Sim2Real

Saurabh Gupta

Solving a RL Problem

Better Reward Signals

Sim2Real

Better Optimization

Convert into a Supervised Training Problem

Solve a Related but Supervision-rich Problem

Build Models and Plan with Them

Model-free RL with sparse rewards

Known reward, known model. Model-based RL

Sim2Real (Discussion)

Train policy in simulation, and then deploy in the real world

Let's assume we want to learn policies,

- A: what are some of the advantages of sim2real?
- B: what are some of the short-comings of sim2real?

Getting Sim2Real to Work

Minimizing the domain gap

- sensing
  - data augmentation (aka domain randomization)
  - building rich simulators
  - use modalities that are easier to simulate
  - domain adaptation
- actuation
  - operating at the appropriate level of abstraction

Building Rich Simulators

Simulator based on scans of
Real World Environments

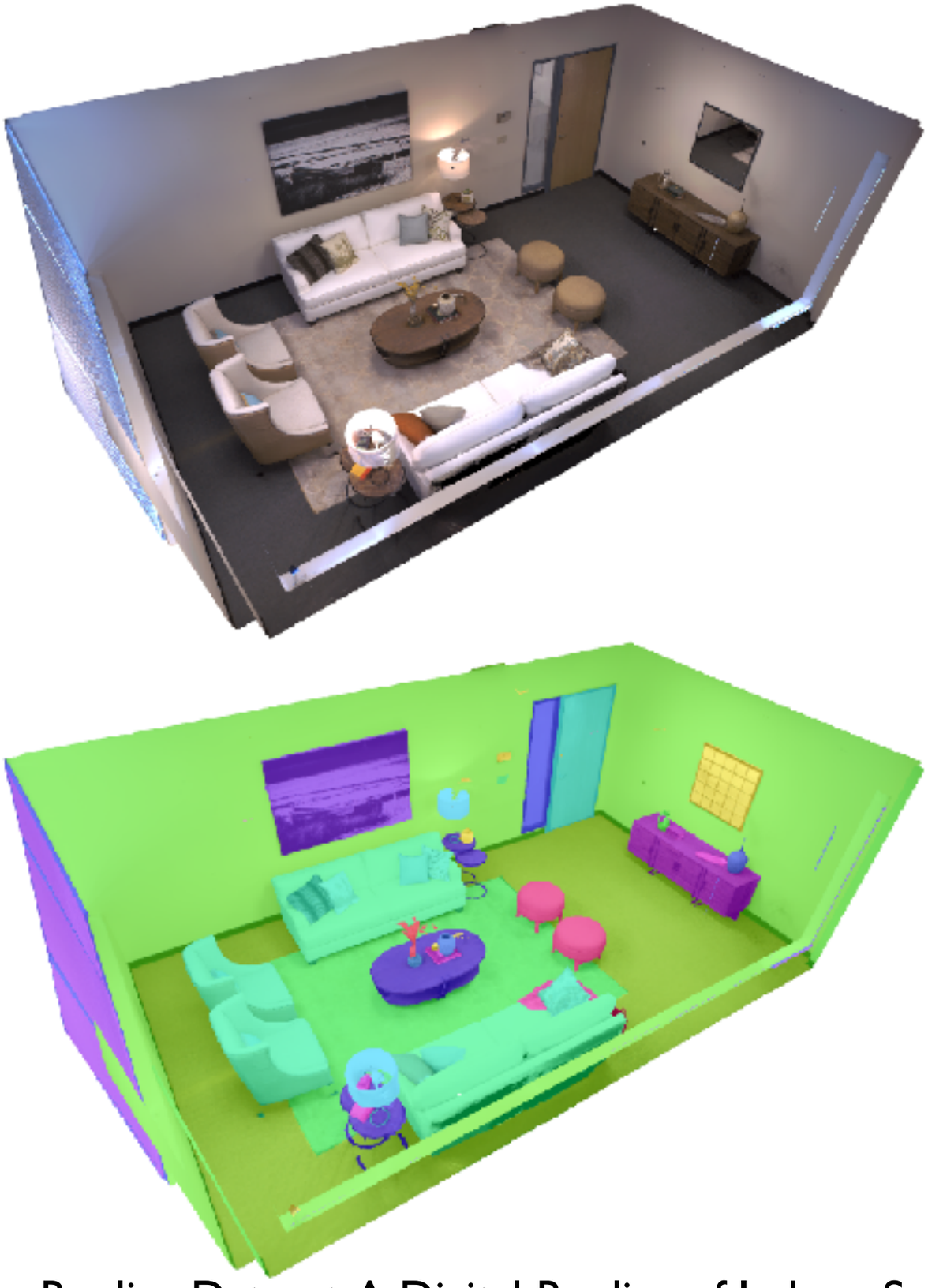
Simulate robot views
and motion

Compute ground
truth traversability

Gupta et al., CVPR 2017. Cognitive Mapping and Planning for Visual Navigation
Armeni et al. CVPR 2016. 3D Semantic Parsing of Large-Scale Indoor Spaces

Building Rich Simulators

Building Rich Simulators



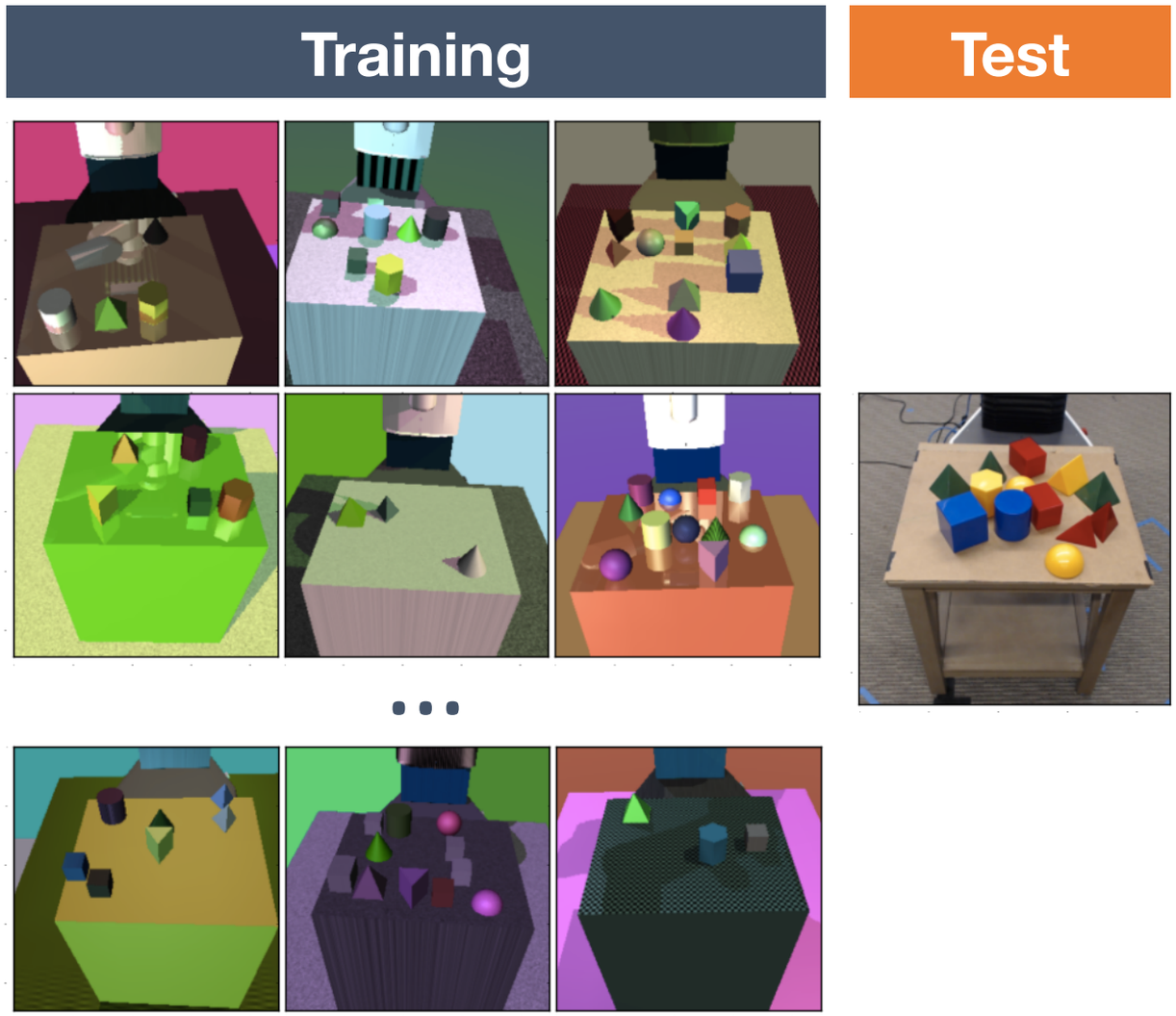
The Replica Dataset: A Digital Replica of Indoor Spaces

Data Augmentation

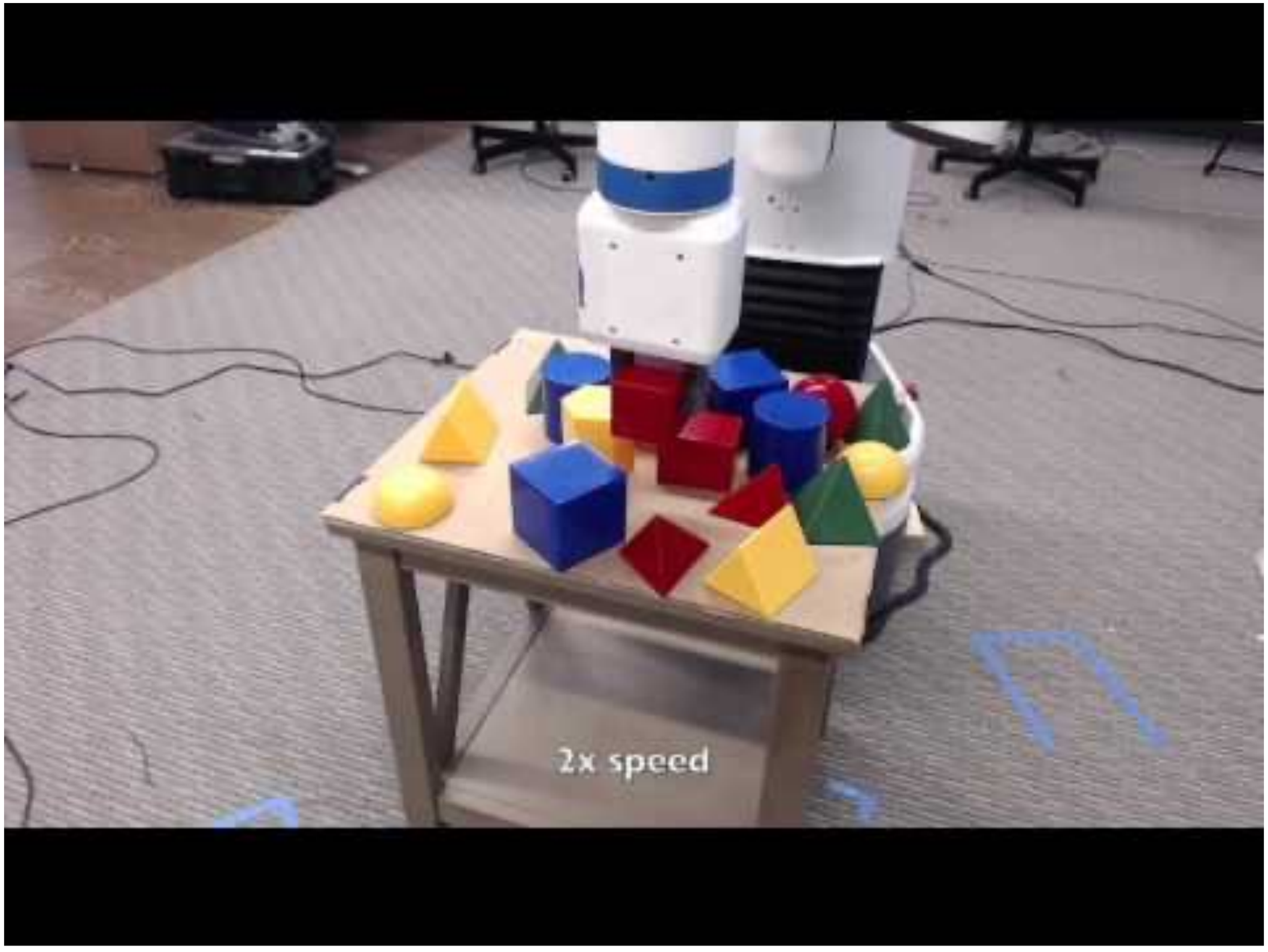
Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World

Josh Tobin[1], Rachel Fong[2], Alex Ray[2], Jonas Schneider[2], Wojciech Zaremba[2], Pieter Abbeel[3]

Demonstrate that with enough variations during training, models trained in simulation can transfer to real world for estimating object locations.



Data Augmentation

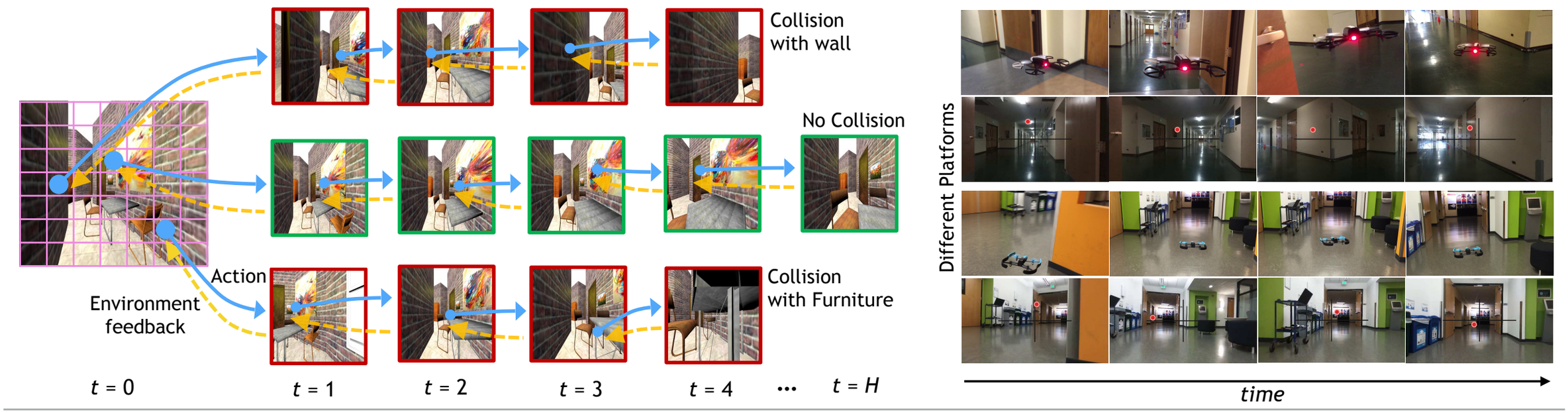


Data Augmentation

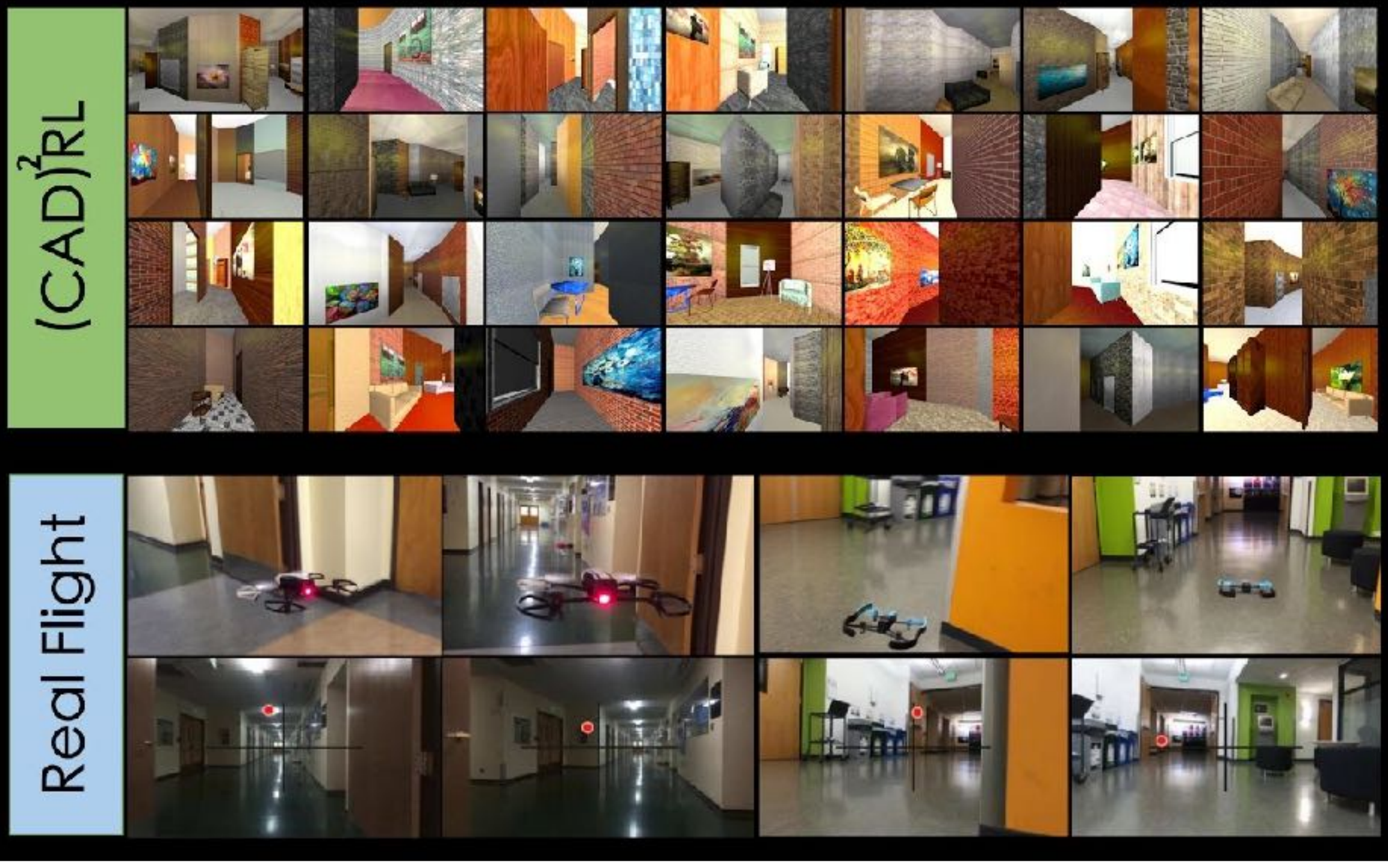
CAD^{2}RL: Real Single-Image Flight Without a Single Real Image

Fereshteh Sadeghi
University of Washington
fsadeghi@cs.washington.edu

Sergey Levine
University of California, Berkeley
svlevine@eecs.berkeley.edu

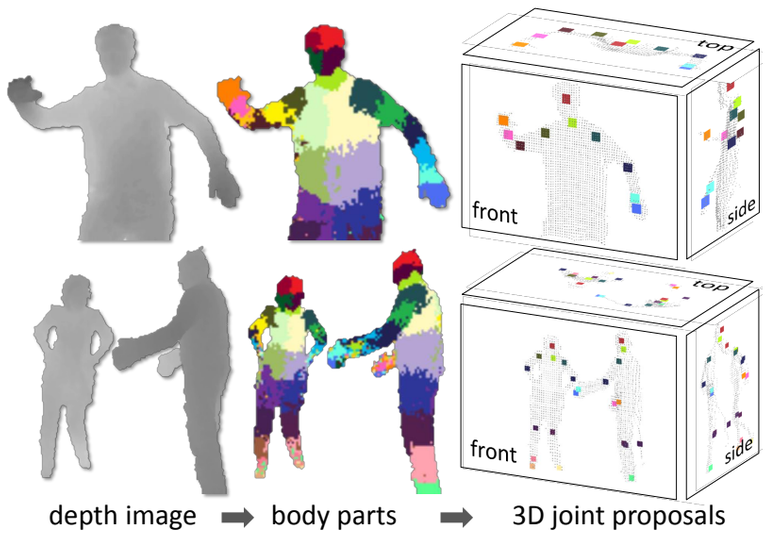


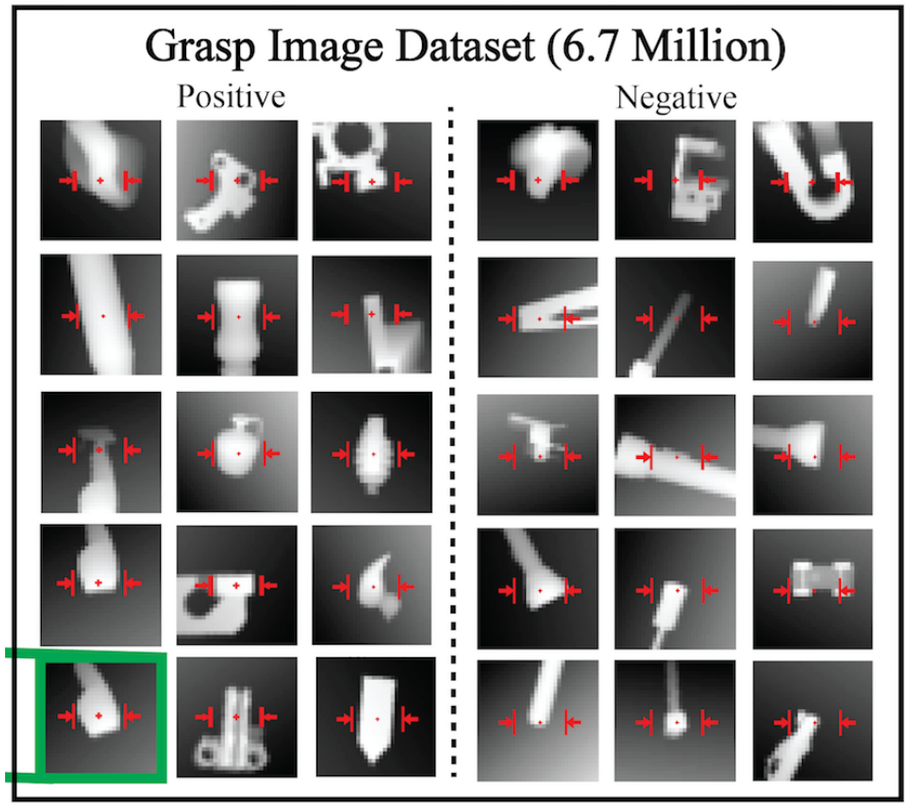
Data Augmentation



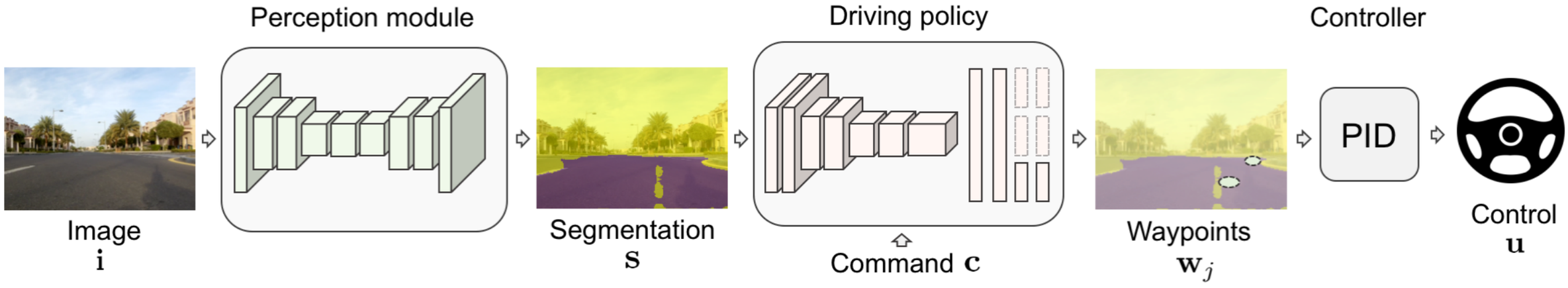
Modalities that have smaller domain gap

eg. depth





eg. semantic segmentation masks



Real-Time Human Pose Recognition in Parts from Single Depth Images. Shotton et al. CVPR 2011
DexNet 2.0. Mahler et al. 2017
Driving Policy Transfer via Modularity and Abstraction. Müller et al. CoRL 2018

Domain Adaptation

ML techniques to learn representations that are invariant to domain gap

Source image (GTA5)

Target image (CityScapes)

Pixel accuracy on target

Source-only: 54.0%

Adapted (**ours**): **83.6%**

Source images (SVHN)

Target images (MNIST)

Accuracy on target

Source-only: 67.1%

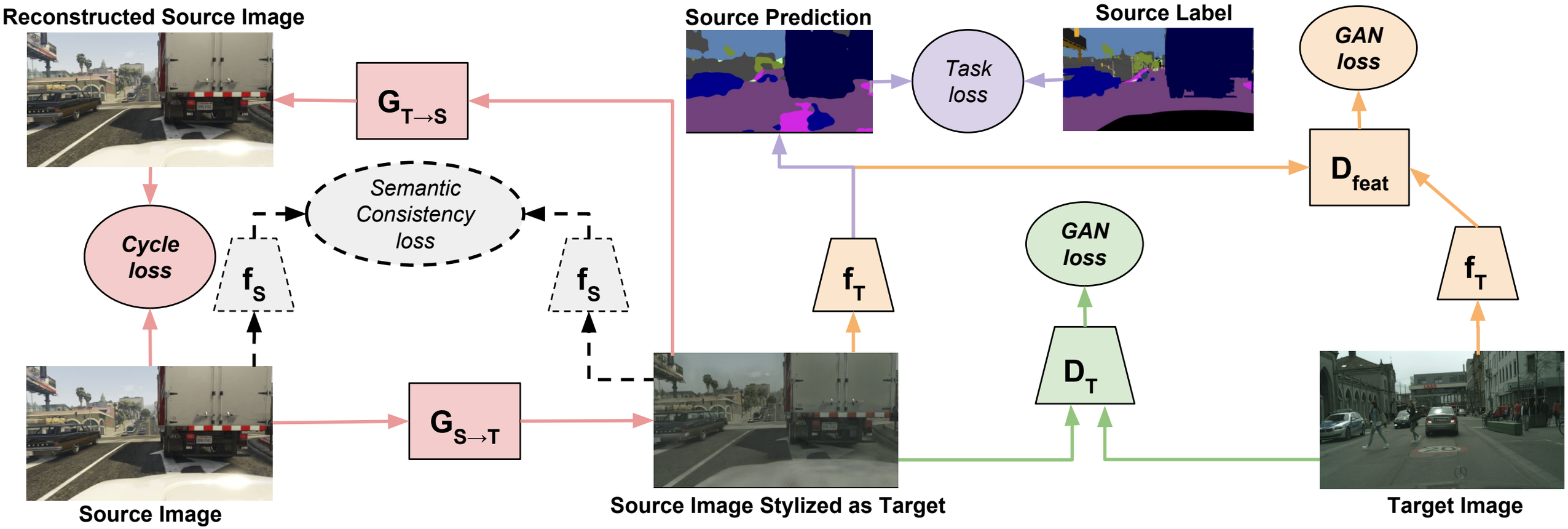
Adapted (**ours**): **90.4%**

Adversarial discriminative domain adaptation E Tzeng, J Hoffman, K Saenko, T Darrell. ICCV 2017.

Cycada: Cycle-consistent adversarial domain adaptation. J Hoffman, et al. ICML 2018.

Domain Adaptation

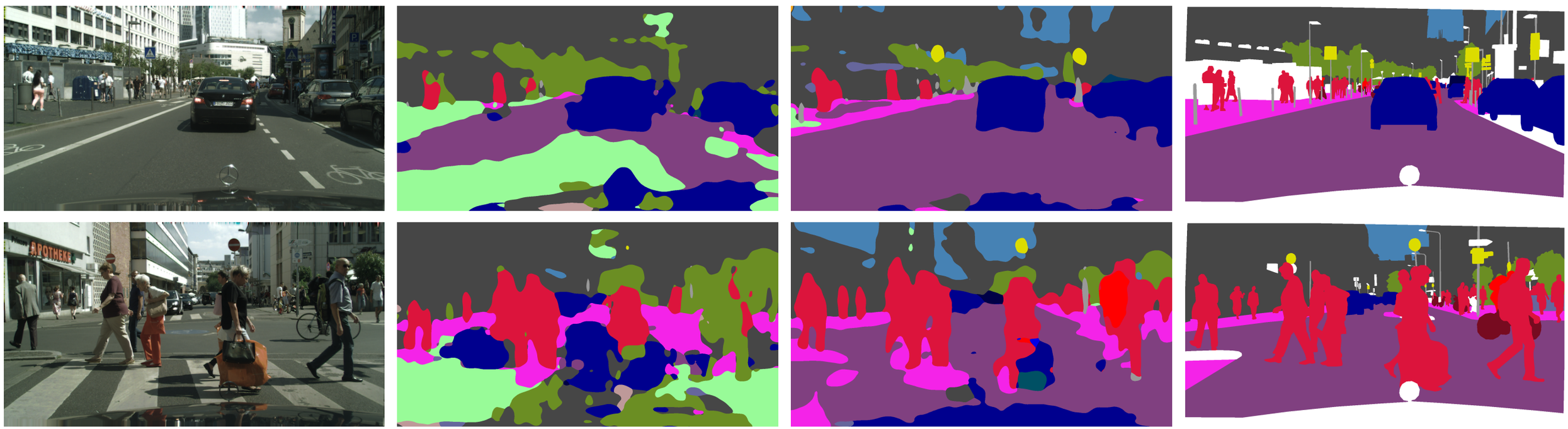
ML techniques to learn representations that are invariant to domain gap



Adversarial discriminative domain adaptation E Tzeng, J Hoffman, K Saenko, T Darrell. ICCV 2017.
Cycada: Cycle-consistent adversarial domain adaptation. J Hoffman, et al. ICML 2018.

Domain Adaptation

ML techniques to learn representations that are invariant to domain gap



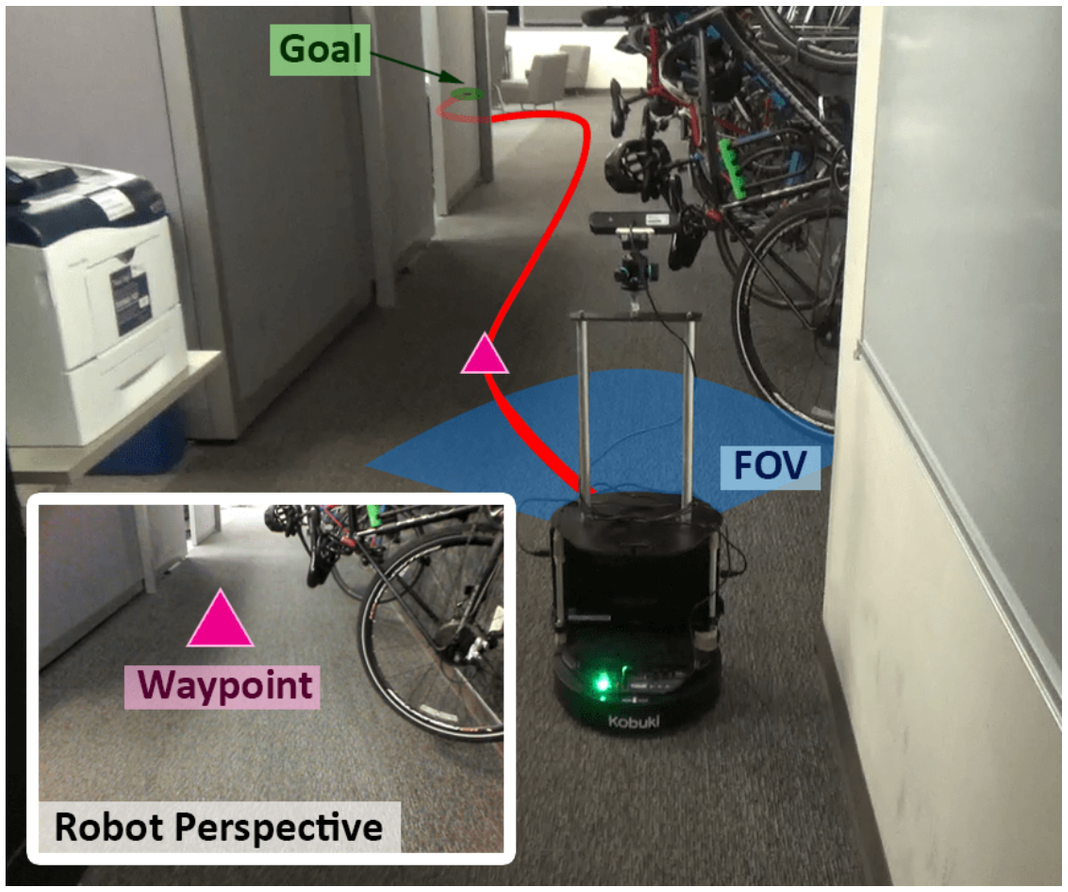
(a) Test Image (b) Source Prediction (c) CyCADA Prediction (d) Ground Truth

Adversarial discriminative domain adaptation E Tzeng, J Hoffman, K Saenko, T Darrell. ICCV 2017.
Cycada: Cycle-consistent adversarial domain adaptation. J Hoffman, et al. ICML 2018.

Operating at the right abstraction level

Combining Optimal Control and Learning for Visual Navigation in Novel Environments

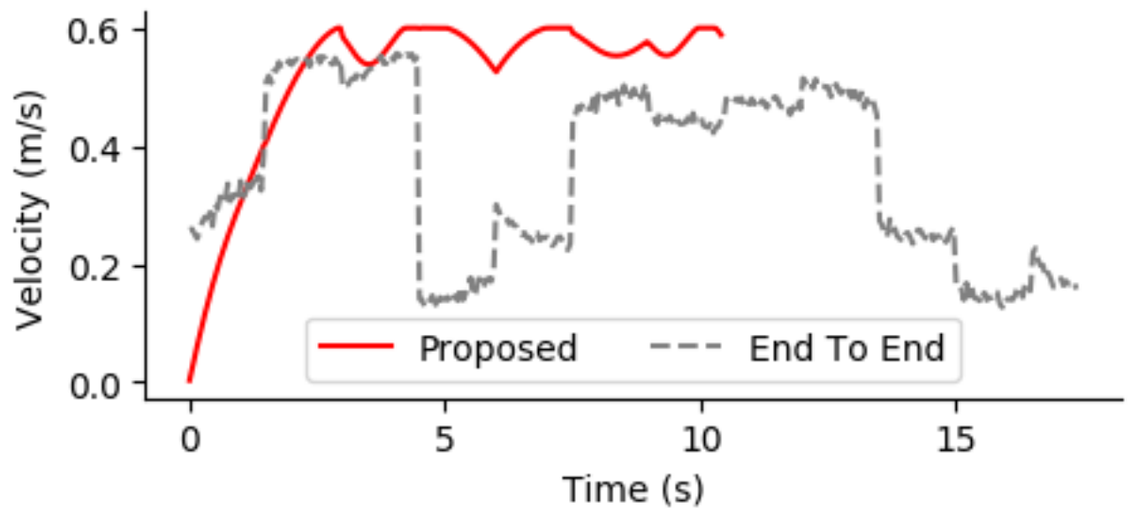
Somil Bansal[*1] **Varun Tolani**[*1] **Saurabh Gupta**[2] **Jitendra Malik**[1,2] **Claire Tomlin**[1]



Output:

a) way-points that are tracked using a LQR controller

b) angular and linear velocities



LB-WayPtNav (our)	RGB	95	22.93 ±2.38	0.09 ±0.01	3.01 ±0.38
End To End	RGB	50	33.88 ±3.01	0.19 ±0.01	6.12 ±0.18

Operating at the right abstraction level

LEARNING LOCOMOTION SKILLS USING DEEPRL: DOES THE CHOICE OF ACTION SPACE MATTER?

Xue Bin Peng & Michiel van de Panne
Department of Computer Science
University of the British Columbia
Vancouver, Canada
{xbpeng,van}@cs.ubc.ca

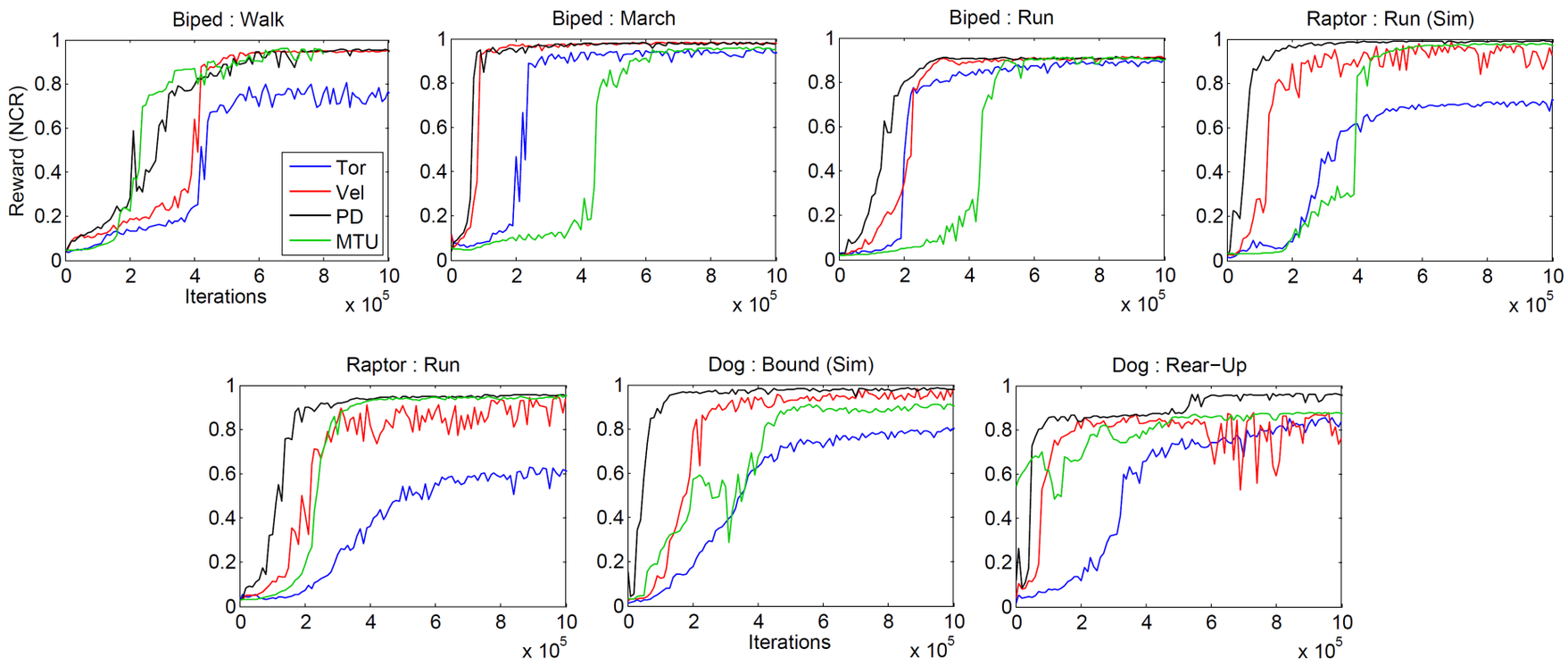
Policy action space:

a) torques

b) target joint velocities $\tau^i = k_d^i(\hat{\dot{q}}^i - \dot{q}^i)$

c) target joint angles $k_p^i(\hat{q}^i - q^i) + k_d^i(\hat{\dot{q}}^i - \dot{q}^i)$

d) muscle activations



Getting Sim2Real to Work

Minimizing the domain gap

- sensing
  - data augmentation (aka domain randomization)
  - building rich simulators
  - use modalities that are easier to simulate
  - domain adaptation
- actuation
  - operating at the appropriate level of abstraction

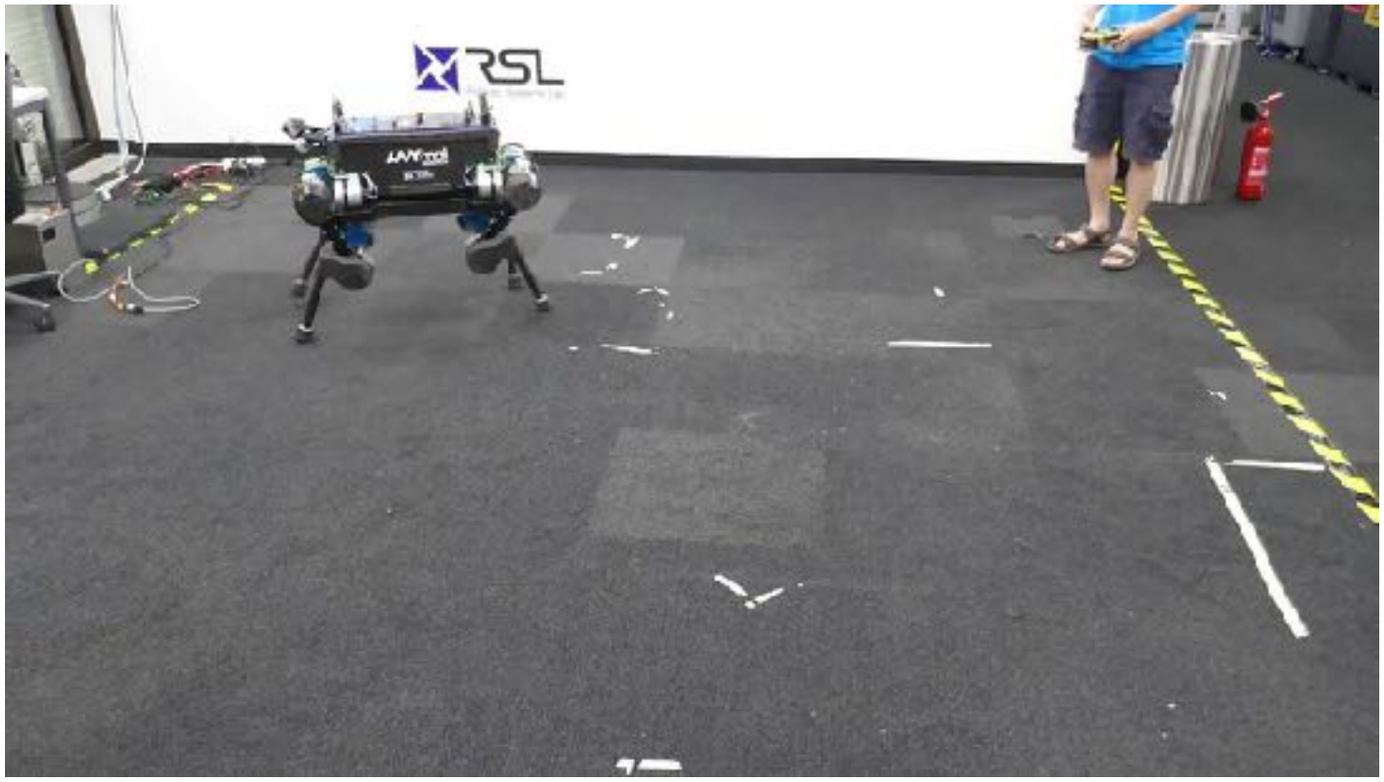
Learning Agile and Dynamic Motor Skills for Legged Robots

JEMIN HWANGBO[1*], JOONHO LEE[1], ALEXEY DOSOVITSKIY[2], DARIO BELLICOSO[1], JOONHO LEE[1], VASSILIOS TSOUNIS[1], VLADLEN KOLTUN[2], AND MARCO HUTTER[1]

[1] *Robotic Systems Lab, ETH Zurich*
[2] *Intelligent Systems Lab, Intel*
[*] *Corresponding author: jemin.hwangbo@gmail.com*

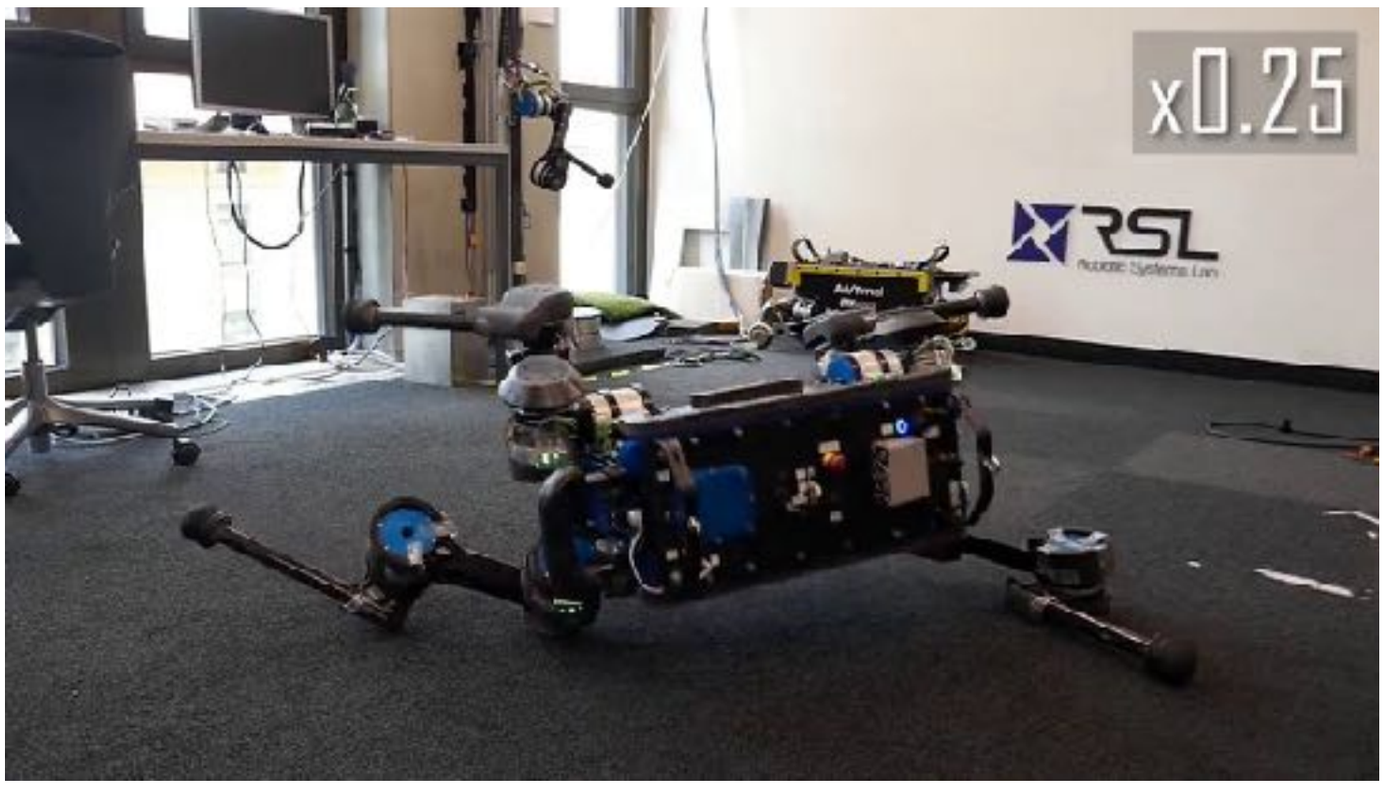
- Successful demonstration of sim2real for quadruped locomotion
- RL inside a learned simulation for successful transfer
- Case study of how to get sim2real to work in realistic scenarios
- Combining classical physics contact dynamics with learned models for actuation

ANYmal

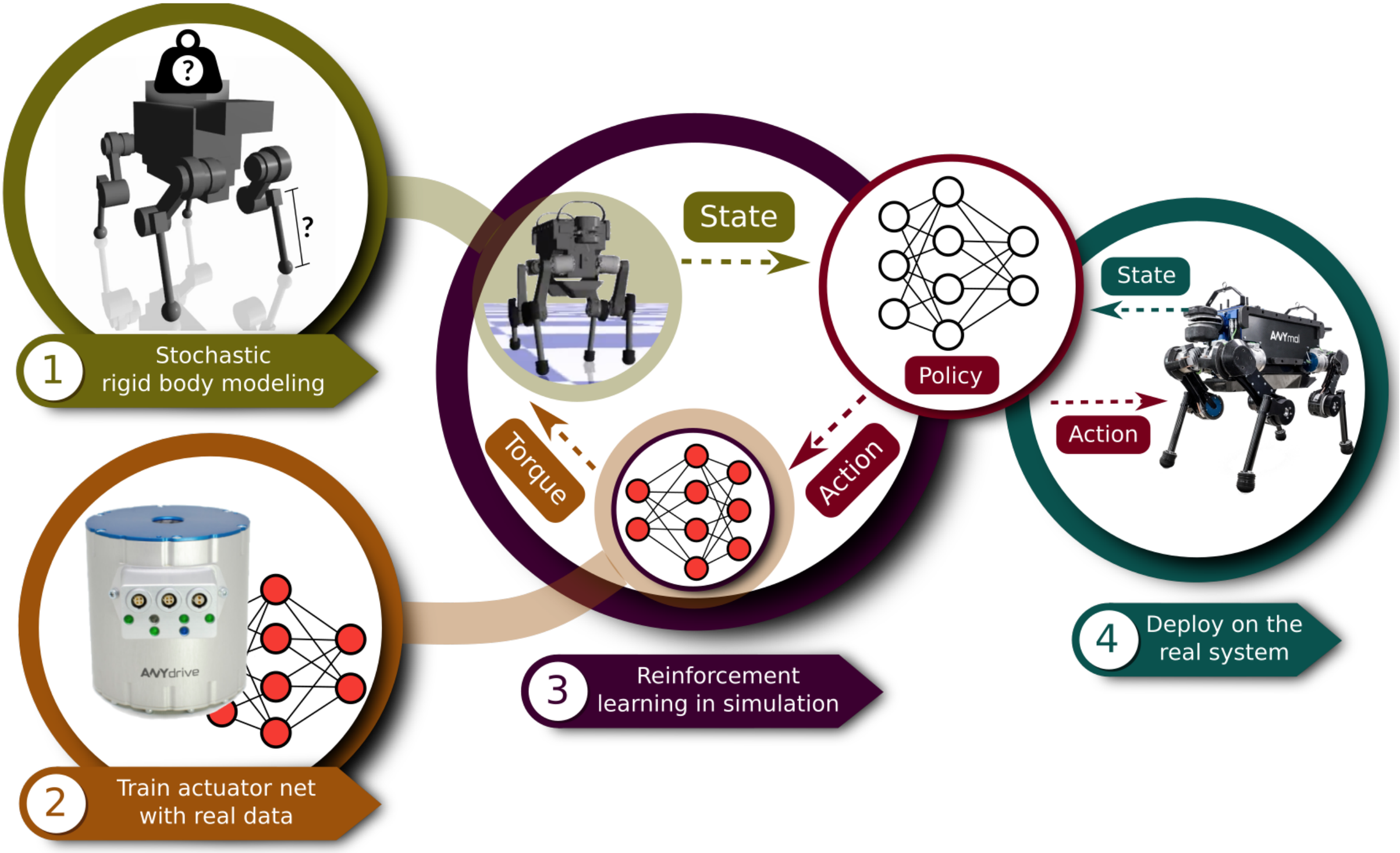


Controlled via SEA motors, hard to analytically model

- Faster, more energy efficient motion than hand-crafted controllers
- Robust recovery motions

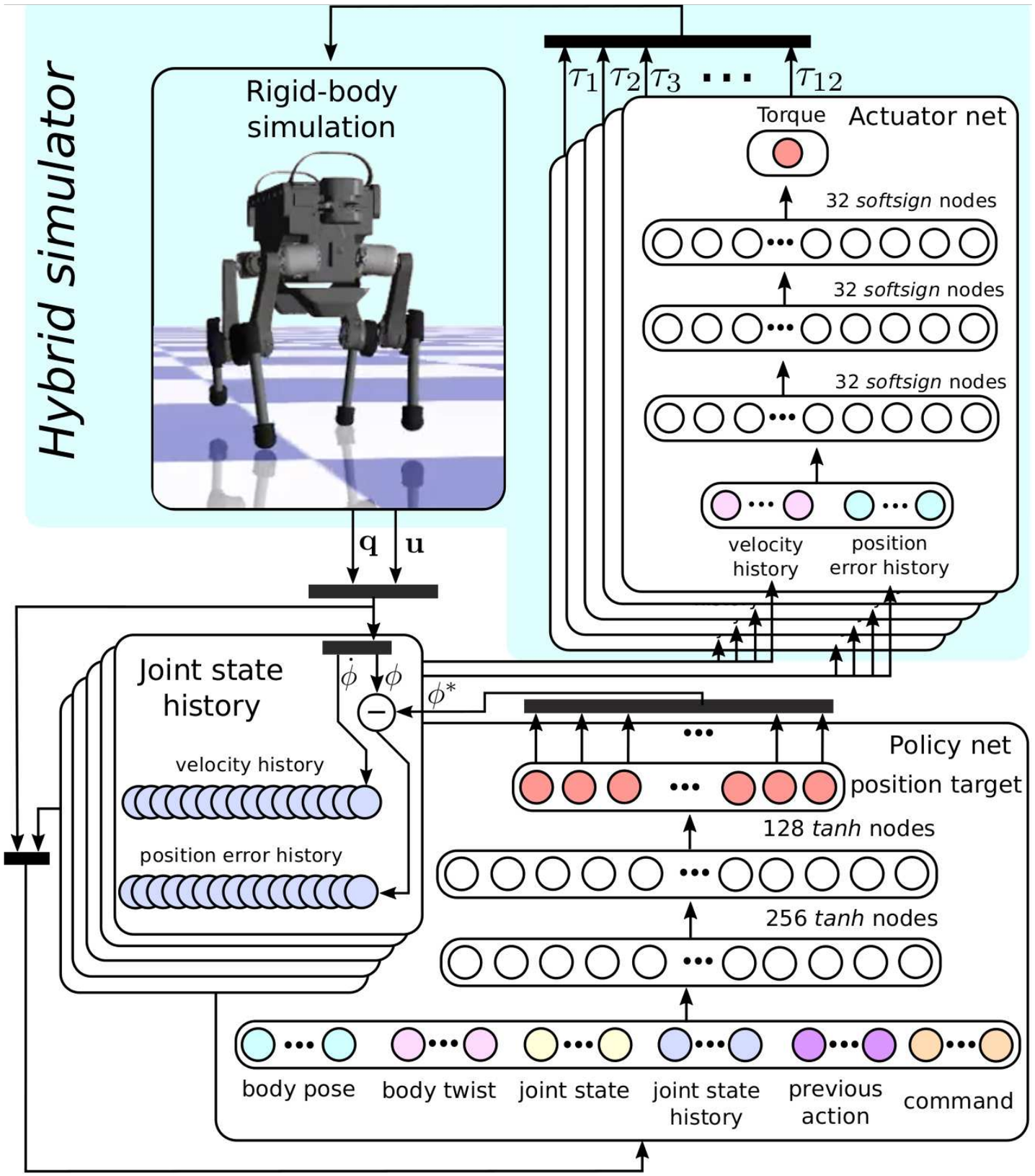


Overall Pipeline

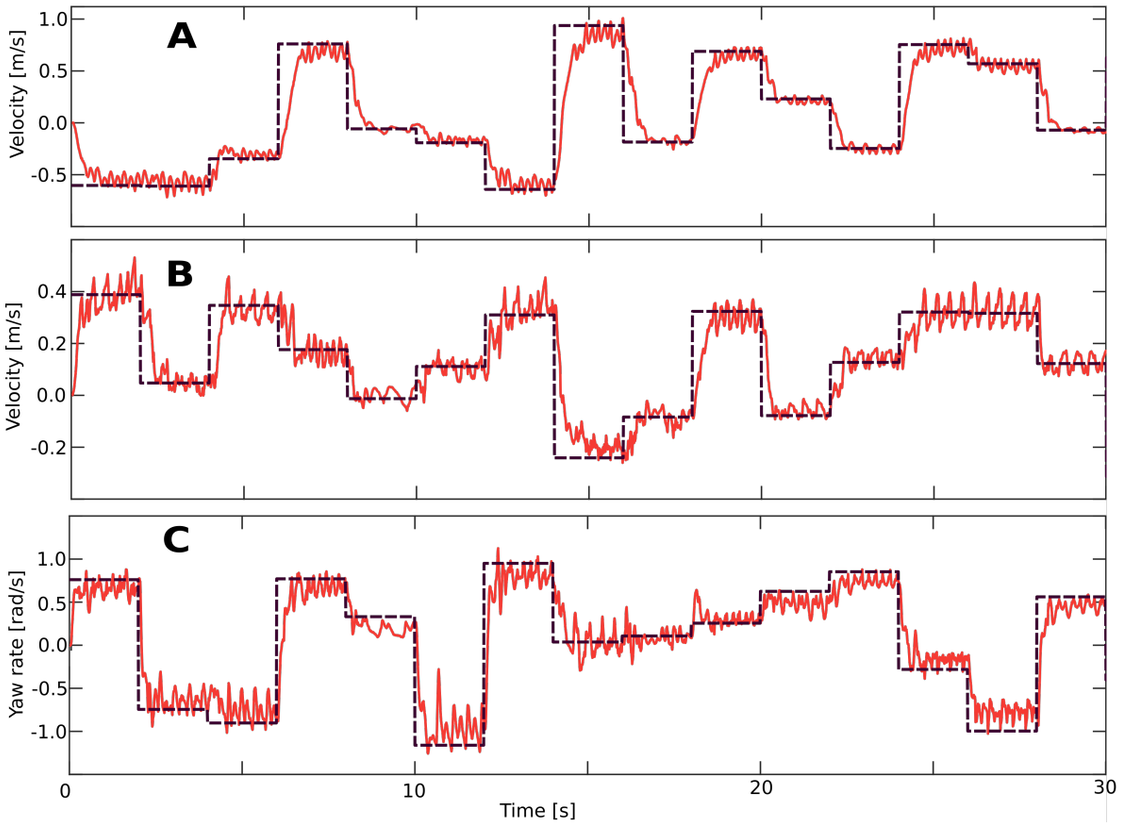


Actuator Net

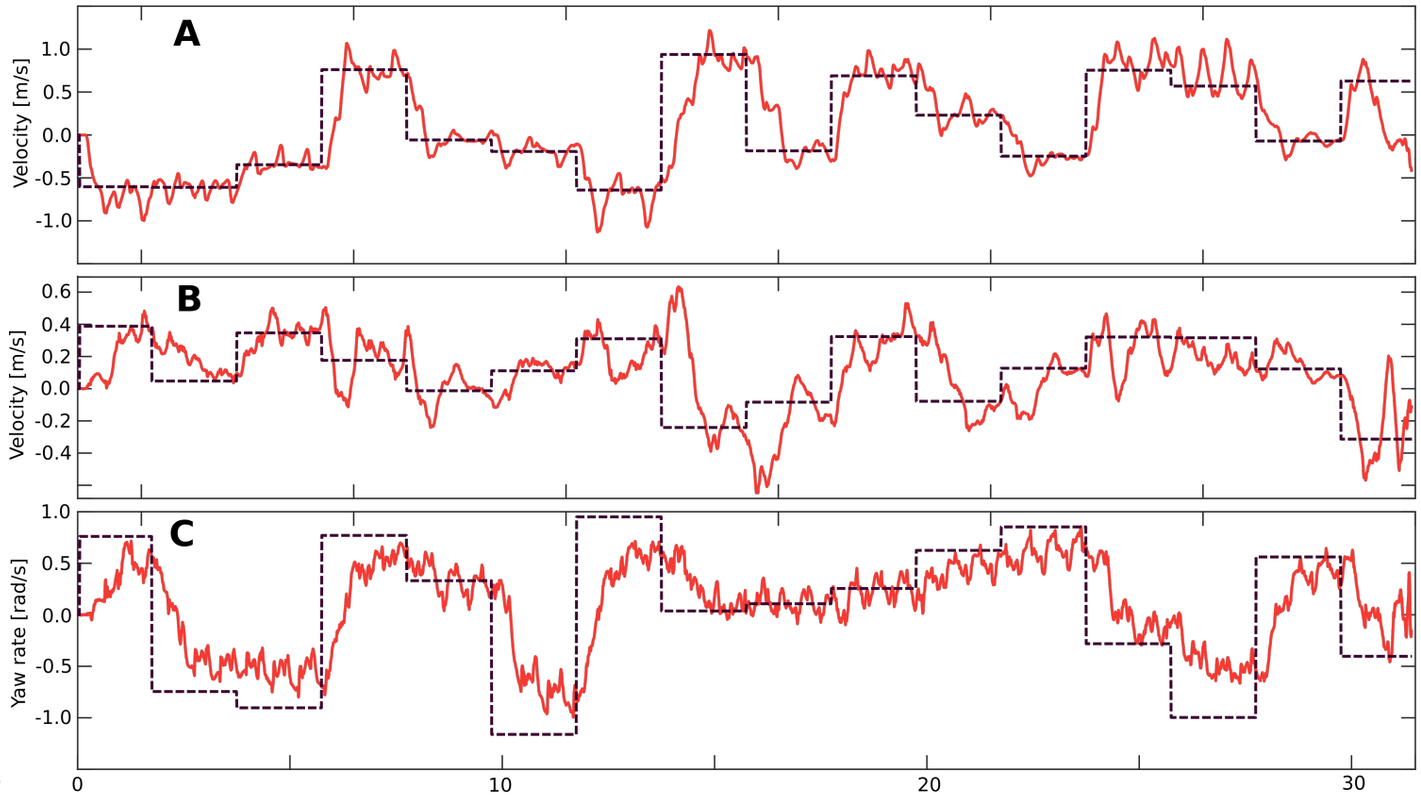
- Training:
  - Collect data on physical platform by executing hand-crafted gaits
    - varying frequencies, speeds
  - Record commanded joint locations, and associated torques



Control Performance at Test-time



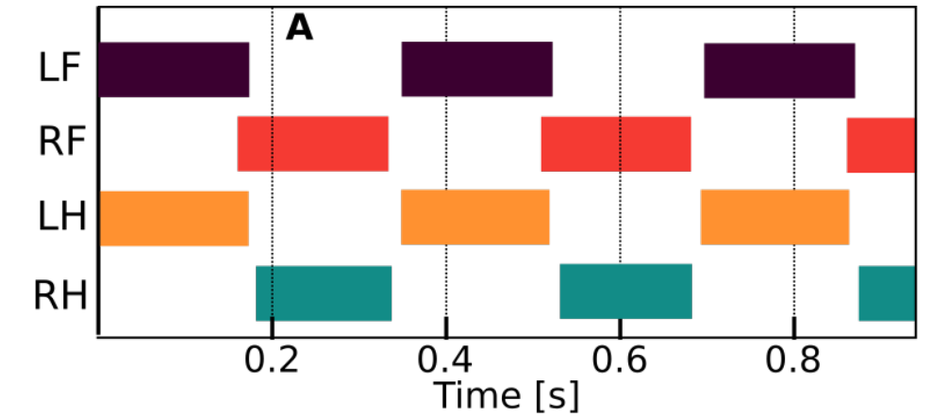
ANYmal

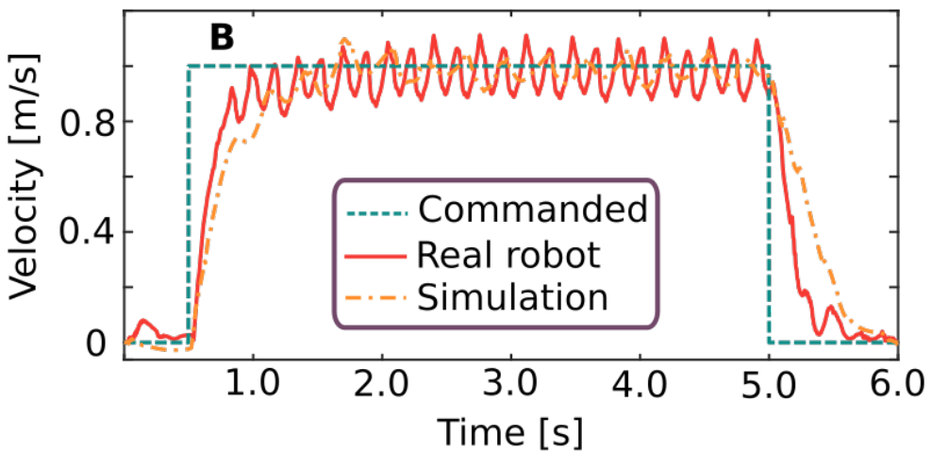


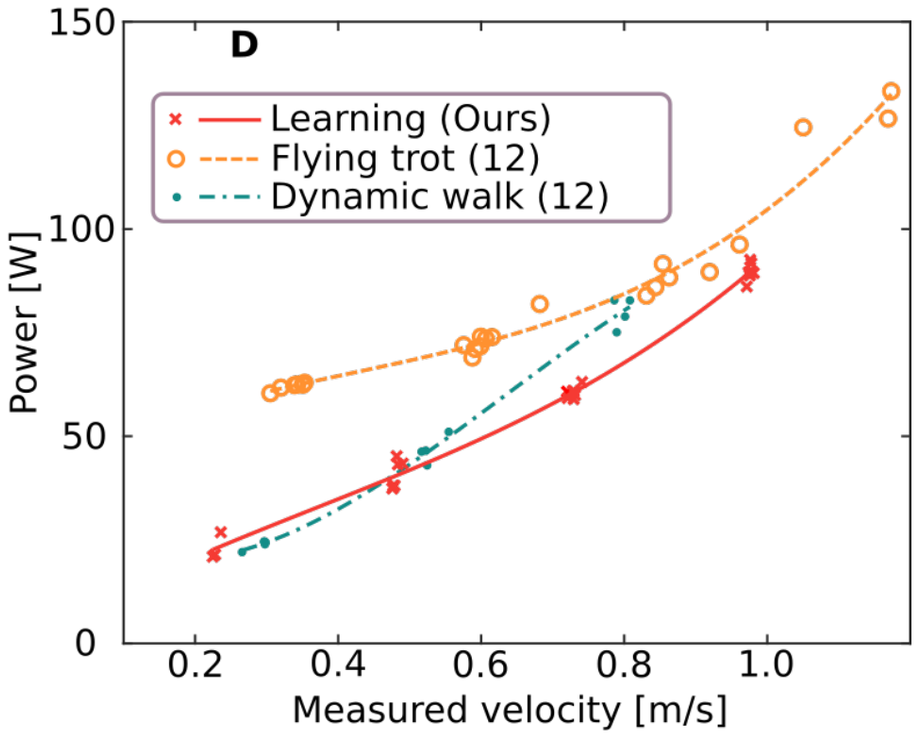
Baseline

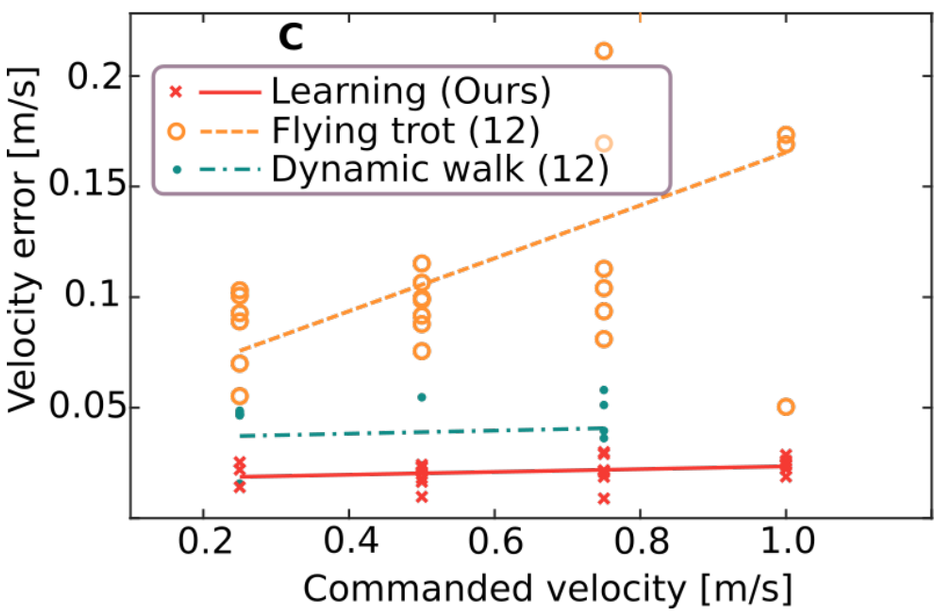
Control Performance at Test-time

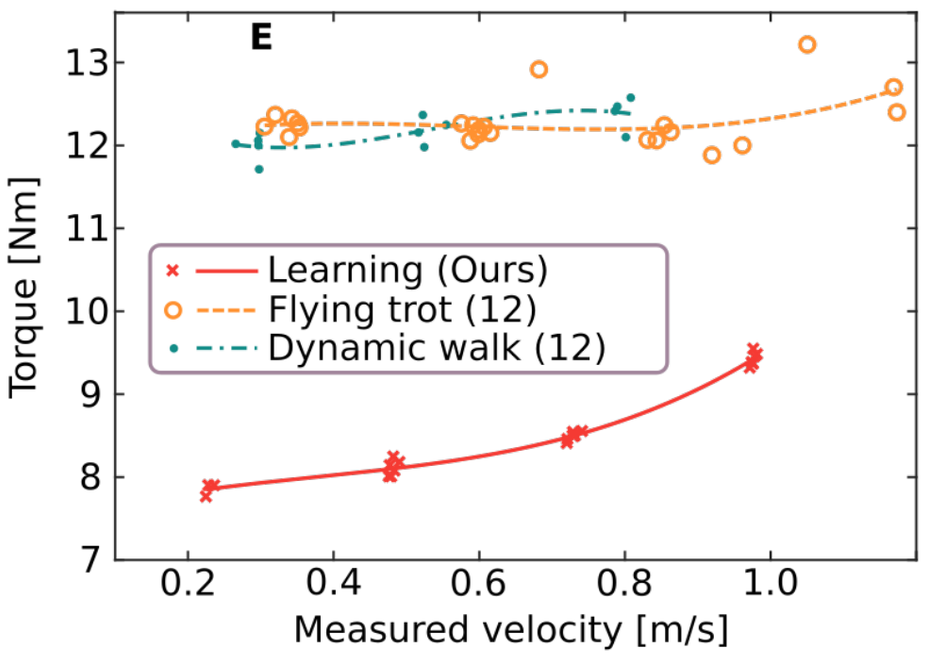
Efficient test-time motion

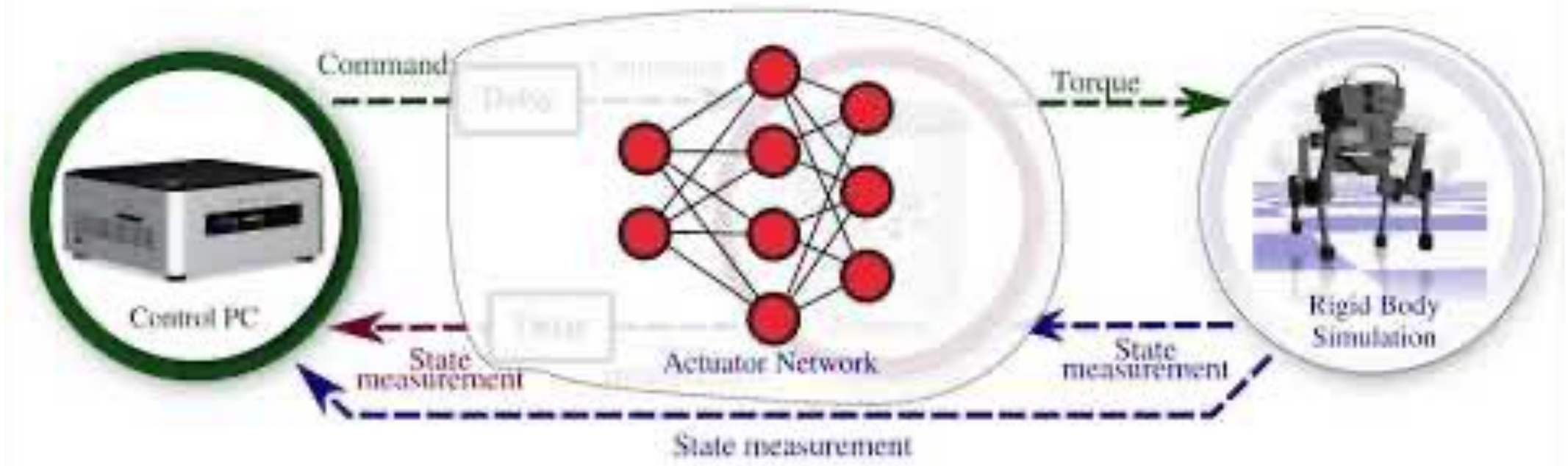






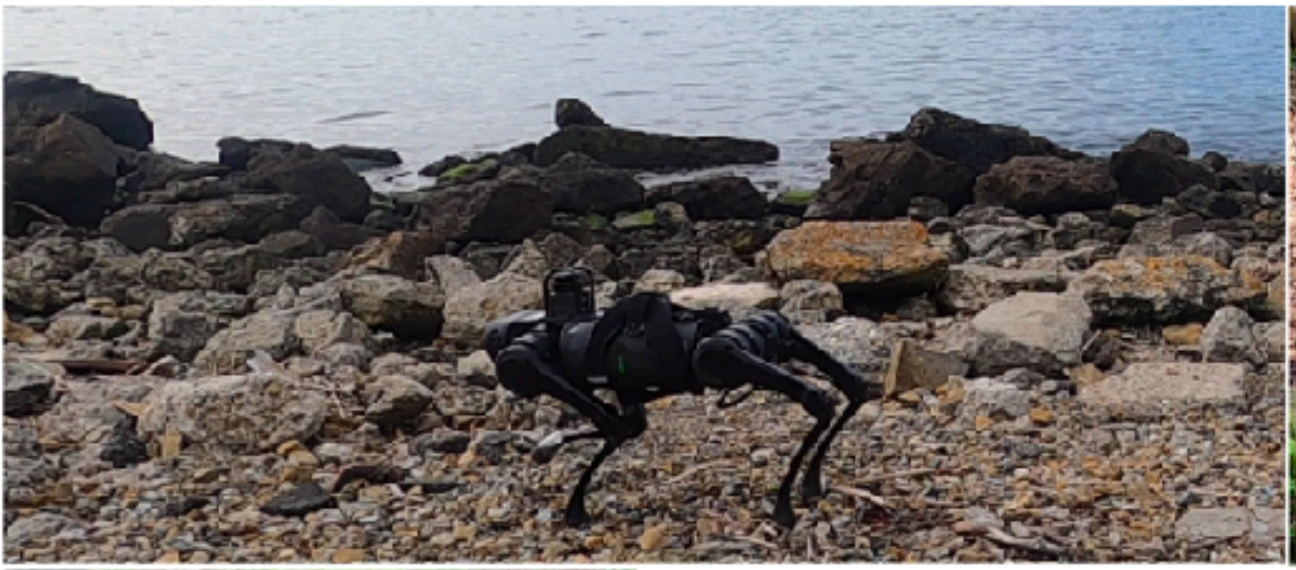


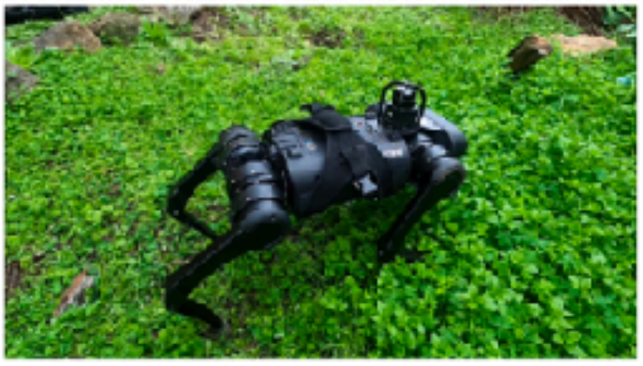




To this end, we train a neural network representing this complex dynamics with data from the real robot.

RMA (Rapid Motor Adaptation)





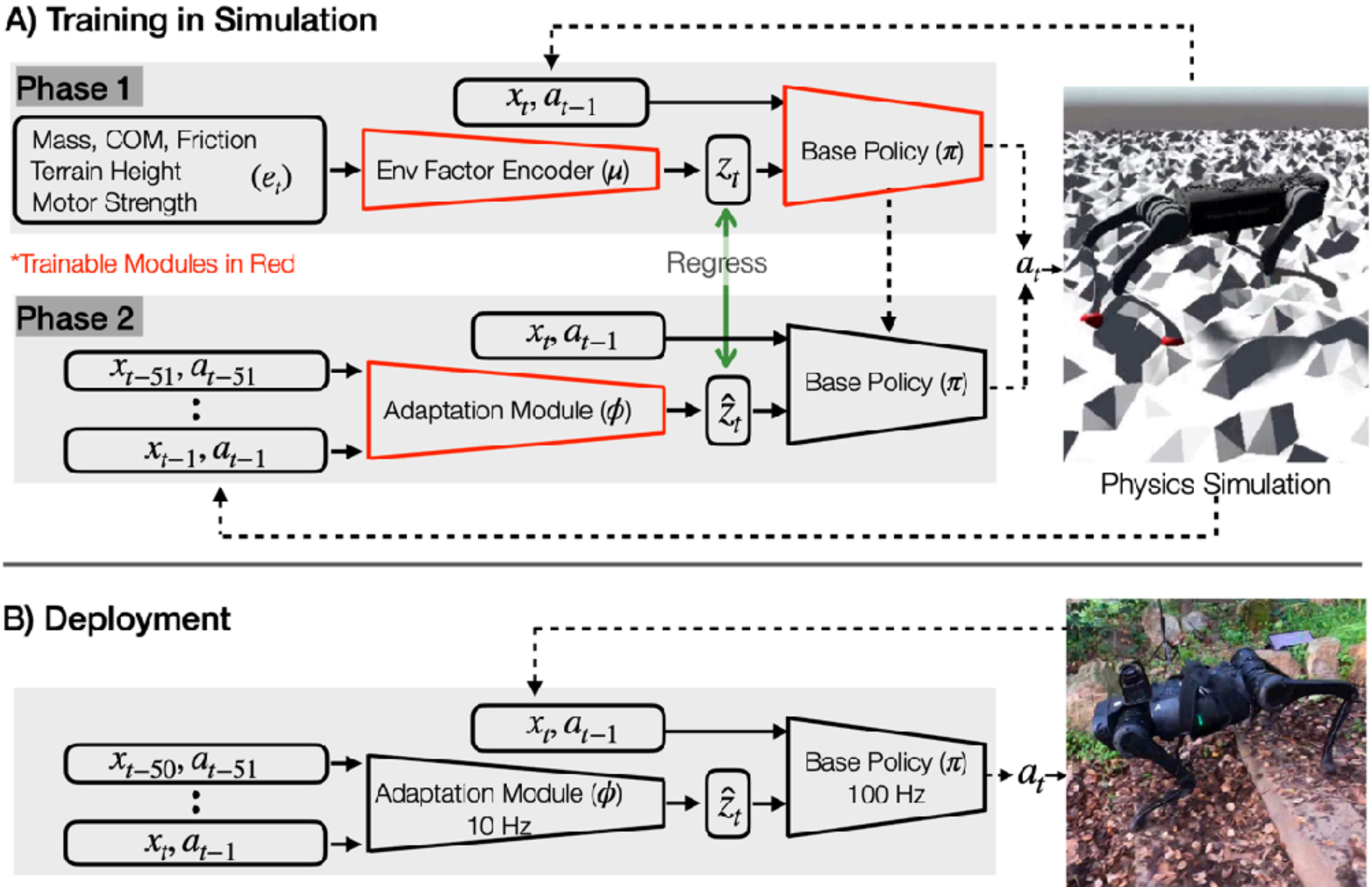





Motivation

- Hand-designing controllers for complex systems is hard
  - Employ learning in simulation
- Learning with domain randomization causes overly-cautious behavior, walking on different terrains may need different behavior
  - Adapt controllers to the situation you are in
- System identification
  - Full system identification may not be necessary

RMA (Rapid Motor Adaptation)



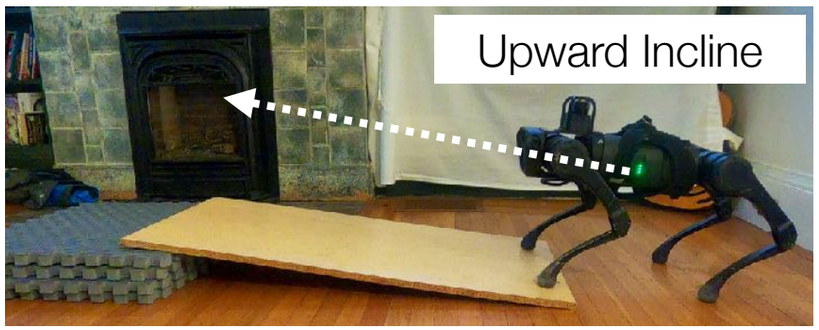
PID Controllers

- Consider a 1D system with 1D control. Unknown dynamics, but we know that a positive control u, increases x
- Suppose we want to bring the system to a desired state x_d
- Proportional control: $u_P = -k_p(x - x_d)$
- Just P control has the risk that system has a non-zero steady state error (think a robot arm under gravity)
  - Add an integral term: $u_I = -k_I I(t), I(t) = \int_0^t (x(t) - x_d(t))dt$
- Just PI control can cause systems to oscillate
  - Add a derivative term: $u_D = -k_D(\dot{x} - \dot{x}_d)$
- Total control: $u_P + u_I + u_D$
- https://www.matthewpeterkelly.com/tutorials/pdControl/index.html

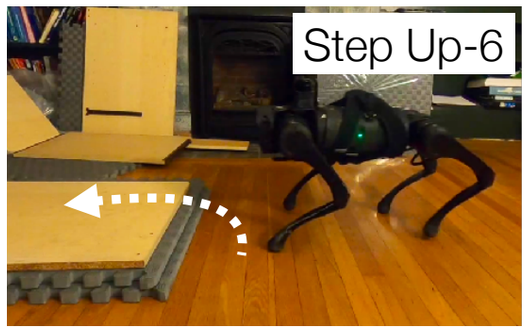
Results

	Success	TTF	Distance
RMA	80	0.9	0.9
RMA w/o Adapt	0	0.1	0.1
A1	20	0.36	0.36

	Success
RMA	100
RMA w/o Adapt	0
A1	60

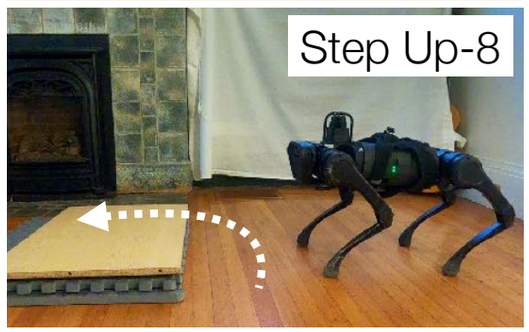


RMA	100	1	1
RMA w/o Adapt	20	1	0.6
A1	100	1	1

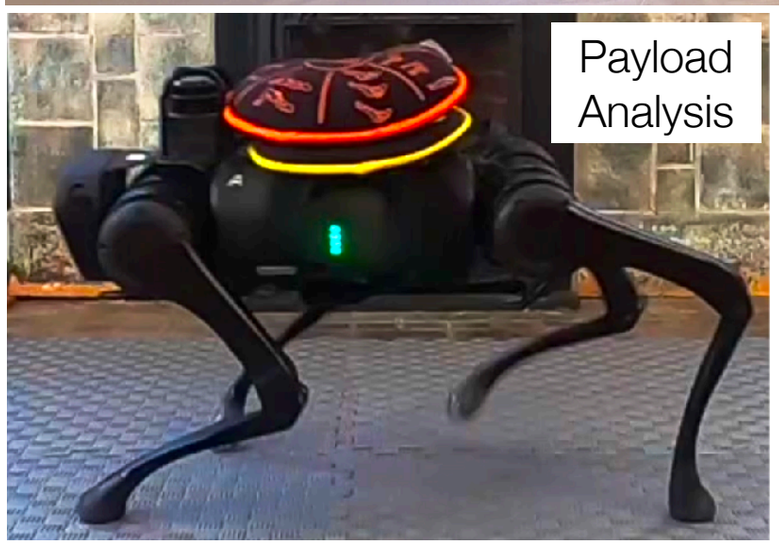


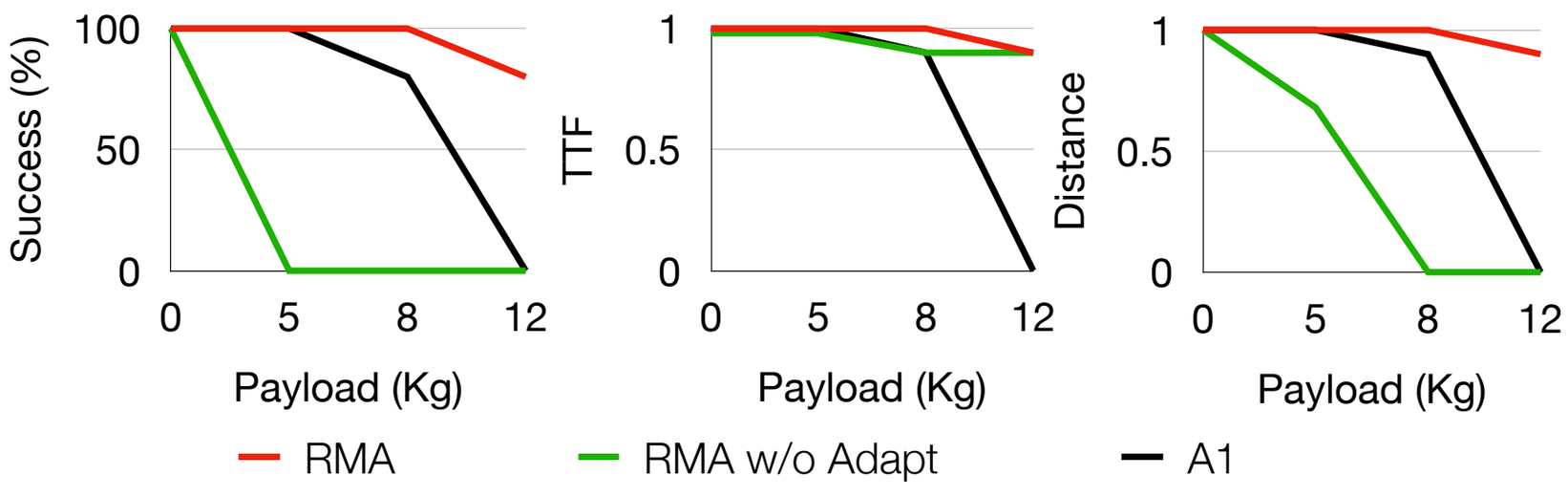
RMA	100
RMA w/o Adapt	40
A1	80

RMA	100	1	1
RMA w/o Adapt	0	1	0.25
A1	100	1	1

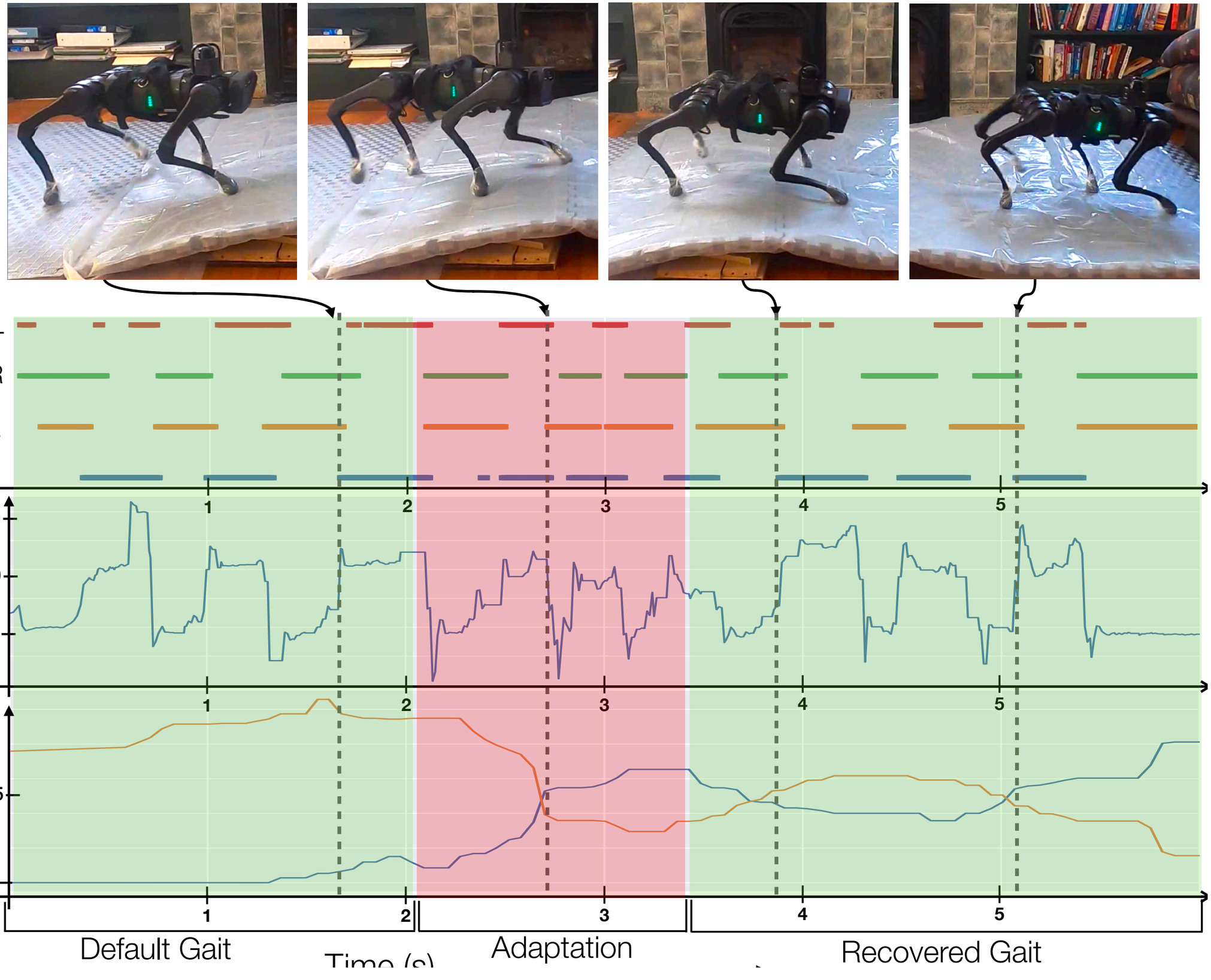


RMA	60
RMA w/o Adapt	0
A1	20





Adaptation


RL
RR
FL
FR
Knee Torque (Nm)
20
10
0
z_1, z_5 value
0.5
0
1
2
3
4
5
Default Gait
Adaptation
Recovered Gait
Time (s)

Thank you