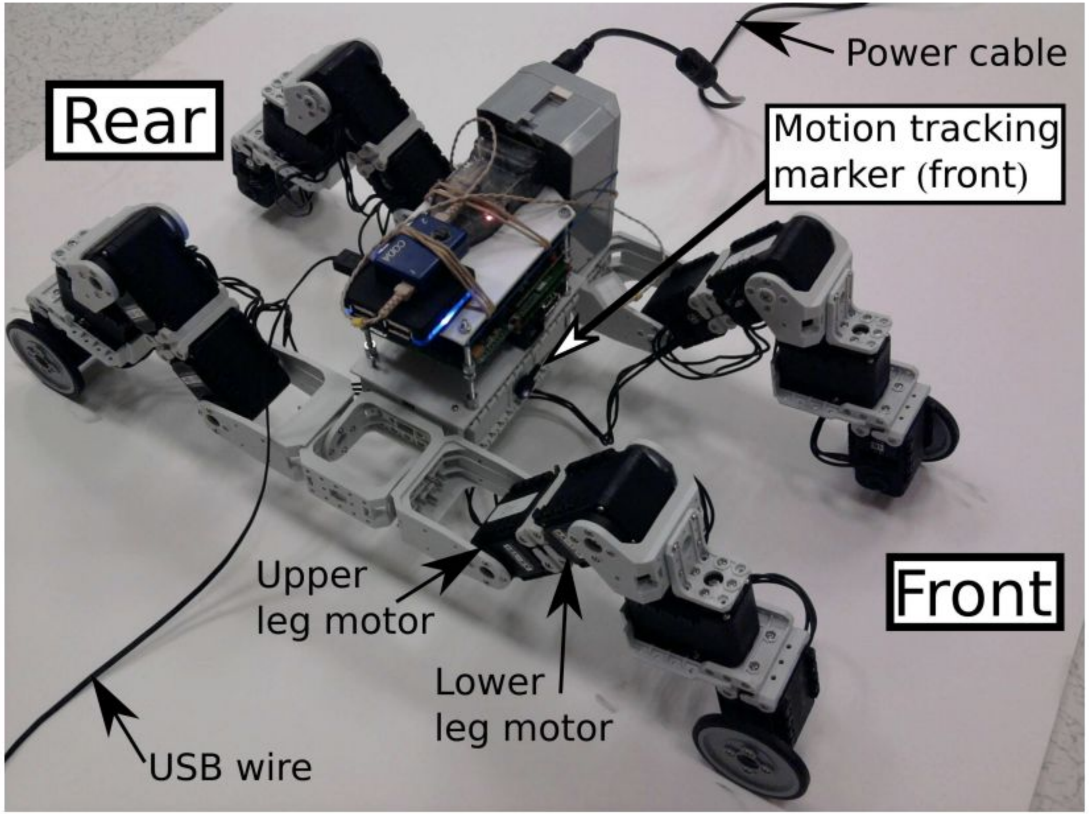
Evolutionary Strategies

Make “transferability” part of your fitness function.

Evaluate frequently.



Koos S, Mouret JB, Doncieux S. The transferability approach: Crossing the reality gap in evolutionary robotics. IEEE Transactions on Evolutionary Computation. 2013 Feb;17(1).



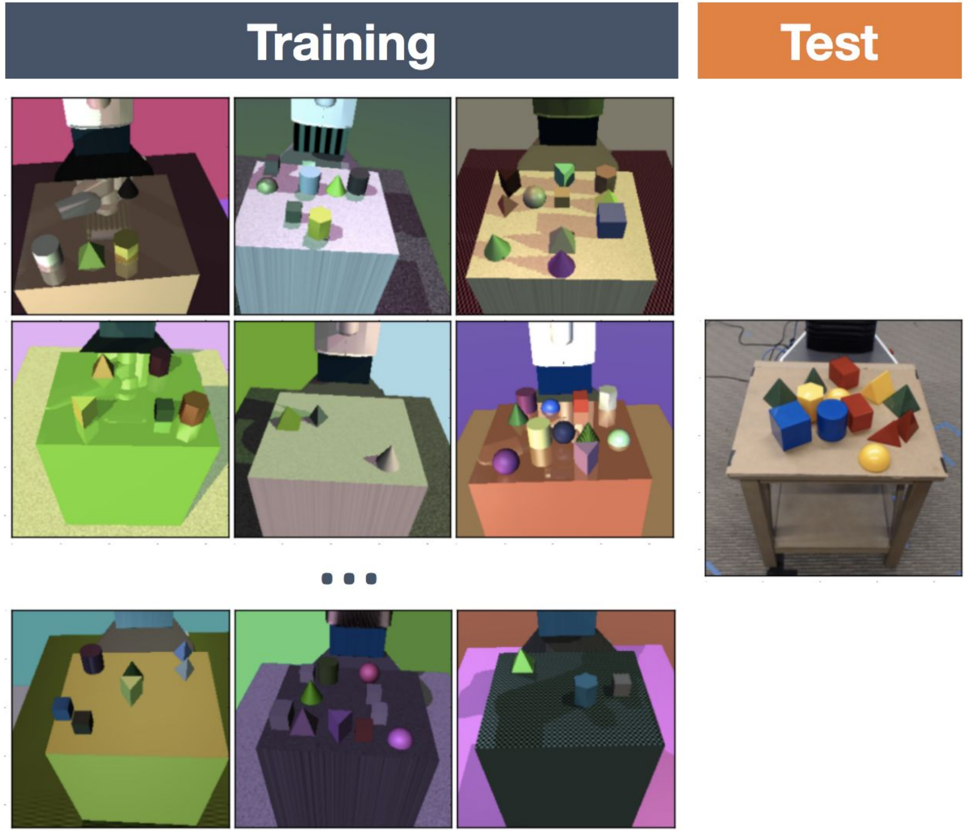
Koos S, Mouret JB, Doncieux S. The transferability approach: Crossing the reality gap in evolutionary robotics. IEEE Transactions on Evolutionary Computation. 2013 Feb;17(1).

Domain Randomization

Progressively randomize environment.

Train policy in all kinds of random situations.

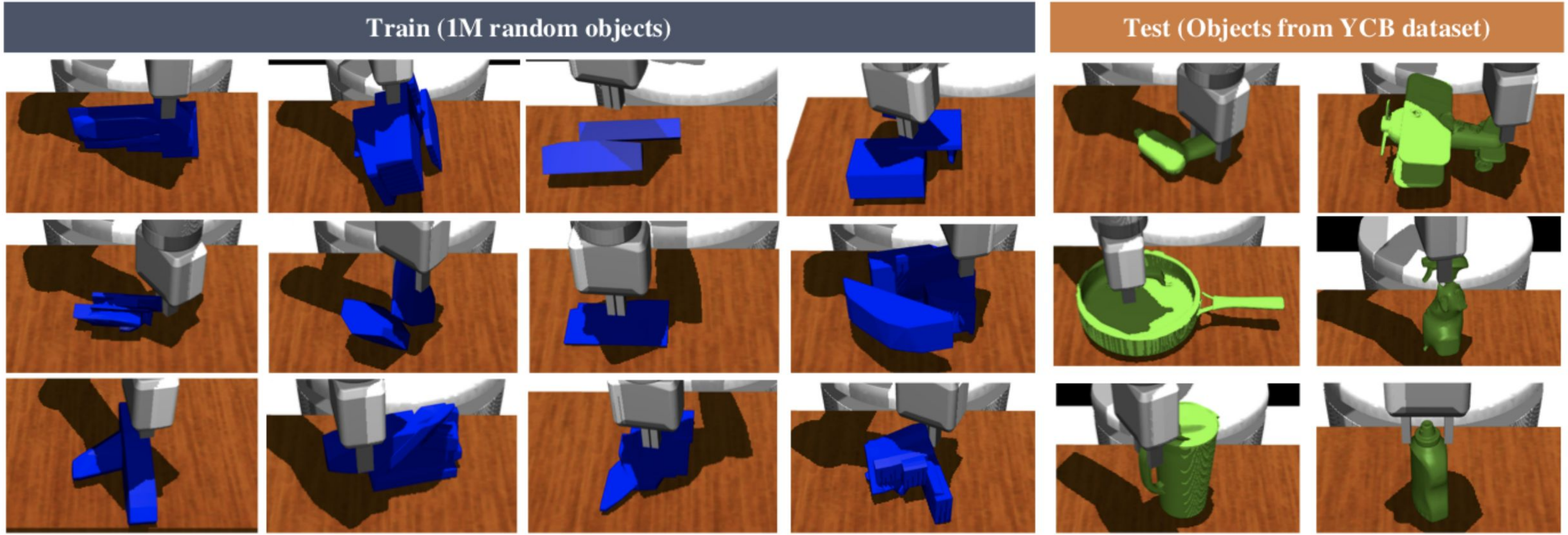
Reality becomes one of the possible instances.



Training

Test

Tobin J, Fong R, Ray A, Schneider J, Zaremba W, Abbeel P. Domain randomization for transferring deep neural networks from simulation to the real world. In Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on 2017 Sep 24. IEEE.



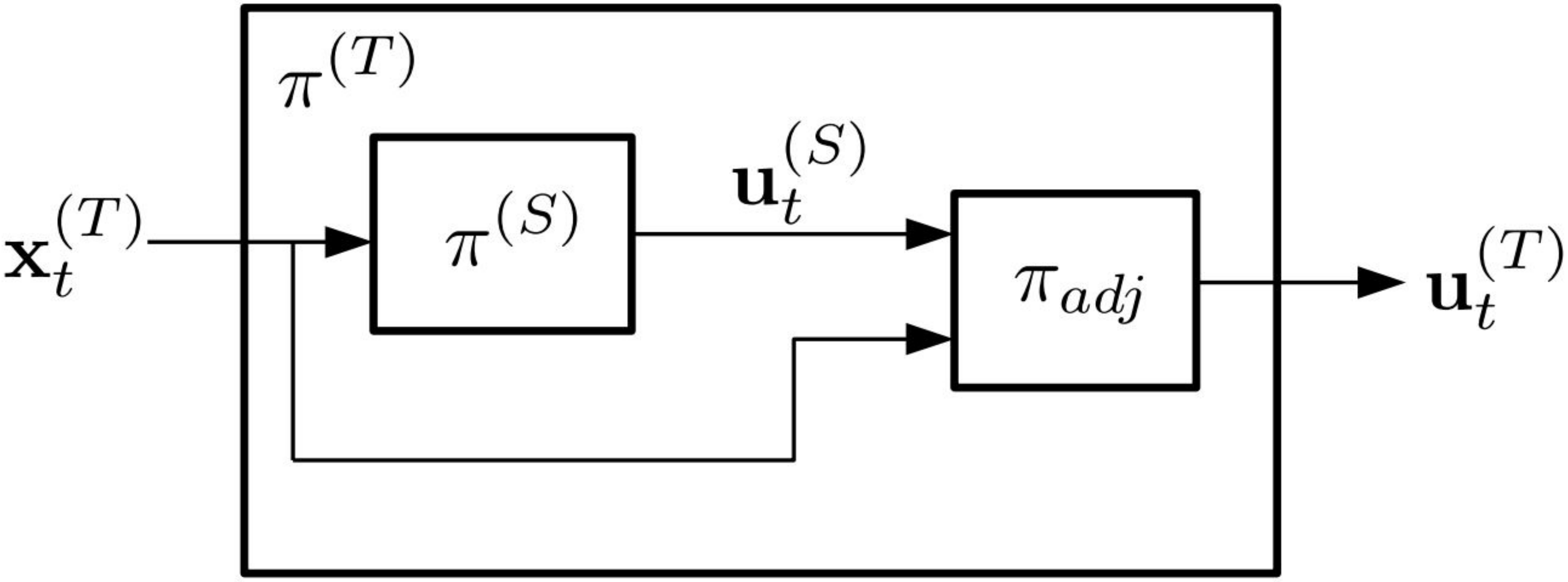
Tobin J, Zaremba W, Abbeel P. Domain Randomization and Generative Models for Robotic Grasping. arXiv preprint arXiv:1710.06425. 2017 Oct 17.

Policy Adjustment

Learn source policy (simulation).

Transfer policy and record “deviation” / inverse dynamics model via small noise perturbation.

For target (real) apply source policy + learned policy adjustment.



Higuera JC, Meger D, Dudek G. Adapting learned robotics behaviours through policy adjustment. In Robotics and Automation (ICRA), 2017 IEEE International Conference on 2017 May 29. IEEE.

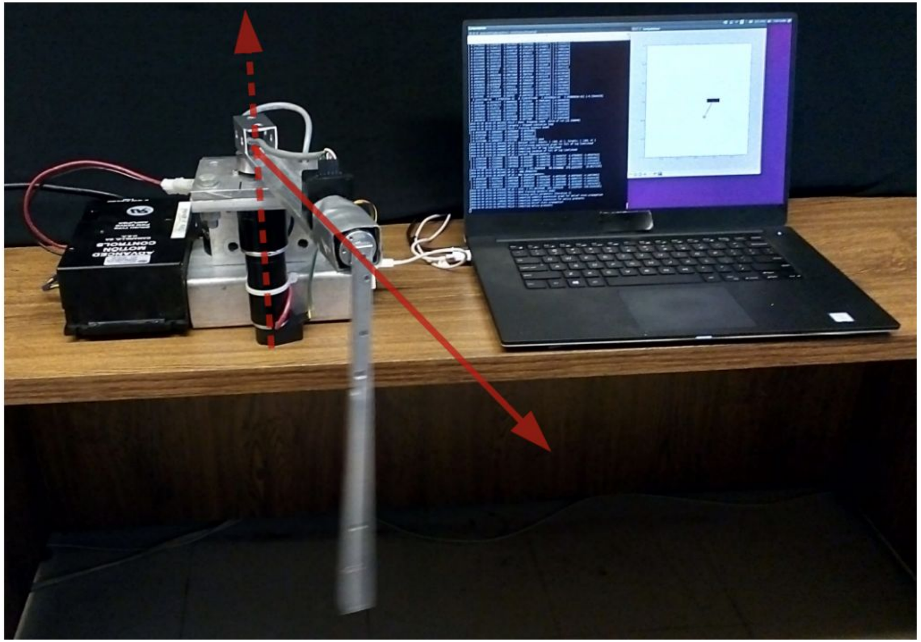


Fig. 4. Our experimental setup for the cart-pole domain. The solid arrow denotes the rotation axis of the pendulum. The dashed line denotes the rotation axis of the actuator.

Higuera JC, Meger D, Dudek G. Adapting learned robotics behaviours through policy adjustment. In Robotics and Automation (ICRA), 2017 IEEE International Conference on 2017 May 29. IEEE.

Our Approach

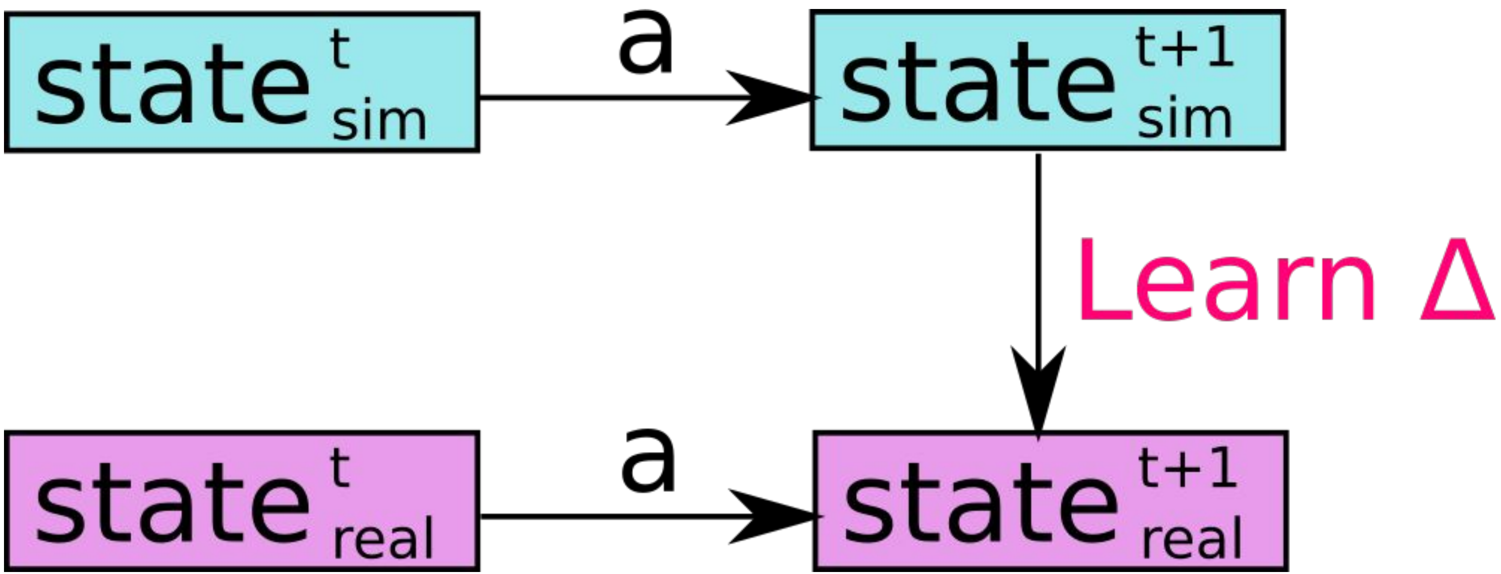
Our Approach

Focus on tasks that are hard to reset IRL (like juggling).

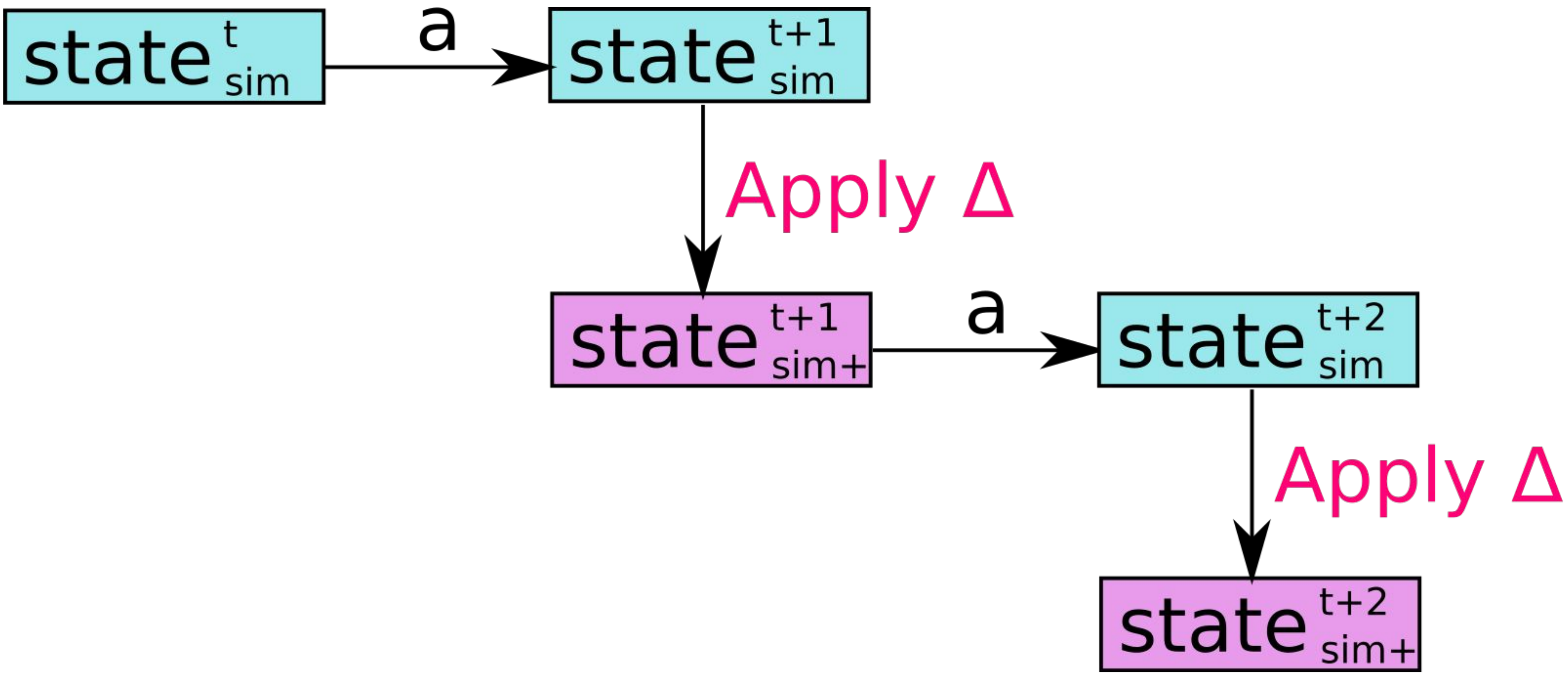
Learn model of robot dynamics on completely unrelated "easy" task.

Take dynamics model and apply it to simulation states of hard task.

Learn policy from "new" observed dynamics.



Easy task (random exploration or goal babbling)



$\text{state}_{\text{sim}}^{t}$
a
$\text{state}_{\text{sim}}^{t+1}$
Apply Δ
$\text{state}_{\text{sim+}}^{t+1}$
a
$\text{state}_{\text{sim}}^{t+2}$
Apply Δ
$\text{state}_{\text{sim+}}^{t+2}$

Hard Task, learn policy on “Sim+” states

Handle Compounding Effects with RNN

Methods

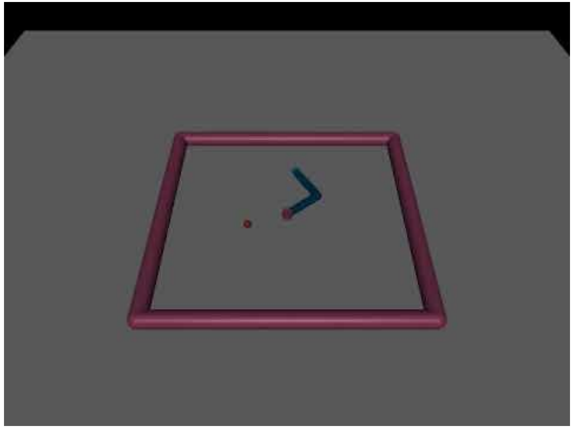
Environments

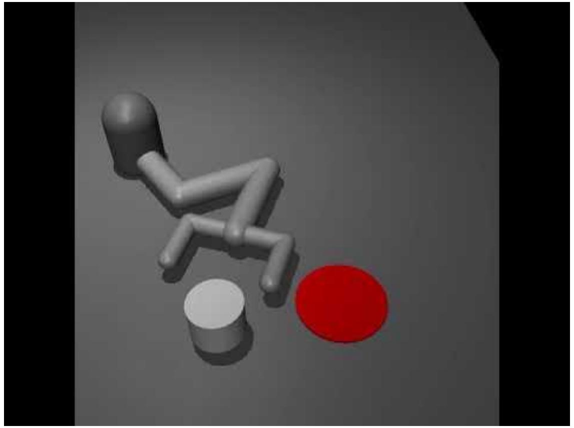
MuJoCo (with multi-joint backlash):

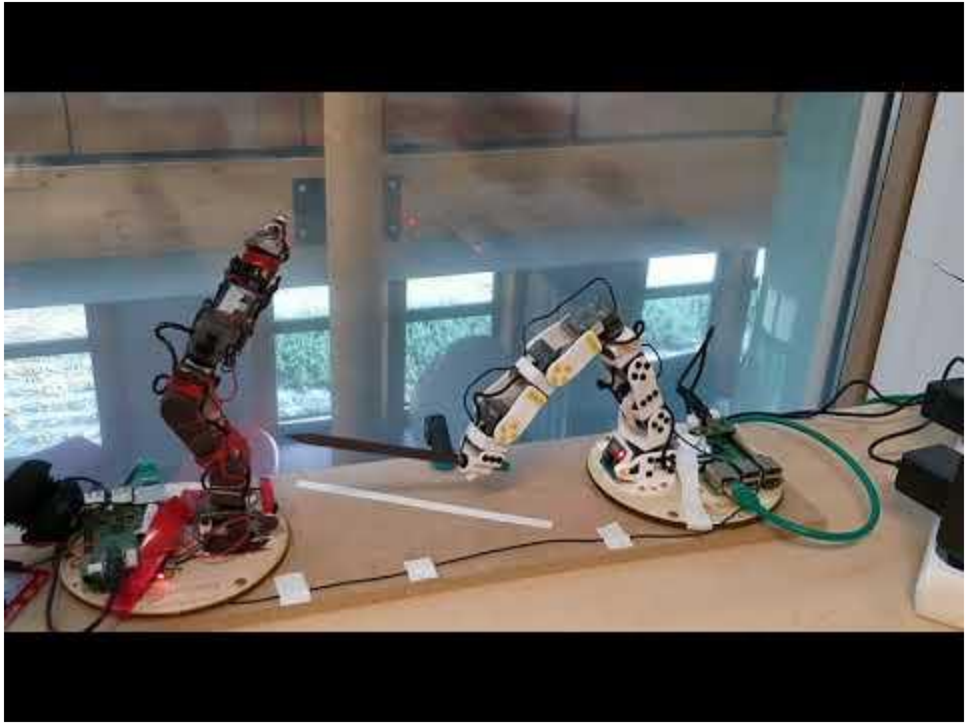
- Reacher,
- Pusher... failed,
- Pusher3Dof,
- Striker

Poppy Ergo Jr (with added weights on tip):

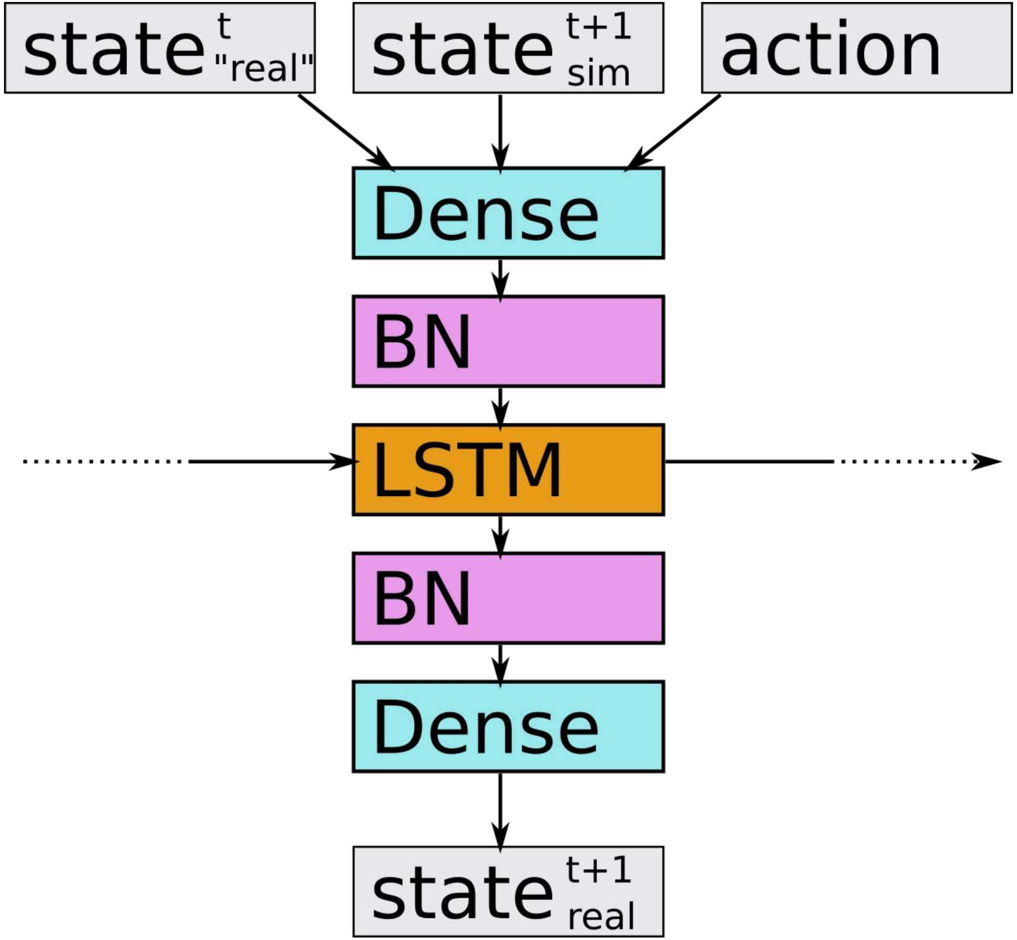
- Sword Fight,
- (Ball Throw)





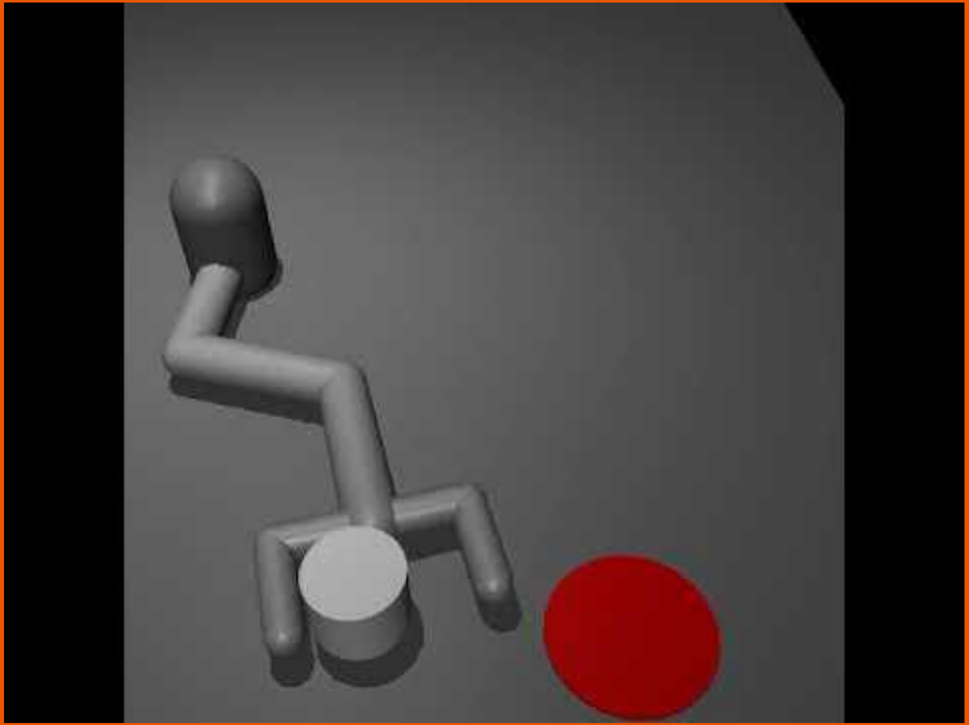


Architecture

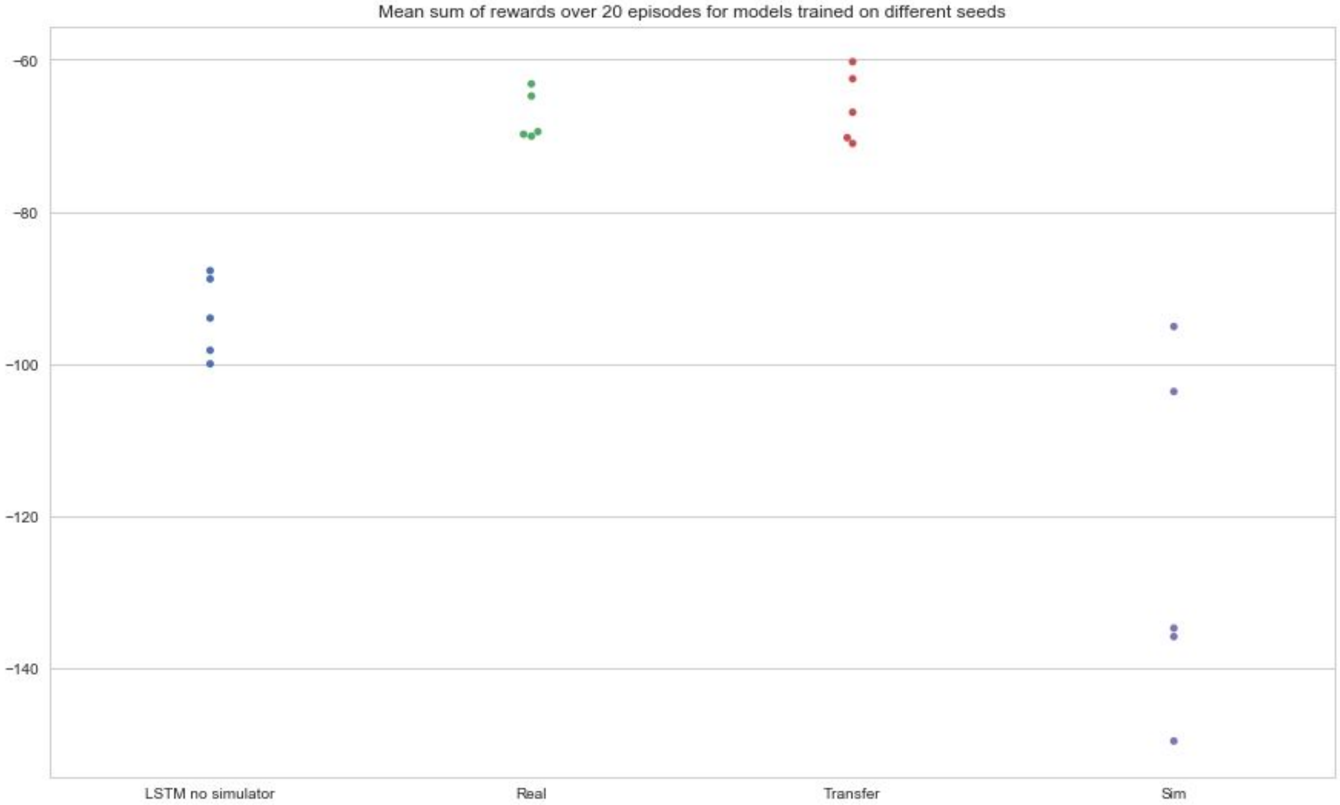


Results

It works



Results


Mean sum of rewards over 20 episodes for models trained on different seeds
−60
−80
−100
−120
−140
LSTM no simulator
Real
Transfer
Sim

Discussion

Next Steps

Find HP that work across all tasks

Finish robot implementation / self-play policy

GPR good at modelling positions, but not velocities (also no temporal correction)

CycleGAN for state ground truth

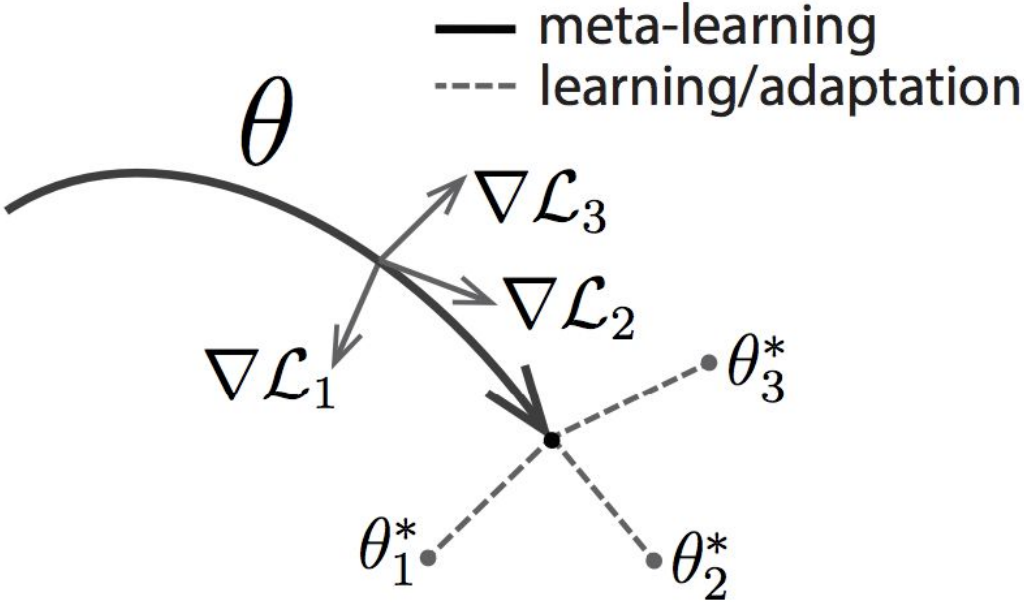
Problem

Target task needn’t introduce too many **dimensions** of dynamics.

Need to verify temporal-predictive power.

Future

Check out Chelsea Finn’s MAML.



Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. arXiv preprint arXiv:1703.03400. 2017 Mar 9.

Thanks. That’s it.

Pro tip: normalize your data