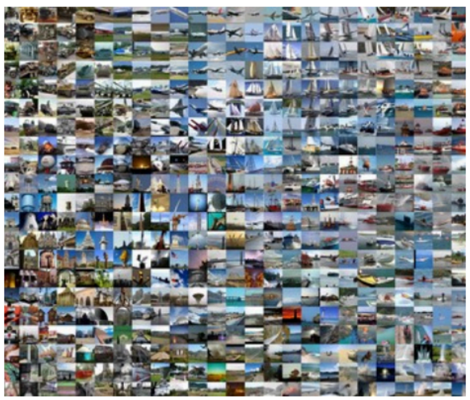
Sim-to-real

Will policies trained in simulated worlds perform well in the real world?

If not, how can this be fixed?

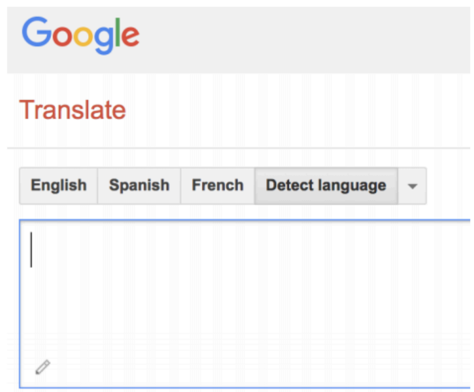
Deep learning is data hungry…

ImageNet



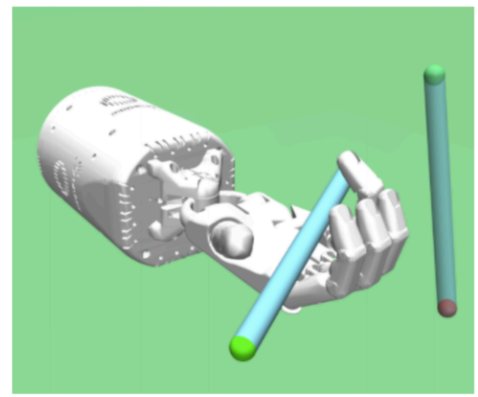
1.2M labeled images

Machine Translation



36M sentence pairs (WMT En->Fr)
"Several orders of magnitude more" (production data)

DeepRL



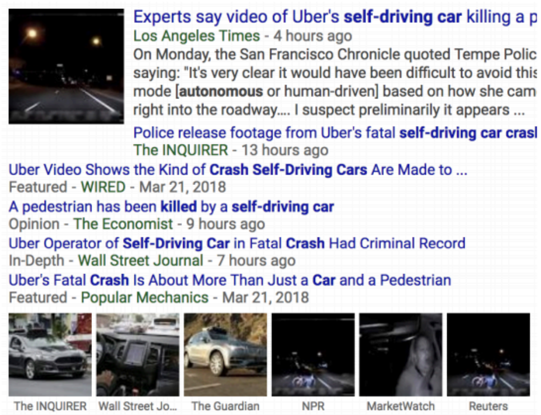
38M timesteps

[Josh Tobin]]

... But robotic data is expensive

Robot cost

Safety



Labeling

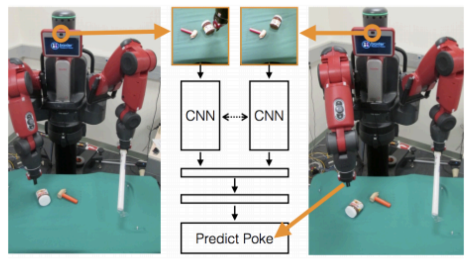
[Josh Tobin]]

SCALE-UP
ROBOTIC DATA COLLECTION?

Large-scale robotic data collection

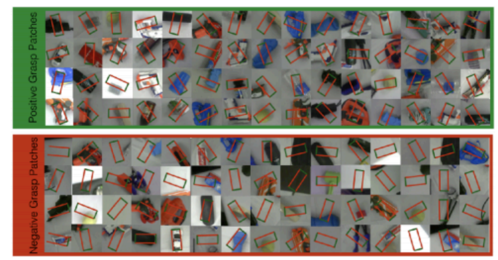
3,000 hours

Learning Hand-Eye Coordination with Deep Learning and Large Scale Data Collection [Levine, Pastor, Krizhevsky, Quillen, 2016]



400 hours

Learning to Poke by Poking: Experiential Learning of Intuitive Physics [Agarwal, Nair, Abbeel, Malik, Levine, 2016]

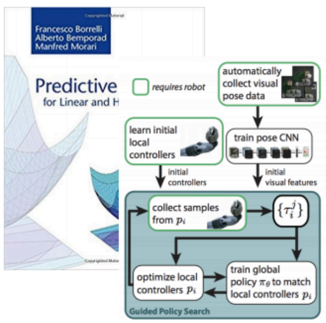


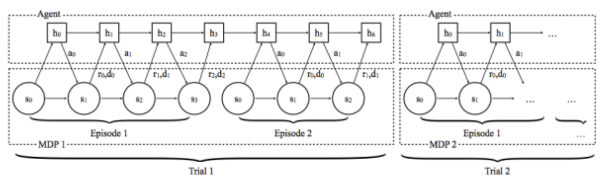
700 hours

Supervising Self-Supervison: Learning to Grasp from 50K Tries [Pinto, Gupta, ICRA 2016]

MORE DATA-EFFICIENT LEARNING?

Efficient reinforcement learning





Model-based

Predictive Control
[Borrelli, Bemporad, Morari, 2017]
End-to-End Training of Deep Visuomotor Policies
[Levine*, Finn*, Darrell, Abbeel, 2016]

Meta-learning

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks
[Finn, Abbeel, Levine, 2017]
RL2: Fast Reinforcement Learning Via Slow Reinforcement Learning
[Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel, 2016]

Learning from Demonstrations

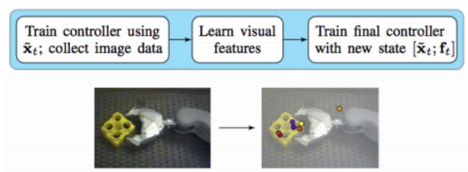
Deep Object-Centric Representations for Generalizable Robot Learning [Devin, Abbeel, Darrell, Levine, 2017]
Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation [Zhang, McCarthy, Jow, Lee, Chen, Goldberg, Abbeel, 2017]
One-Shot Imitation from Observing Humans via Domain-Adaptive Meta-Learning [Yu*, Finn*, Xie, Dasari, Zhang, Abbeel, Levine, 2018]

Unsupervised robotic learning

Augment with self-supervised tasks

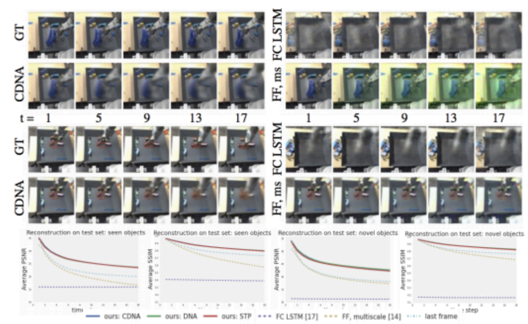
Hindsight Experience Replay [Andrychowicz, Wolski, Ray, Schneider, Fong, Welinder, McGrew, Tobin, Abbeel, Zaremba, 2017]
Loss is its own Reward: Self-Supervision for Reinforcement Learning [Shelhamer, Mahmoudich, Argus, Darrell, 2017]

Learn a feature space



Deep Spatial Autoencoders for Visuomotor Learning [Finn, Tan, Duan, Darrell, Levine, Abbeel 2016]

Learn a model



Unsupervised Learning for Physical Interaction through Video Prediction [Finn, Goodfellow, Levine, 2016]

CURRENTLY: TRAIN IN SIMULATION

Real-Time Human Pose Recognition in Parts from Single Depth Images

Jamie Shotton Andrew Fitzgibbon Mat Cook Toby Sharp Mark Finocchio
Richard Moore Alex Kipman Andrew Blake
Microsoft Research Cambridge & Xbox Incubation

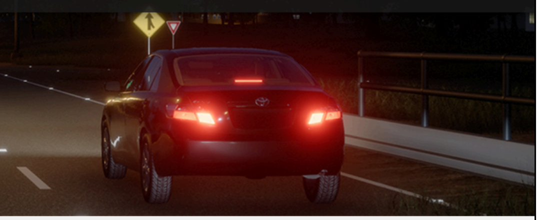
Figure 2. **Synthetic and real data.** Pairs of depth image and ground truth body parts. Note wide variety in pose, shape, clothing, and crop.

SELF-DRIVING CARS PRODUCTS SOLUTIONS PARTNERS FOR DEVELOPERS

Products

NVIDIA DRIVE CONSTELLATION

Virtual Reality Autonomous Vehicle Simulator



TEST AND VALIDATE BILLIONS OF MILES IN THE DATACENTER

Imagine being able to test an autonomous vehicle in a near-infinite variety of conditions and scenarios—before it even reaches the road. NVIDIA is making it happen, enabling the industry to safely drive billions of qualified miles in virtual reality with the powerful new NVIDIA DRIVE™ Constella

Careers

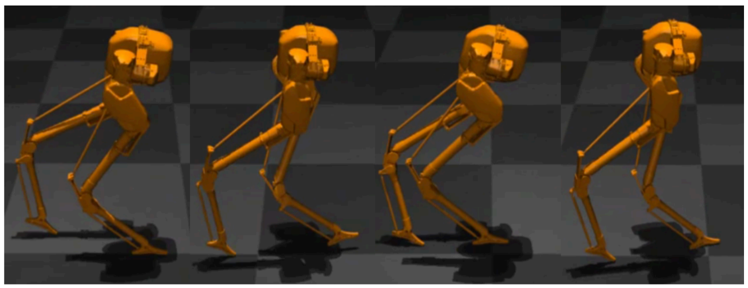
Autopilot Simulation, Rendering Engineer

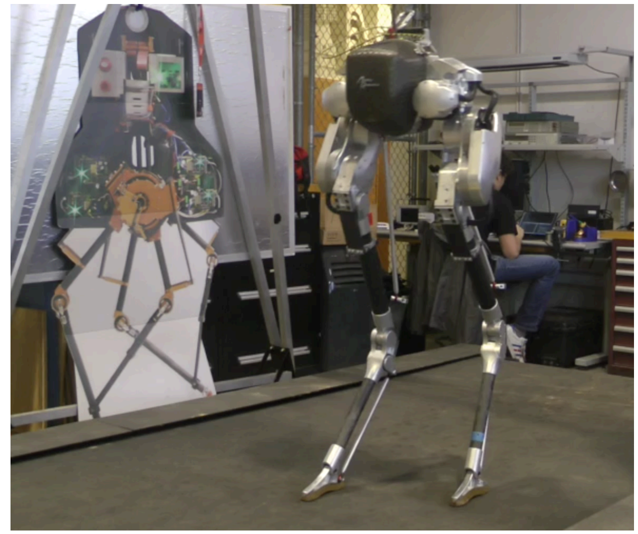
Job Category	Engineering & Information Technology
Location	Palo Alto, California
Req. ID	50002
Job Type	Full-time

Play and Learn: Using Video Games to Train Computer Vision Models

Alireza Shafaei
http://cs.ubc.ca/~shafaei
James J. Little
http://cs.ubc.ca/~little
Mark Schmidt
http://cs.ubc.ca/~schmidtm

Department of Computer Science
University of British Columbia
Vancouver, Canada

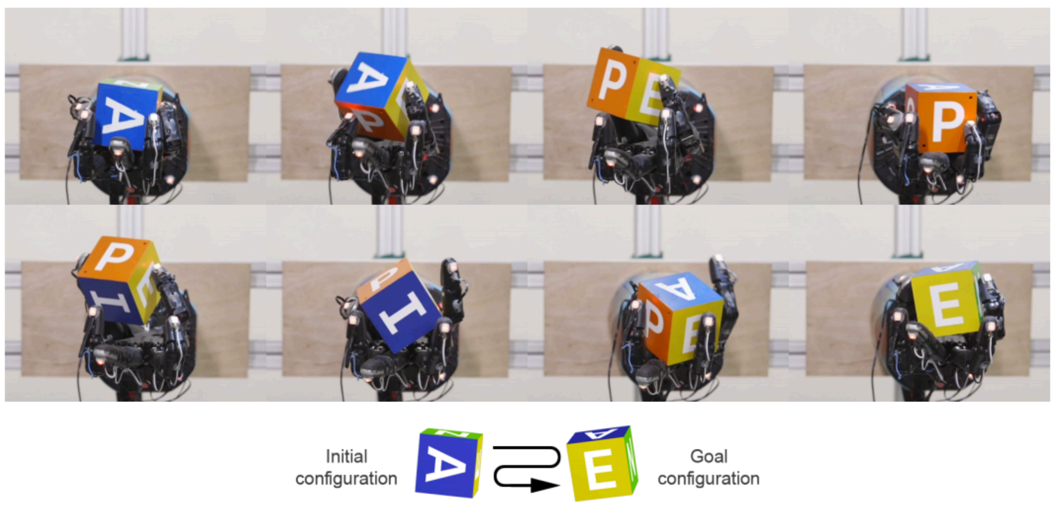





DARPA Virtual Robotics Challenge results
by Andra Keay
Research-Innovation
Articles

Learning Dexterous In-Hand Manipulation

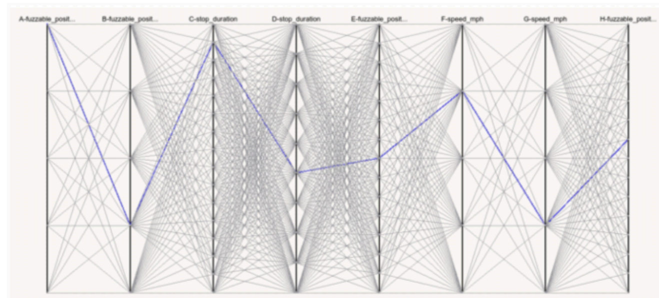
OpenAI,* Marcin Andrychowicz, Bowen Baker, Maciek Chociej,
Rafał Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron,
Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor,
Josh Tobin, Peter Welinder, Lilian Weng, Wojciech Zaremba



SIMULATIONS: THE GOOD

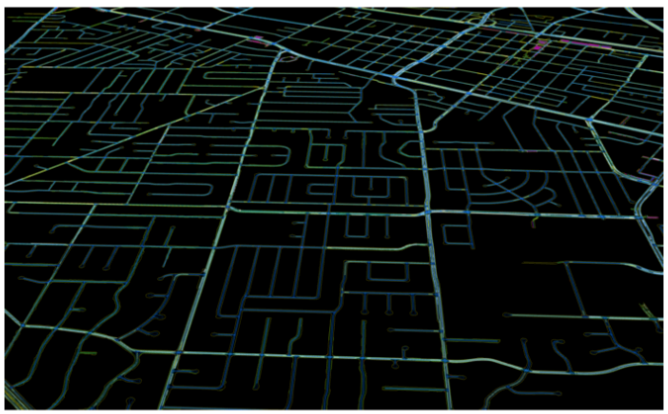
- can collect massive experience data
- no safety concerns
- labeled
- faster than real-time (simulation speed, parallelism)
- easy to reset to initial state

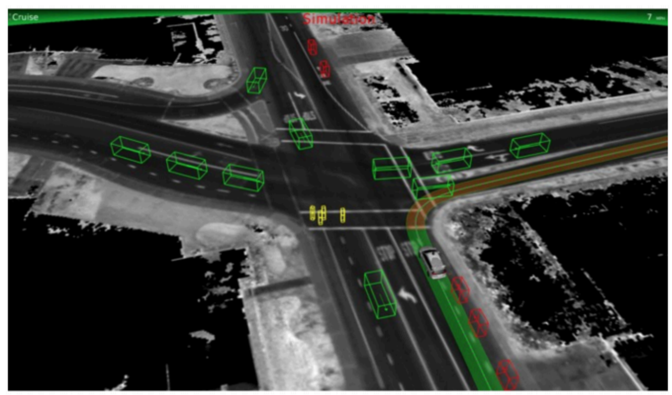
Simulation for testing autonomous vehicles



A Carcraft "fuzzing" chart (Waymo)

1M miles on the road

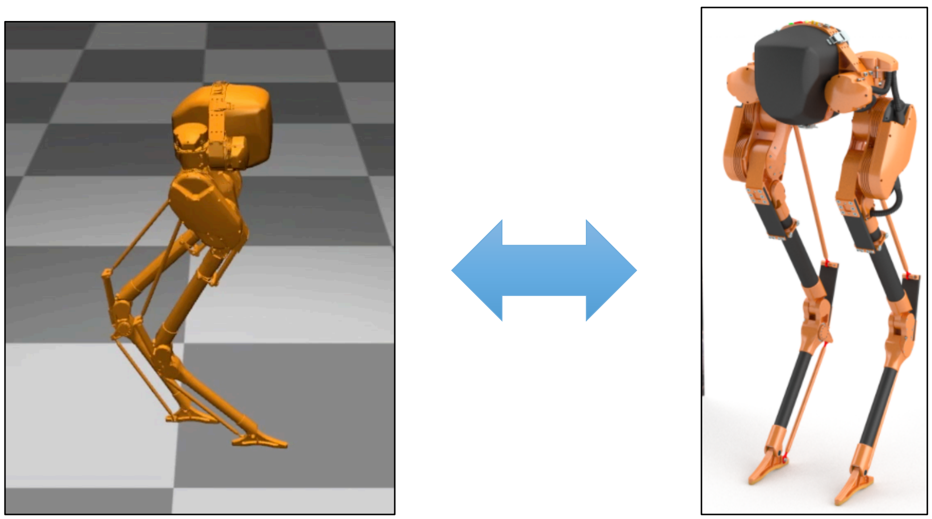




1B miles in simulation

https://www.theatlantic.com/technology/archive/2017/08/inside-waymos-secret-testing-and-simulation-facilities/537648/

CASSIE BIPEDAL ROBOT



key issues for successful sim-to-real:

- good model (SysID)
- good policy architecture
- train using state estimator
- train with latency
- train for a robust policy

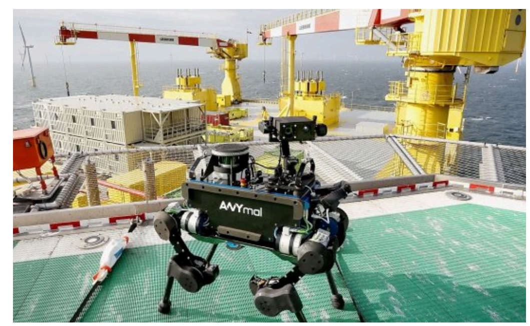
ANYMAL

SCIENCE ROBOTICS | RESEARCH ARTICLE

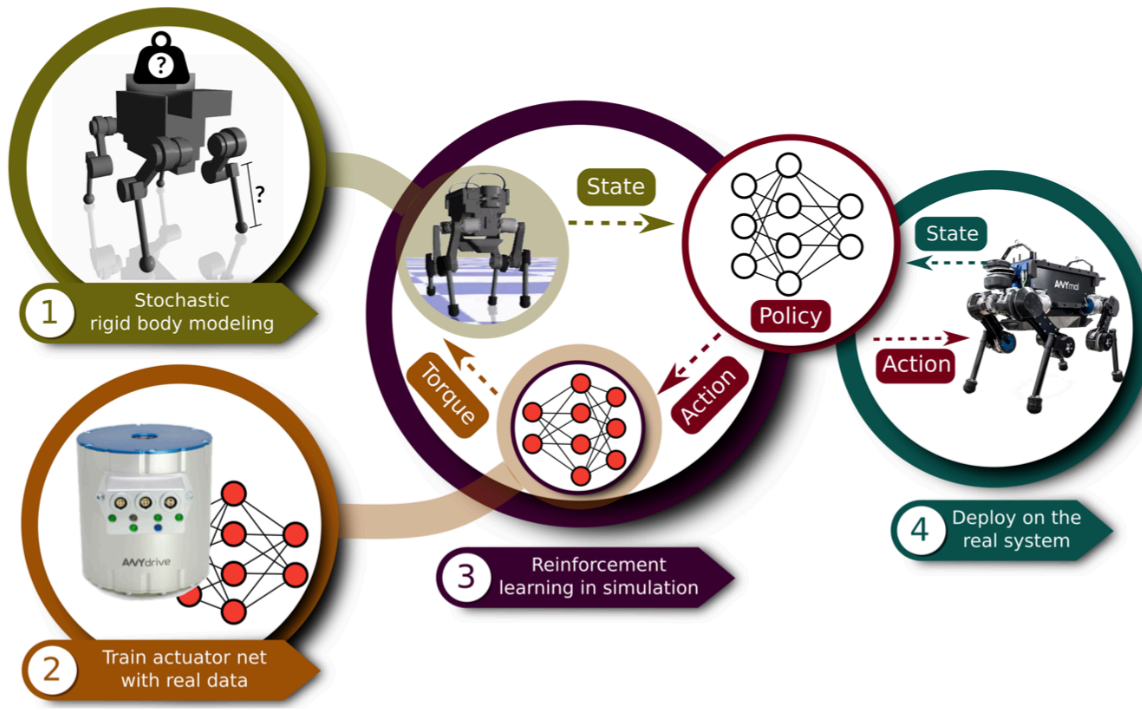
ARTIFICIAL INTELLIGENCE

Learning agile and dynamic motor skills for legged robots

Jemin Hwangbo[1]*, Joonho Lee[1], Alexey Dosovitskiy[2], Dario Bellicoso[1], Vassilios Tsounis[1], Vladlen Koltun[3], Marco Hutter[1]



The inertial properties of the links were estimated from the CAD model. We expected up to about 20% error in the estimation due to unmodeled cabling and electronics. To account for such modeling inaccuracies, we robustified the policy by training with 30 different ANYmal models with stochastically sampled inertial properties. The center of mass positions, the masses of links, and joint positions were randomized by adding a noise sampled from $U(-2, 2)$ cm, $U(-15, 15)$%, and $U(-2, 2)$ cm, respectively.



REWARDS

variable speed locomotion

angular velocity of the base cost ($c_w = -6\Delta t$)

$$c_w K(|\omega_{IB}^I - \hat{\omega}_{IB}^I|) \tag{2}$$

linear velocity of the base cost ($c_{v1} = -10\Delta t, c_{v2} = -4\Delta t$)

$$c_{v1} K(|c_{v2} \cdot (v_{IB}^I - \hat{v}_{IB}^I)|) \tag{3}$$

torque cost ($c_\tau = 0.005\Delta t$)

$$k_c c_\tau ||\tau||^2 \tag{4}$$

joint speed cost ($c_{js} = 0.03\Delta t$)

$$k_c c_{js} ||\dot{\phi}^i||^2 \quad \forall i \in \{1, 2..., 12\} \tag{5}$$

foot clearance cost ($c_f = 0.1\Delta t, \hat{p}_{f,i,z} = 0.07$ m)

$$k_c c_f (\hat{p}_{f,i,z} - p_{f,i,z})^2 ||v_{ft,i}||, \; \forall i, g_i > 0, i \in \{0, 1, 2, 3\}, \tag{6}$$

foot slip cost ($c_{fv} = 2.0\Delta t$)

$$k_c c_{fv} ||v_{ft,i}||, \; \forall i, g_i = 0, i \in \{0, 1, 2, 3\} \tag{7}$$

orientation cost ($c_o = 0.4\Delta t$)

$$k_c c_o ||[0, 0, -1]^T - \phi_g|| \tag{8}$$

smoothness cost ($c_s = 0.5\Delta t$)

$$k_c c_s ||\tau_{t-1} - \tau_t||^2 \tag{9}$$

get-up

torque cost ($c_\tau = 0.0005\Delta t$)

$$k_c c_\tau ||\tau||^2 \tag{10}$$

joint speed cost ($c_{js} = 0.2\Delta t$, $c_{jsmax} = 8$ rad/s)

$$\text{If } |\dot{\phi}^i| > |c_{jsmax}|, \quad k_c c_{js} ||\dot{\phi}^i||^2 \quad \forall i \in \{1, 2...12\} \tag{11}$$

joint acceleration cost ($c_{ja} = 0.0000005\Delta t$)

$$k_c c_{ja} ||\ddot{\phi}^i||^2 \quad \forall i \in \{1, 2...12\} \tag{12}$$

HAA cost ($c_{HAA} = 6.0\Delta t$)

$$\text{If } |\phi_{roll}| < 0.25\pi, \quad k_c c_{HAA} K(\texttt{angleDiff}(\phi^{HAA}, 0)) \tag{13}$$

HFE cost ($c_{HFE} = 7.0\Delta t$, $\hat{\phi}^{HFE} = \pm 0.5\pi$ rad (+ for right legs))

$$\text{If } |\phi_{roll}| < 0.25\pi, \quad k_c c_{HFE} K(\texttt{angleDiff}(\phi^{HFE}, \hat{\phi}^{HFE})) \tag{14}$$

KFE cost ($c_{KFE} = 7.0\Delta t$, $\hat{\phi}^{KFE} = \mp 2.45$ rad)

$$\text{If } |\phi_{roll}| < 0.25\pi, \quad k_c c_{KFE} K(\texttt{angleDiff}(\phi^{KFE}, \hat{\phi}^{KFE})) \tag{15}$$

contact slip cost ($c_{cv} = 6.0\Delta t$)

$$k_c c_{cv} \frac{\sum_{n \in I_c} ||v_{c,n}^I||^2}{|I_c|} \tag{16}$$

body contact impulse cost ($c_{cimp} = 6.0\Delta t$)

$$k_c c_{cimp} \frac{\sum_{n \in I_c \setminus I_{c,f}} ||i_{c,n}^I||}{|I_c| - |I_{c,f}|} \tag{17}$$

internal contact cost ($c_{cint} = 6.0\Delta t$)

$$k_c c_{cint} |I_{c,i}| \tag{18}$$

orientation cost ($c_o = 6.0\Delta t$)

$$c_o ||[0, 0, -1]^T - \phi_g||^2 \tag{19}$$

smoothness cost ($c_s = 0.0025\Delta t$)

$$k_c c_s ||\tau_{t-1} - \tau_t||^2 \tag{20}$$

SIMULATIONS: THE BAD

- dynamics is hard
  - modeling mismatch: kinematic & dynamic parameters
  - unmodeled aspects: contact & friction, bending, non-stationary
- sensor modeling is hard
  - images, tactile sensing, LIDAR
- reward estimation
- latency

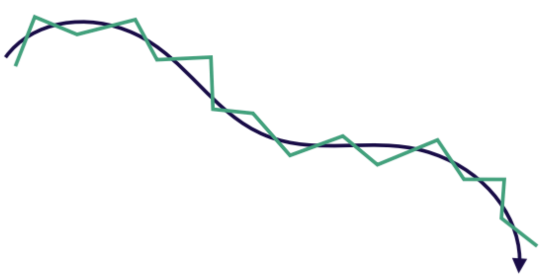
Neural nets overfit to tiny differences in data distribution

Virtual KITTI Dataset
Multi-object tracking accuracy:
Sim: 63.7%
Real: 78.1%

Virtual Worlds as Proxy for Multi-Object Tracking Analysis
[Gaidon*, Wang*, Cabon, Vig, 2016]

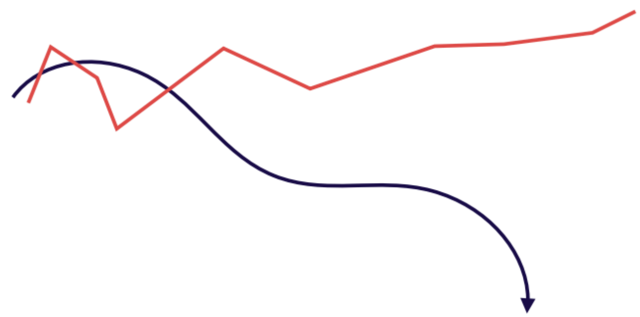
Errors compound

What we hope happens



Uncorrelated errors

What actually happens



Compounding errors

SYSTEM IDENTIFICATION

- invest effort in building a good model
- given a motion, can we estimate the dynamics parameters?
- the choice of motion matters!
  - active interventions vs passive observations
- non-stationary dynamics

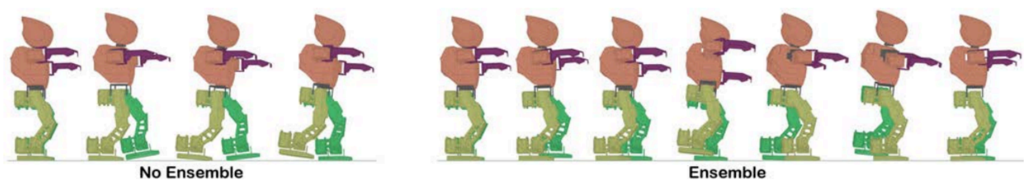
- sysID → simulation model
- online sysID

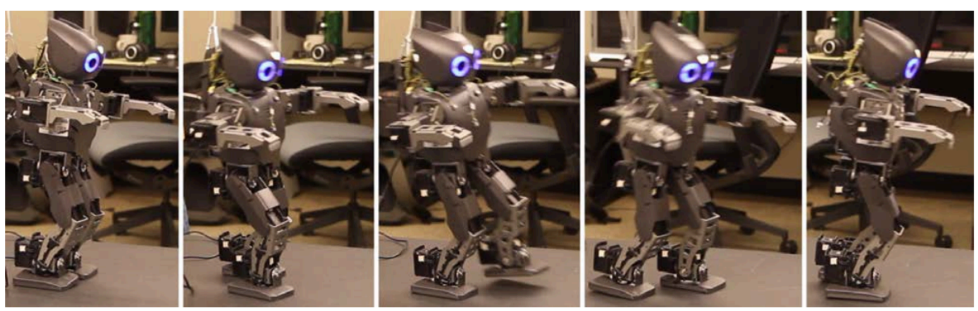
DOMAIN RANDOMIZATION

Train a single policy that is robust to moderate parameter variations.

Ensemble-CIO: Full-Body Dynamic Motion Planning that Transfers to Physical Humanoids

Igor Mordatch, Kendall Lowrey, Emanuel Todorov
Department of Computer Science & Engineering, University of Washington





Video

Preparing for the Unknown: Learning a Universal Policy with Online System Identification

Wenhao Yu[1], Jie Tan[2], C. Karen Liu[1], and Greg Turk[1]

wenhaoyu@gatech.edu, jietan@google.com, karenliu@cc.gatech.edu, turk@cc.gatech.edu
[1]Interactive Computing, Georgia Institute of Technology, USA
[2]Google Brain, Google, USA

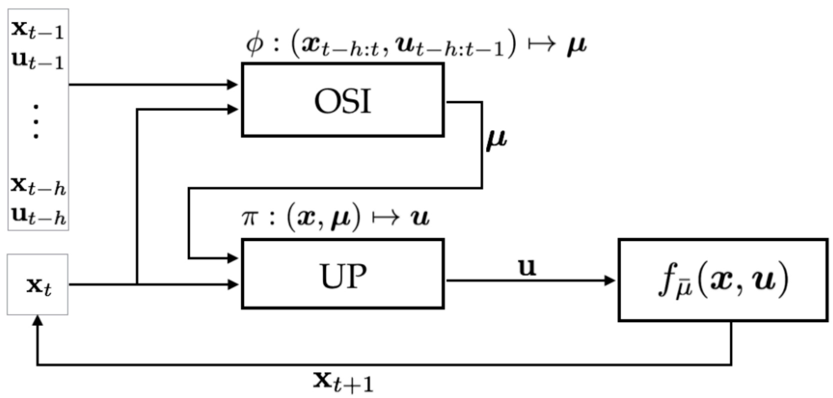


Fig. 1. Overview of UP-OSI. The online system identification model (OSI) takes as input the recent history of the motion and identify the model parameters $\boldsymbol{\mu}$. The universal control policy (UP) then takes the predicted model parameters along with the current state $\mathbf{x}$ to compute the optimal control $\mathbf{u}$.

Note: training requires experiences that include the "parameter expectation" mismatches.

Video

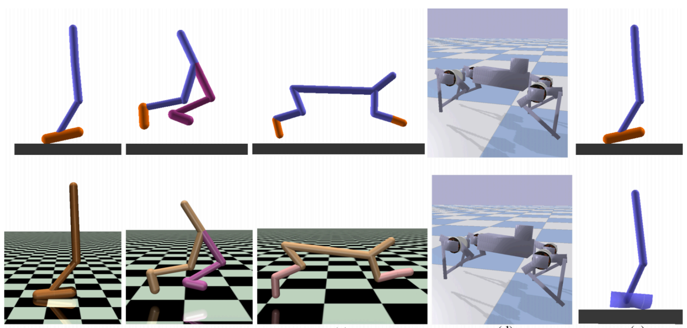
Policy Transfer with Strategy Optimization

Wenhao Yu & C. Karen Liu & Greg Turk
School of Interactive Computing
Georgia Institute of Technology, GA
wyu68@gatech.edu, {karenliu,turk}@cc.gatech.edu

To do adaptation, just learn the Universal Policy, and then
directly search in the parmeter space, using your favourite method (CMA, Bayesian optimization)

$$\mu^* = \arg\max_{\mu} J_{\mathcal{M}^t}(\pi_\mu)$$

Sim-to-Sim



(a) (b) (c) (d) (e)

Figure 1: The environments used in our experiments. Environments in the top row are source environments and environments in the bottom row are the target environments we want to transfer the policy to. (a) Hopper from DART to MuJoCo. (b) Walker2d from DART to MuJoCo with latency. (c) HalfCheetah from DART to MuJoCo with latency. (d) Minitaur robot from inaccurate motor modeling to accurate motor modeling. (e) Hopper from rigid to soft foot.

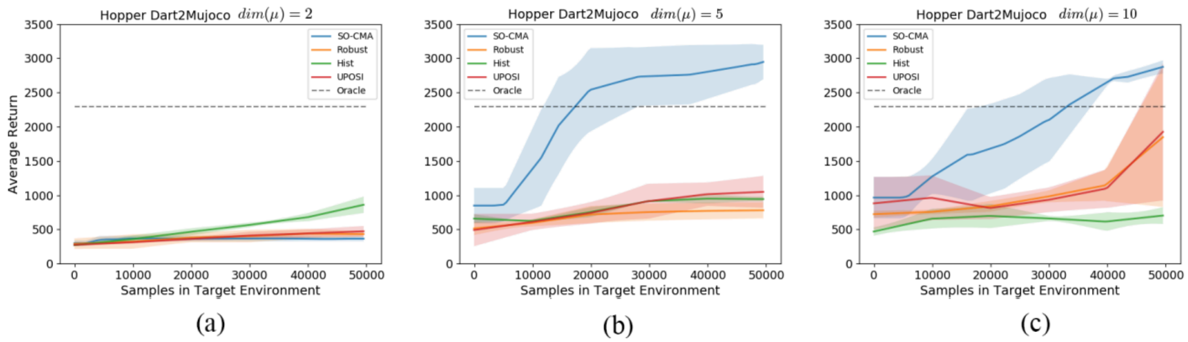


Figure 2: Transfer performance vs Sample number in target environment for the Hopper example. Policies are trained to transfer from DART to MuJoCo.

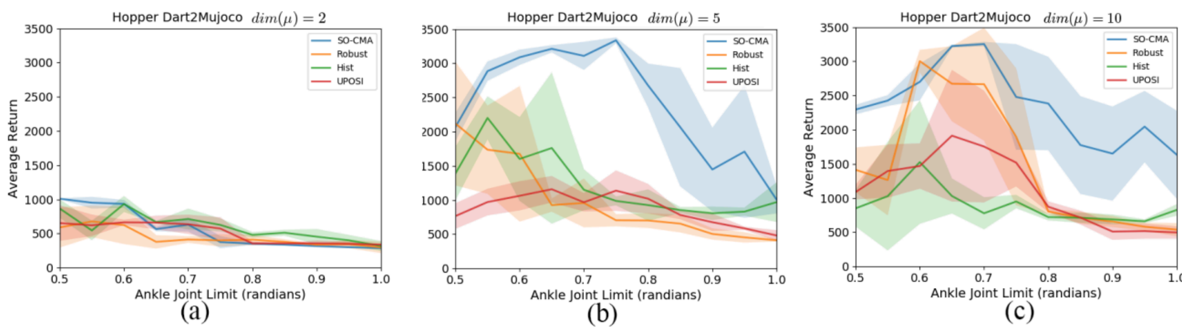
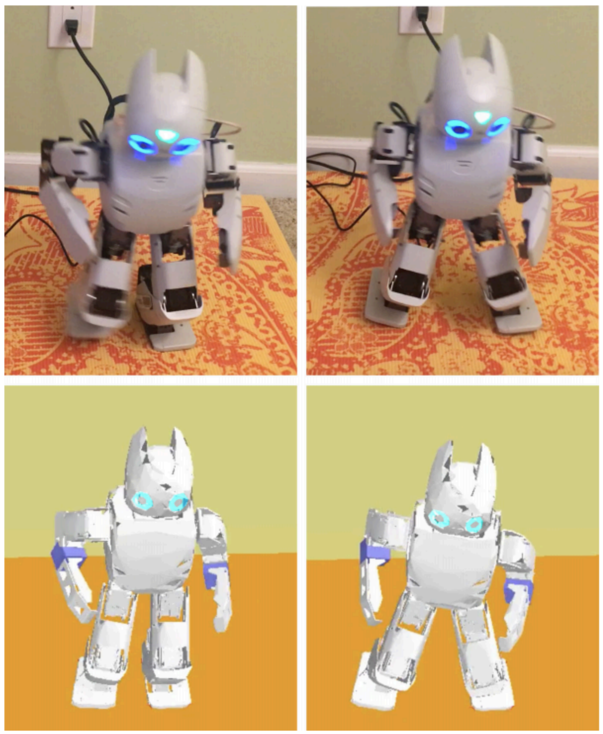
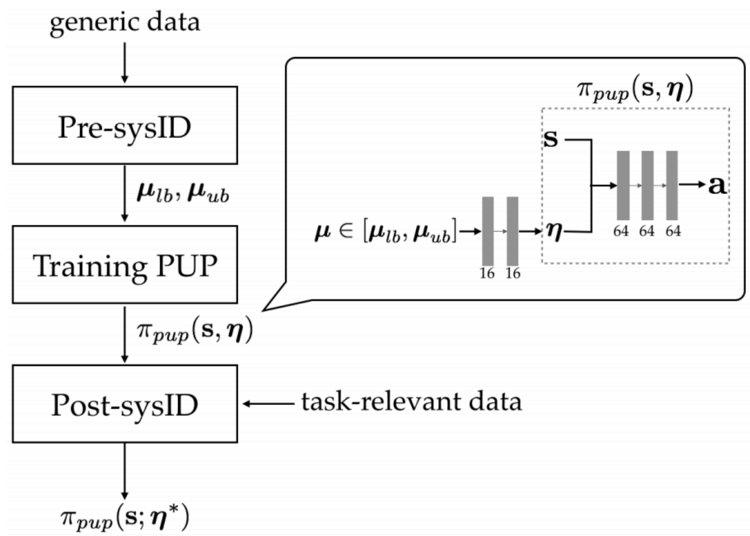


Figure 3: Transfer performance for the Hopper example. Policies are traiend to transfer from DART to MuJoCo with different ankle joint limits (horizontal axis). All trials run with total sample number of $30,000$ in the target environment.

Sim-to-Real Transfer for Biped Locomotion

Wenhao Yu[1], Visak CV Kumar[1], Greg Turk[1], C. Karen Liu[1]





Domain Randomization

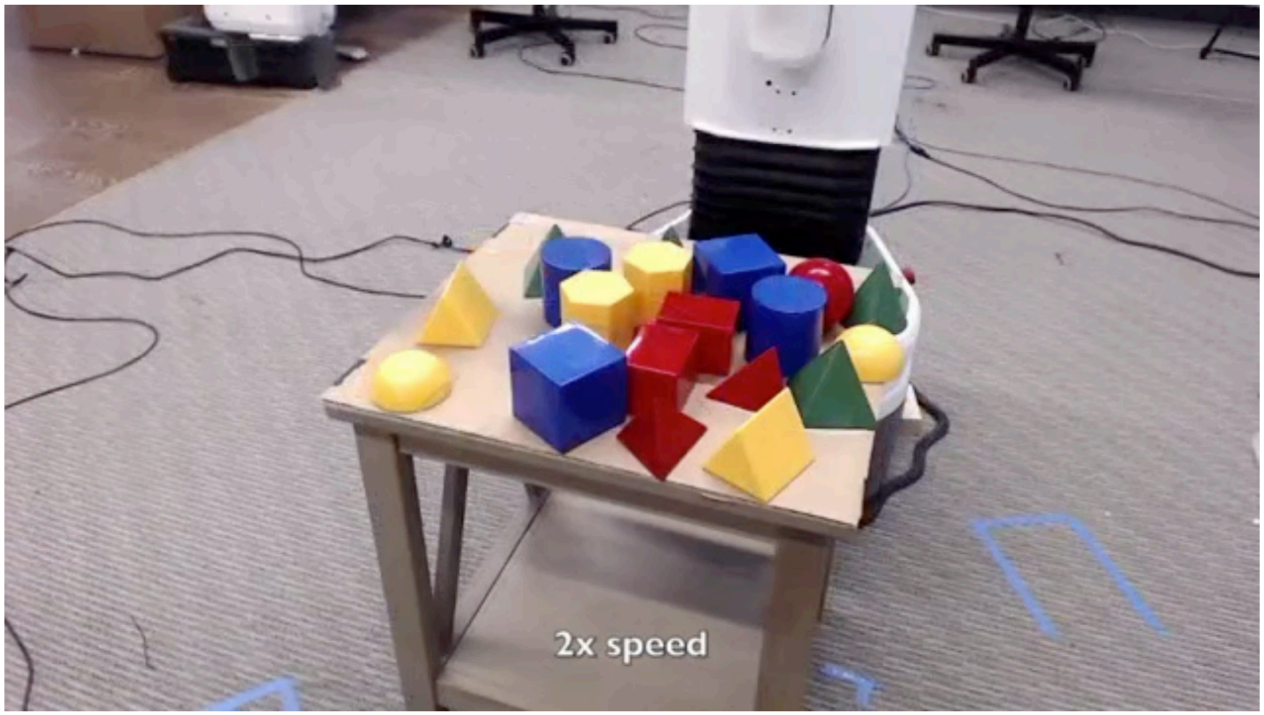
If the model sees enough simulated variation, the real world may look like just the next simulator

Domain randomization for vision: pose estimation

- Each scene has a unique set of randomizations, including:
  - Texture & material properties of all objects, table, background, robot
  - Position of cameras (within a small range)
  - Lighting position, orientation, color, and specular properties
  - Distractor objects in the scene

Domain randomization for transferring deep neural networks from simulation to the real world.
[Josh Tobin et al, 2017]

Grasping using a sim2real-trained pose estimator



Domain randomization for transferring deep neural networks from simulation to the real world.
[Josh Tobin et al, 2017]

DR for dynamics

- Standard RL: train a feedforward neural network policy in a single best environment
- Instead: train a **recurrent** network
- For each rollout, sample a different set of physics parameters

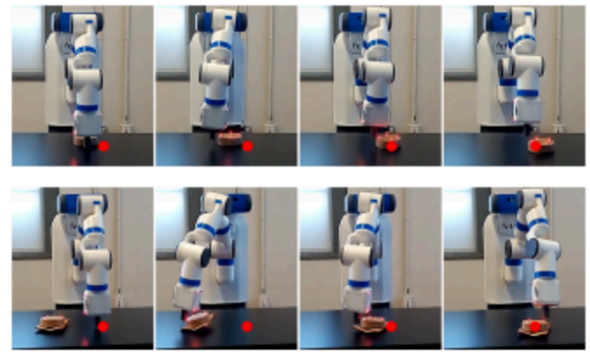


Fig. 3. LSTM policy deployed on the Fetch arm. **Bottom:** The contact dynamics of the puck was modified by attaching a packet of chips to the bottom.

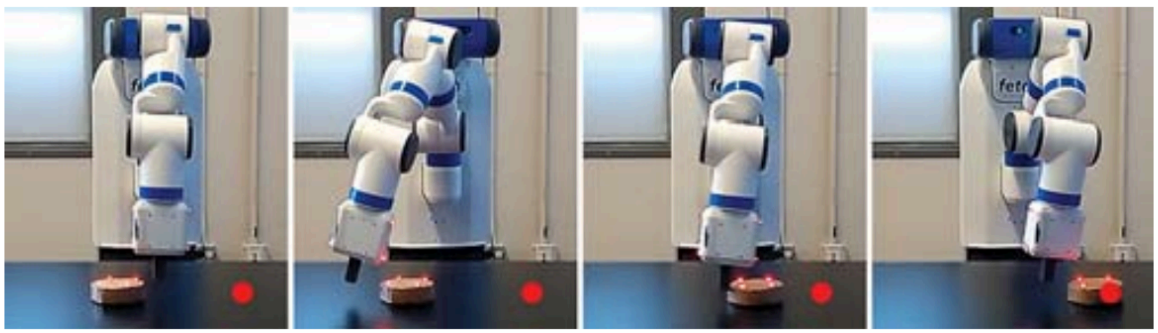
Parameter	Range
Link Mass	$[0.25, 4] \times$ default mass of each link
Joint Damping	$[0.2, 20] \times$ default damping of each joint
Puck Mass	$[0.1, 0.4] kg$
Puck Friction	$[0.1, 5]$
Puck Damping	$[0.01, 0.2] Ns/m$
Table Height	$[0.73, 0.77] m$
Controller Gains	$[0.5, 2] \times$ default gains
Action Timestep λ	$[125, 1000] s^{-1}$

Sim-to-real transfer of robotic control with dynamics randomization
[Peng et al, 2018]

2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Sim-to-Real Transfer of Robotic Control with Dynamics Randomization

Xue Bin Peng[1,2], Marcin Andrychowicz[1], Wojciech Zaremba[1], and Pieter Abbeel[1,2]



B. State and Action

The state is represented using the joint positions and velocities of the arm, the position of the gripper, as well as the puck's position, orientation, linear and angular velocities. The combined features result in a 52D state space. Actions from the policy specify target joint angles for a position controller. Target angles are specified as relative offsets from the current joint rotations. This yields a 7D action space.

Parameter	**Range**
Link Mass	$[0.25, 4] \times$ default mass of each link
Joint Damping	$[0.2, 20] \times$ default damping of each joint
Puck Mass	$[0.1, 0.4] kg$
Puck Friction	$[0.1, 5]$
Puck Damping	$[0.01, 0.2] Ns/m$
Table Height	$[0.73, 0.77] m$
Controller Gains	$[0.5, 2] \times$ default gains
Action Timestep λ	$[125, 1000] s^{-1}$

TABLE I

DYNAMICS PARAMETERS AND THEIR RESPECTIVE RANGES.

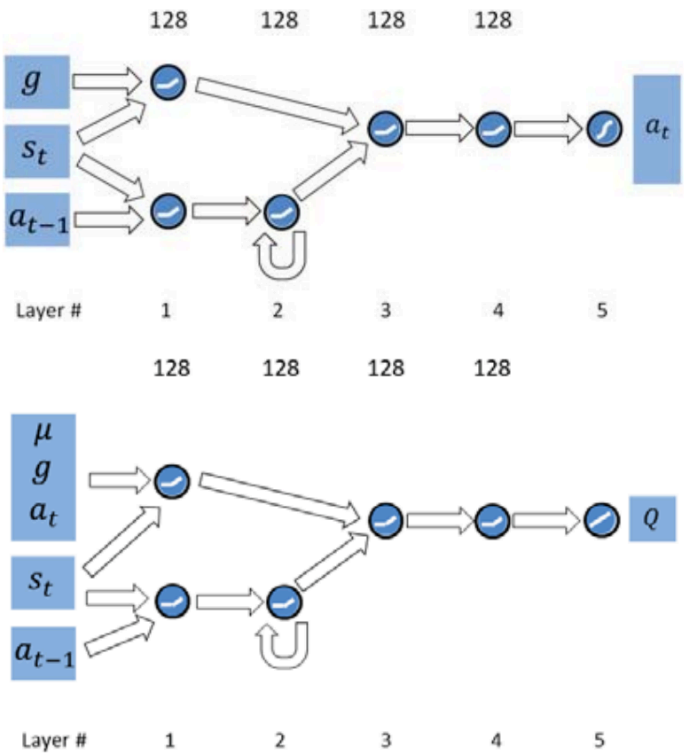


Fig. 4. Schematic illustrations of the policy network **(top)**, and value network **(bottom)**. Features that are relevant for inferring the dynamics of the environment are processed by the recurrent branch, while the other inputs are processed by the feedforward branch.

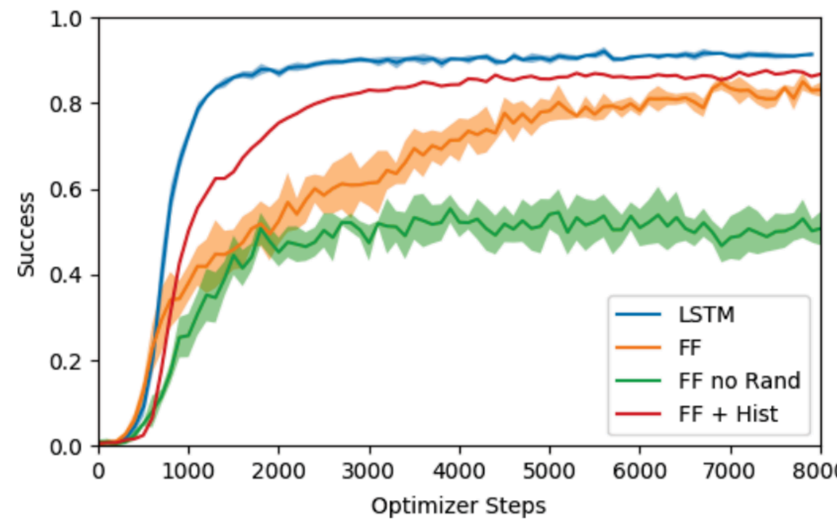
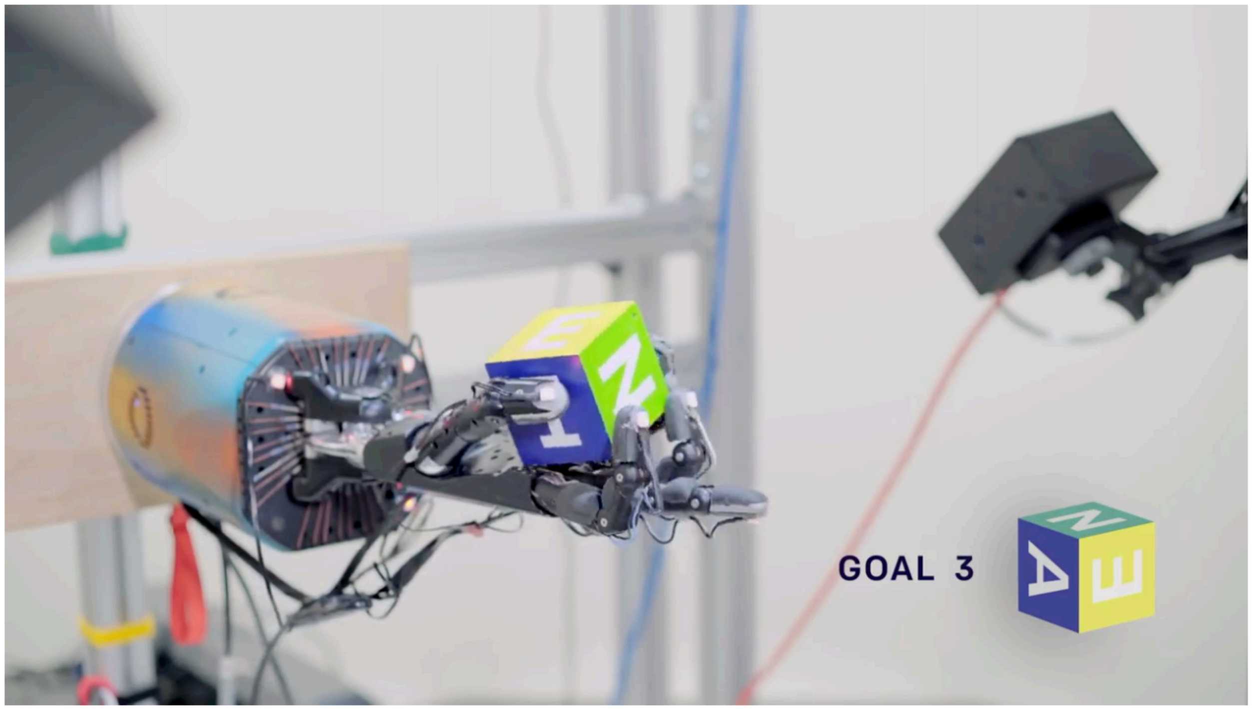


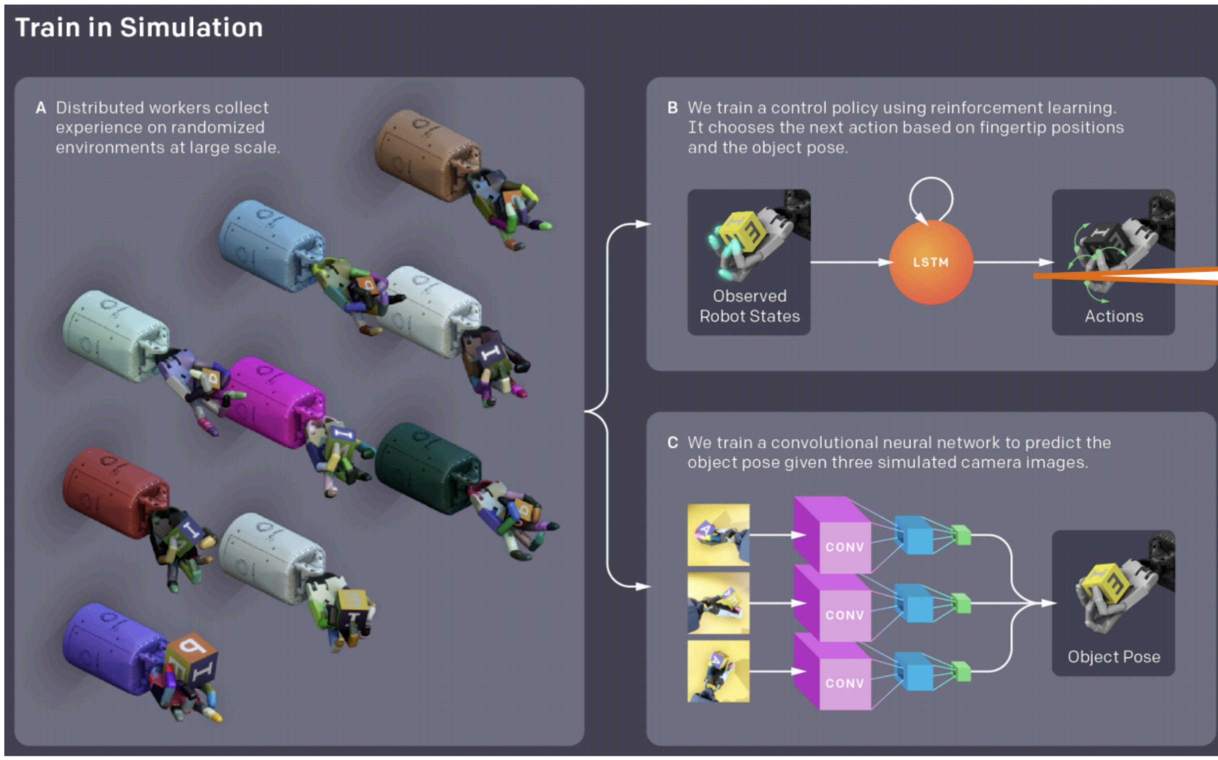
Fig. 6. Learning curves of different network architectures. Four policies are trained for each architecture with different random seeds. Performance is evaluated over 100 episodes in simulation with random dynamics.

In-hand manipulation



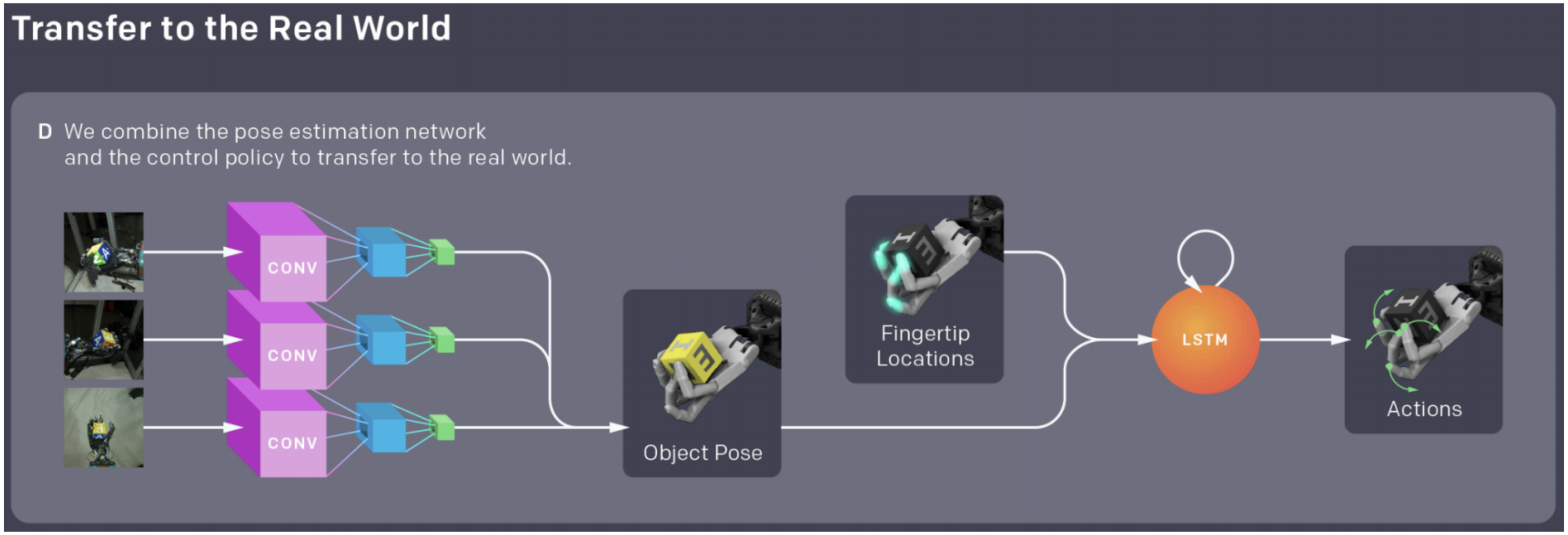
Learning dextrous in-hand manipulation
[OpenAI Robotics, 2018]

How does it work?



Trained using PPO

Learning dextrous in-hand manipulation
[OpenAI Robotics, 2018]



Learning dextrous in-hand manipulation
[OpenAI Robotics, 2018]

Table 1: Ranges of physics parameter randomizations.

Parameter	**Scaling factor range**	**Additive term range**
object dimensions	$\text{uniform}([0.95, 1.05])$	
object and robot link masses	$\text{uniform}([0.5, 1.5])$	
surface friction coefficients	$\text{uniform}([0.7, 1.3])$	
robot joint damping coefficients	$\text{loguniform}([0.3, 3.0])$	
actuator force gains (P term)	$\text{loguniform}([0.75, 1.5])$	
joint limits		$\mathcal{N}(0, 0.15)$ rad
gravity vector (each coordinate)		$\mathcal{N}(0, 0.4)\ \text{m/s}^2$

Table 9: Vision randomizations.

Randomization type	**Range**
number of cameras	3
camera position	$\pm$ 1.5 mm
camera rotation	0–3° around a random axis
camera field of view	$\pm$ 1°
robot material colors	RGB
robot material metallic level	5%–25%[17]
robot material glossiness level	0%–100%[17]
object material hue	calibrated hue $\pm$ 1%
object material saturation	calibrated saturation $\pm$ 15%
object material value	calibrated value $\pm$ 15%
object metallic level	5%–15%[17]
object glossiness level	5%–15%[17]
number of lights	4–6
light position	uniform over upper half-sphere
light relative intensity	1–5
total light intensity	0–15[17]
image contrast adjustment	50%–150%
additive per-pixel Gaussian noise	$\pm$ 10%

HOW DOES DOMAIN RANDOMIZATION WORK IN PRACTICE?

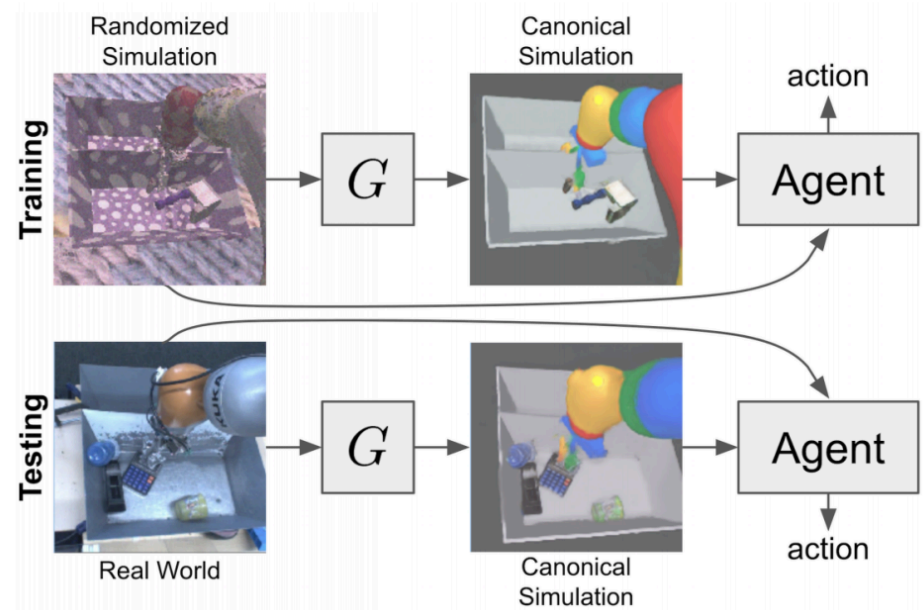
[Josh Tobin]

- Build a simulated world
- Calibrate it to the environment
- Design randomizations to “cover” real-world variability
- Train a model and evaluate in real
- Examine failure modes and add randomization

Issues

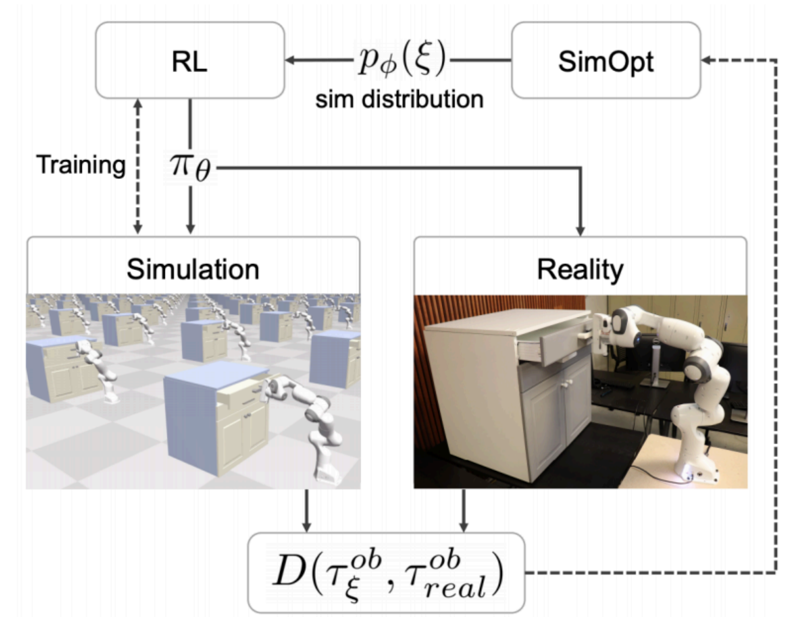
- Building simulations is manual and time consuming
- Deciding what parameters to randomize requires judgment
- Randomizing parameters as much as possible may not be optimal

Randomized-to-Canonical Adaptation Networks



Sim-to-Real via Sim-to-Sim: Data-efficient Robotic Grasping via Randomized-to-Canonical Adaptation Networks
[Stephen James et al, 2019]

SimOpt



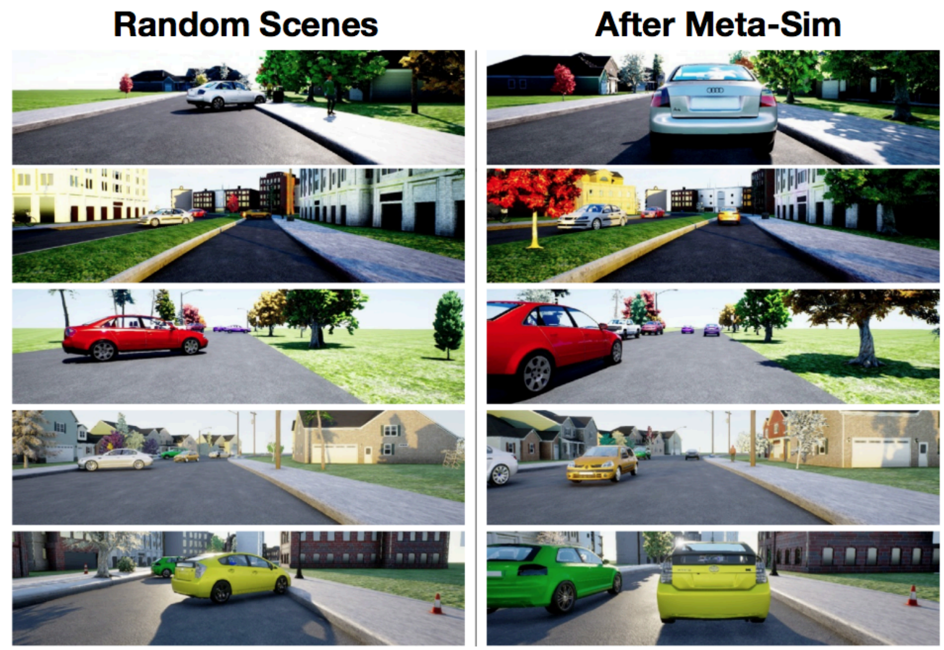
Algorithm 1 SimOpt framework

$p_{\phi_0} \leftarrow$ Initial simulation parameter distribution
$\epsilon \leftarrow$ KL-divergence step for updating p_ϕ
for iteration $i \in \{0, \ldots, N\}$ **do**
  env $\leftarrow$ Simulation(p_{ϕ_i})
  $\pi_{\theta, p_{\phi_i}} \leftarrow$ RL(env)
  $\tau_{real}^{ob} \sim$ RealRollout$(\pi_{\theta, p_{\phi_i}})$
  $\xi \sim$ Sample(p_{ϕ_i})
  $\tau_\xi^{ob} \sim$ SimRollout$(\pi_{\theta, p_{\phi_i}}, \xi)$
  $c(\xi) \leftarrow D(\tau_\xi^{ob}, \tau_{real}^{ob})$
  $p_{\phi_{i+1}} \leftarrow$ UpdateDistribution$(p_{\phi_i}, \xi, c(\xi), \epsilon)$

Closing the Sim-to-Real Loop: Adapting Simulation Randomization with Real World Experience
[Yevgen Cehbotar et al, 2019]

Meta-Sim

- Generating realistic randomization distributions is hard
- You end up with scenes like the left
- **Goal:** use some real data to make the scenes realistic



Meta-Sim: Learning to Generate Structured Datasets
[Amlan Kar et al, 2019]

Automatic Domain Randomization (ADR)

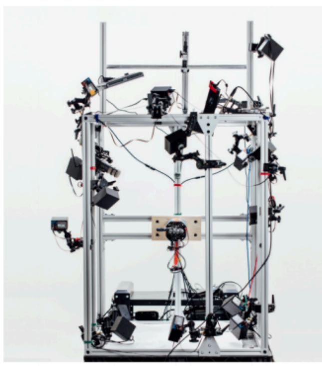
Intuition

- More randomization = better transfer, **given the same performance in sim**
- But wide randomization ranges lead to poor performance
- ADR: automatically create a curriculum of expanding randomization ranges
- How? Widen the distribution if performance is good near the boundary of the range

Solving Rubik's Cube with a Robot Hand
[OpenAI et al, 2019]

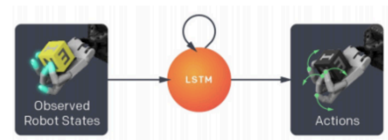
The dream: real-sim-real

Collect data about the world



Automatically design and randomize a sim

Optimize a policy in that simulation



Figures from: **Learning Dextrous In-Hand Manipulation**
[OpenAI et al, 2018]

QUESTIONS & DISCUSSION