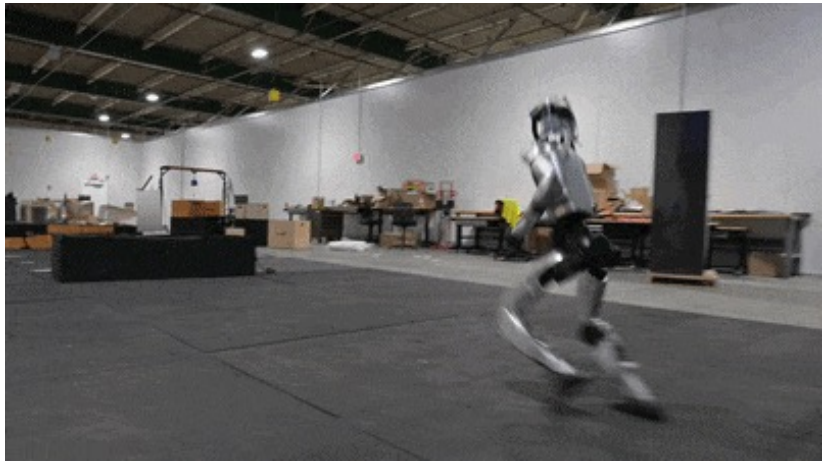
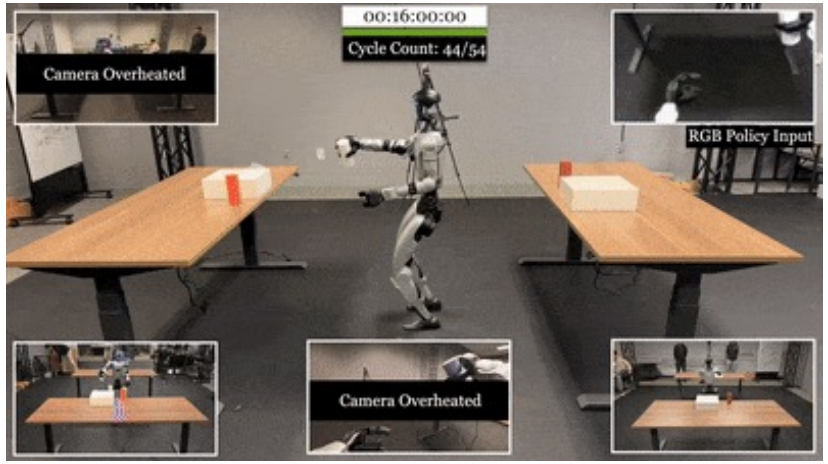
Sim2Real Robot Learning: A Holistic Overview

OmniRetarget (ICRA'26)
https://omniretarget.github.io/



PHP (RSS'26)
https://php-parkour.github.io/



VIRAL (CVPR'26)
https://viral-humanoid.github.io/

Guanya Shi

Assistant Professor, Robotics Institute, CMU

Amazon Scholar, Frontier AI & Robotics (FAR)

https://lecar-lab.github.io/

LeCAR
Learning and Control for Agile Robotics Lab

Carnegie Mellon University
Robotics Institute

Before We Start

- ❑ Sim2real robot learning is a big topic! Today we will bias towards breadth so there will be many subtopics :)

- ❑ The papers I selected represent only a very small subset, and the selection is heavily biased (many papers are from my group)

- ❑ All selected papers are open-sourced!

- ❑ Humanoid sim2real framework: https://github.com/amazon-far/holosoma
  - Regularly maintained and updated
  - Retargeting, policy training, deployment

Outline

- ❑ Intro: What is robot simulation? Why is sim2real useful but challenging?

- ❑ Methods to reduce the sim2real gap:
  - Domain randomization
  - Learning to adapt / teacher-student
  - Real2sim2real

- ❑ Advanced topics:
  - Human data + sim2real learning
  - RL algorithms for sim2real policy learning

- ❑ Zooming out: Sim2Real 1.0 to 4.0

Outline

- ❑ Intro: What is robot simulation? Why is sim2real useful but challenging?

- ❑ Methods to reduce the sim2real gap:
  - Domain randomization
  - Learning to adapt / teacher-student
  - Real2sim2real

- ❑ Advanced topics:
  - Human data + sim2real learning
  - RL algorithms for sim2real policy learning

- ❑ Zooming out: Sim2Real 1.0 to 4.0

Robot Simulators

- ❑ Thanks to computer graphics and computational physics, there are many simulators!
- ❑ We focus on *physics-based simulators* today
  - I.e., the simulation process is governed by some *explicit* physical laws
- ❑ Note that there are also non-physics-based simulators (e.g., world model)

Robot Simulators

- ❑ Simulator taxonomy by https://simulately.wiki/docs/comparison
- ❑ Many names!
- ❑ Simulator ≠ framework
  - IsaacSim: Simulator
  - IsaacLab: Robot learning framework on top of IsaacSim
- ❑ MuJoCo family
  - MuJoCo (CPU)
  - MuJoCo XLA (MJX) based on JAX
  - MuJoCo Warp
  - mjlab: IsaacLab-style manager-based API. mjlab = MuJoCo Warp + IsaacLab - IsaacSim

Simulator	Physics Engine	Rendering	Sensor	Dynamics	GPU-accelerated Simulation	Open-Source
IsaacSim	PhysX 5	Rasterization; RayTracing	RGBD; Lidar; Force; Effort; IMU; Contact; Proximity	Rigid;Soft;Cloth;Fluid	✓	✘
IsaacGym	PhysX 5, Flex	Rasterization;	RGBD; Force; Contact;	Rigid;Soft;Cloth	✓	✘
SAPIEN	PhysX 5, Warp	Rasterization; RayTracing;	RGBD; Force; Contact;	Rigid;Soft;Fluid	✓	✓
Pybullet	Bullet	Rasterization;	RGBD; Force; IMU; Tactile;	Rigid;Soft;Cloth	✘	✓
MuJoCo	MuJoCo	Rasterization;	RGBD; Force; IMU; Tactile;	Rigid;Soft;Cloth	✓	✓
CoppeliaSim	MuJoCo; Bullet; ODE; Newton; Vortex	Rasterization; RayTracing;	RGBD; Force; Contact;	Rigid;Soft;Cloth	✘	✓
Gazebo	Bullet; ODE; DART; Simbody	Rasterization;	RGBD; Lidar; Force; IMU;	Rigid;Soft;Cloth	✘	✓
Genesis	Genesis	Rasterization; Raytracing	RGBD; Tactile; (update soon)	Rigid;Soft;Cloth;Fluid	✓	✓

Sim2Real Policy Learning: A General Recipe

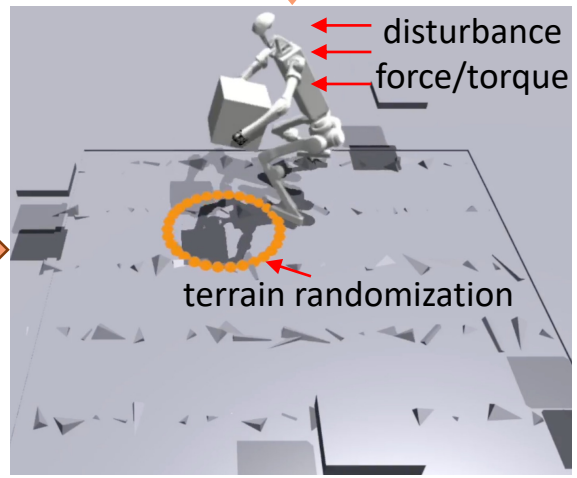
Real2Sim

NVIDIA ISAAC

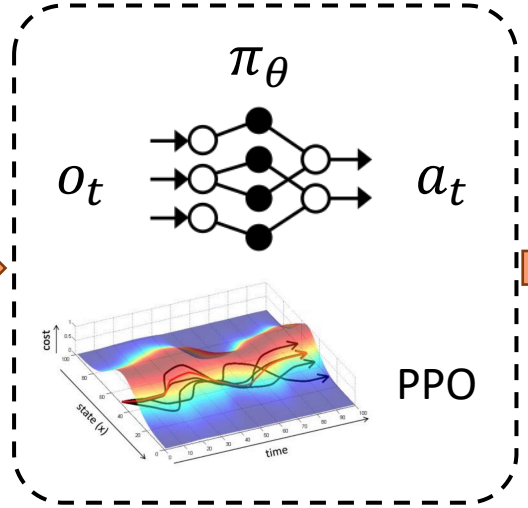
GENESIS

MuJoCo

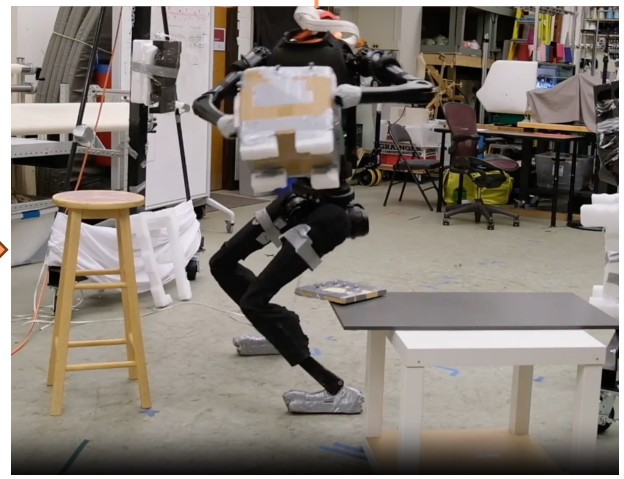
Physics-based Simulators



Massively Parallel
Training Environments



Policy Optimization
using RL



Real-world
Deployment

Why Simulators for Robot Learning?

Advantages of using simulated data:

- ❑ Cheap, fast, and scalable
- ❑ Safe
- ❑ Labeled (we have access to “oracle” or “ground truth”)
- ❑ Avoid wear and tear of the robot

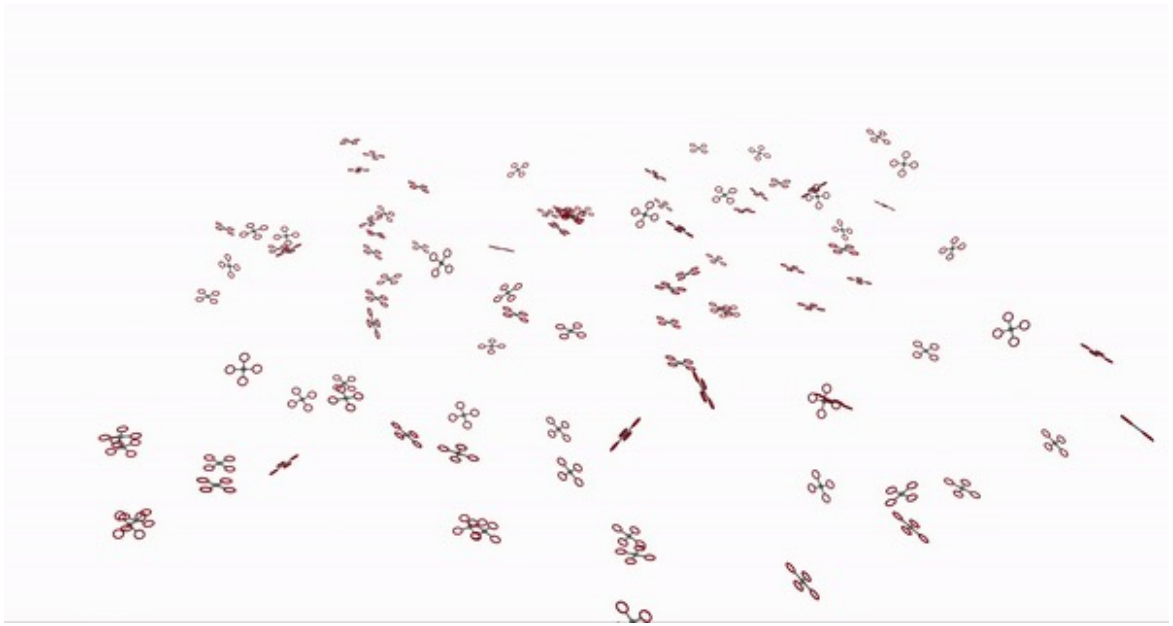
Why Simulators for Robot Learning?

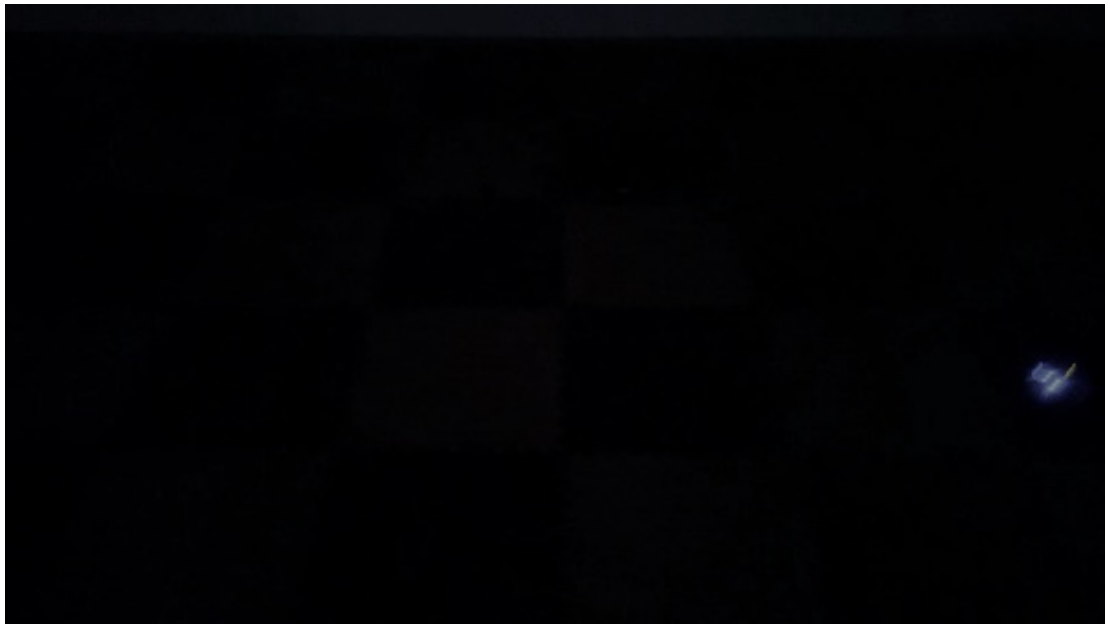
- ❑ Extremely fast and scalable!

Learning to Fly in Seconds

Jonas Eschmann[1,2], Dario Albani[2], and Giuseppe Loianno[1]

Simulation. We run our multirotor dynamics simulator on a Nvidia T2000 laptop GPU and attain 1284 million steps/s. To reach this level of performance, 64 blocks of 128 threads are each executing the forward dynamics in parallel. This amounts to 8192 environments in total which are run for 1 000 000 steps each. The required execution time of the GPU kernel is 6380 ms amounting to 1284 million steps/s. At a simulation frequency of 100 Hz this is about 5 months of simulated flight per second. Compared to Flightmare [13] which is the state of the art in terms of dynamics simulation speed with a reported frequency of 200 000 steps/s on a laptop, our simulation is about 6420× faster.



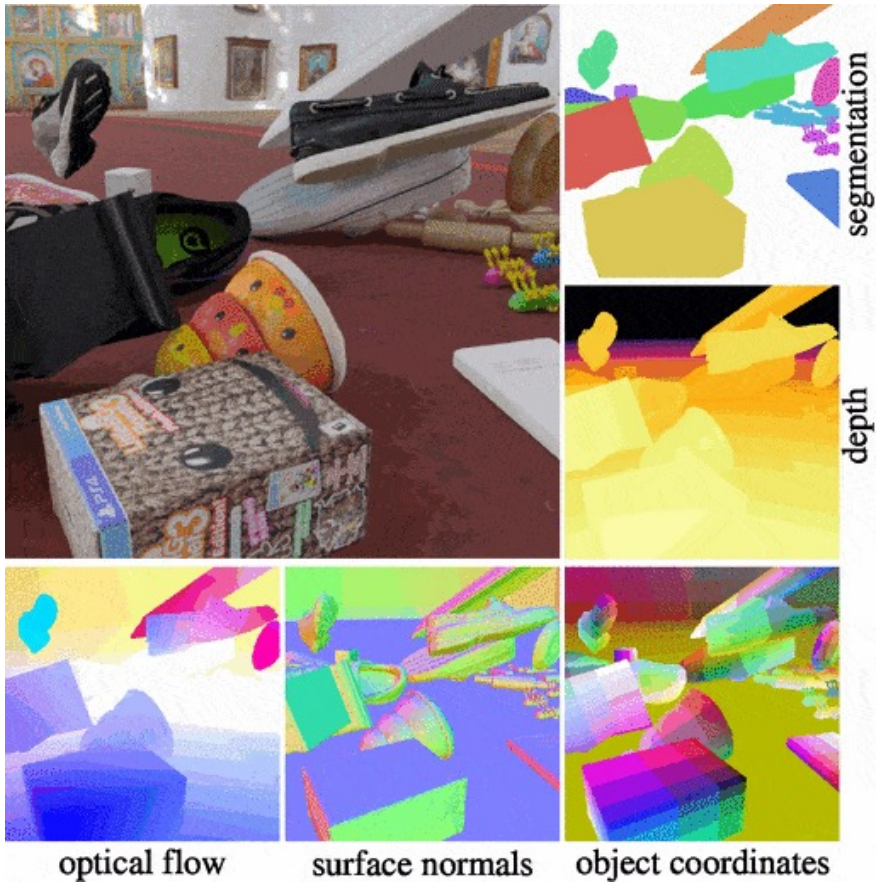


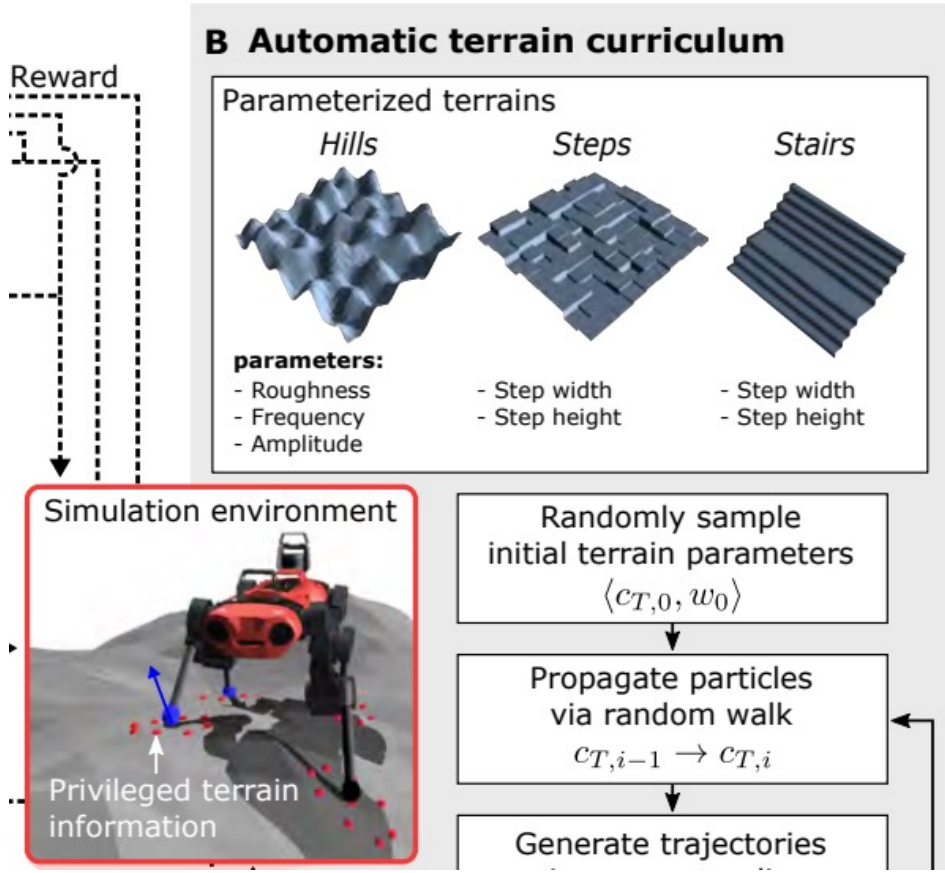
Trained for **18s** on a 2020 MacBook Pro (M1) using RLtools.

Why Simulators for Robot Learning?

- ❑ Get expensive labels for free!

Ground truth in PyBullet





Learning quadrupedal locomotion over challenging terrain, Joonho Lee et al., Science Robotics

The Problems/Challenges of Sim2Real

However, sim2real is never easy!

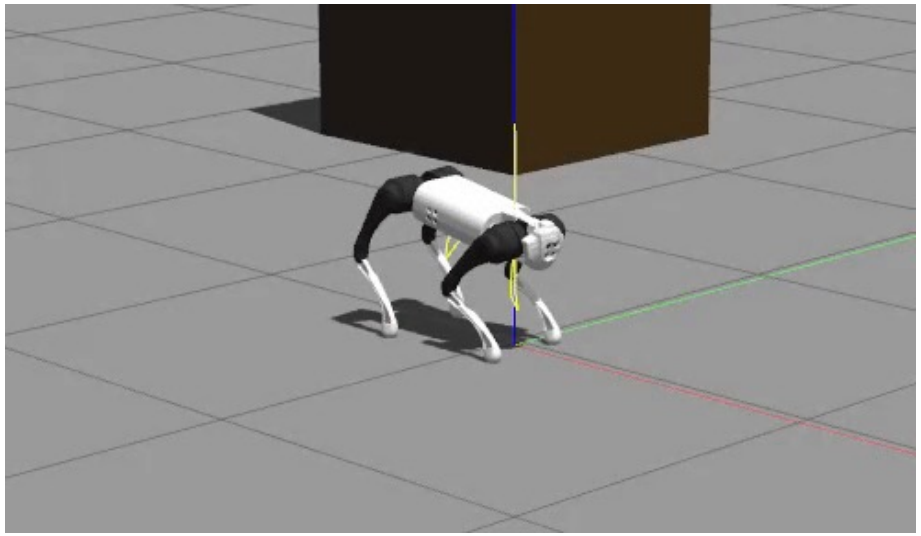
- ❑ It is hard to exactly replicate the real world
- ❑ Parametric mismatches (i.e., simulator uses different parameters than the real)
- ❑ Non-parametric mismatches (i.e., simulator doesn't consider some effects at all)

Another big challenge of sim2real: How to design rewards or define tasks?

- ❑ Will back to this point in the "human data + sim2real learning" session

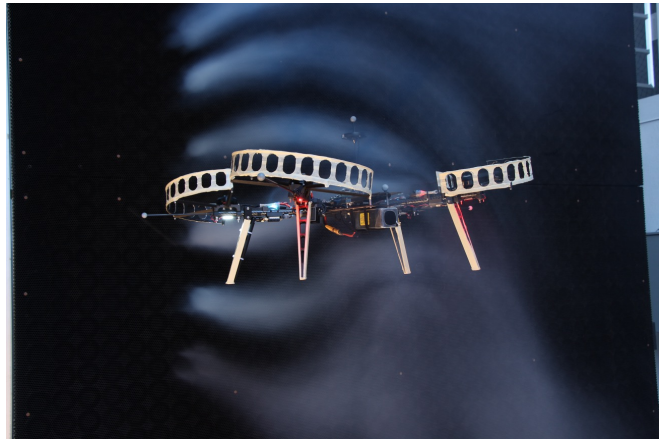
The Problems/Challenges of Sim2Real

- Two types of mismatches
  - **Parametric mismatches:** robot mass/inertia, friction, etc
  - Non-parametric mismatches: complex aerodynamics, fluid dynamics, tire dynamics, etc

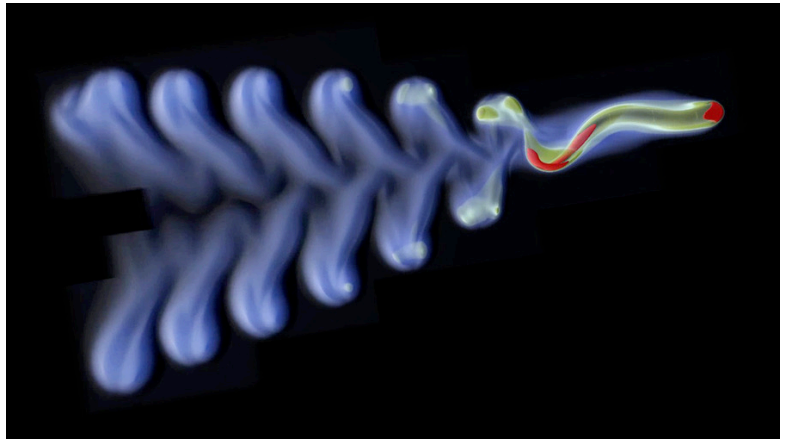


The Problems/Challenges of Sim2Real

- Two types of mismatches
  - Parametric mismatches: robot mass/intentia, friction, etc
  - **Non-parametric mismatches:** complex aerodynamics, tire dynamics, etc



Aerodynamics in wind, *Neural-Fly*



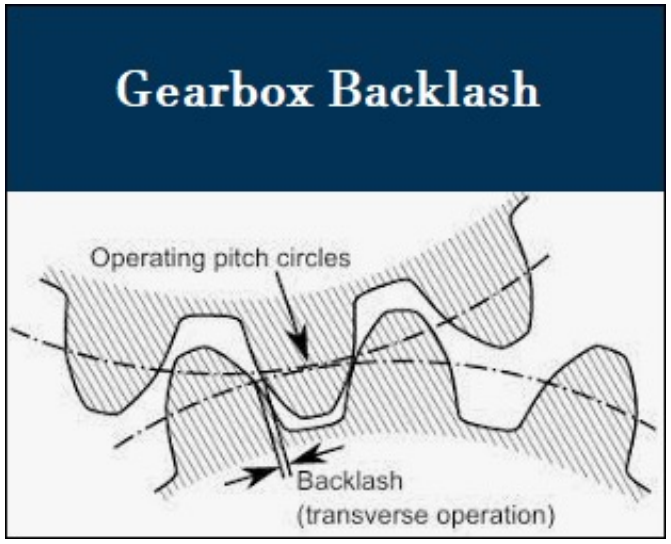
Fluid dynamics, MIT van Rees Lab

Offroad vehicle dynamics, UW Racer team

The Problems/Challenges of Sim2Real

- Two types of mismatches
  - Parametric mismatches: robot mass/intentia, friction, etc
  - **Non-parametric mismatches:** complex aerodynamics, tire dynamics, etc

Imperfect modeling of links



Outline

- ❑ What is robot simulators? Why is sim2real useful but challenging?

- ❑ Methods to reduce the sim2real gap:
  - Domain randomization
  - Learning to adapt / teacher-student
  - Real2sim2real

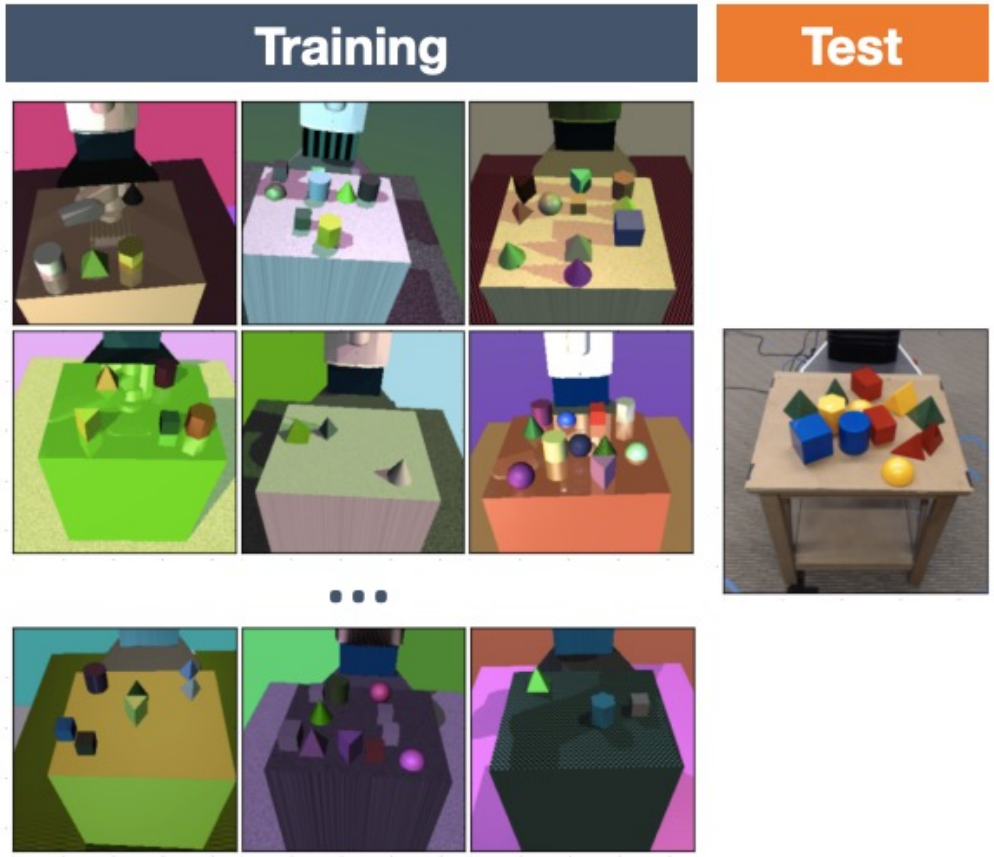
- ❑ Advanced topics:
  - Human data + sim2real learning
  - RL algorithms for sim2real policy learning

- ❑ Zooming out: Sim2Real 1.0 to 4.0

Domain Randomization

- Key idea: randomize e in $x_{t+1} = f_{\text{sim}}(x_t, u_t, e)$
- Train a *single* RL policy $\pi(x)$ that works for *many* e
- Essentially, **robust control**!

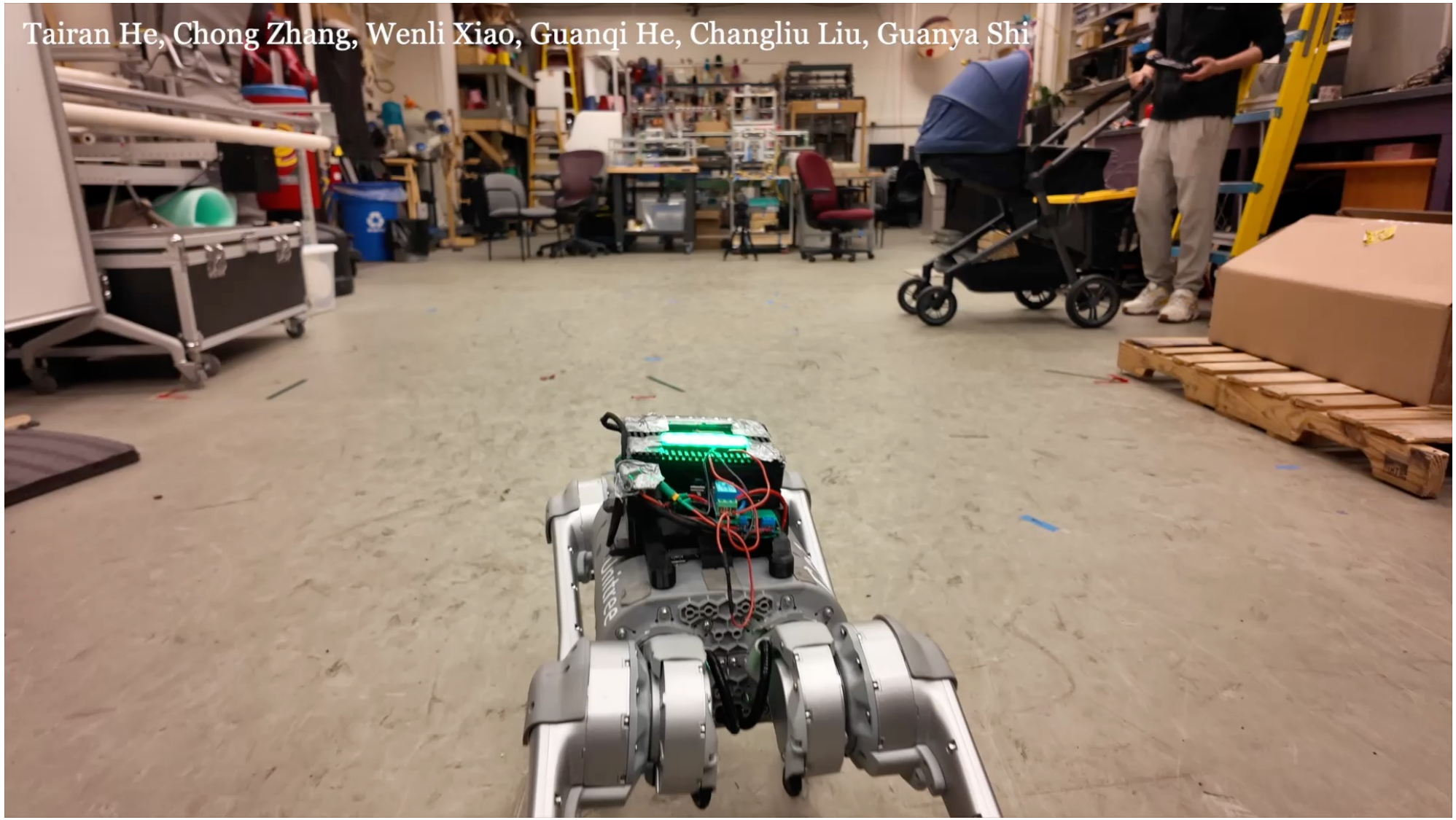
- In the original paper, focusing on perception
- But this idea has been widely used in all parts in simulation (perception, dynamics, sensor input, delay, etc)



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]

Domain Randomization Example: Agile But Safe



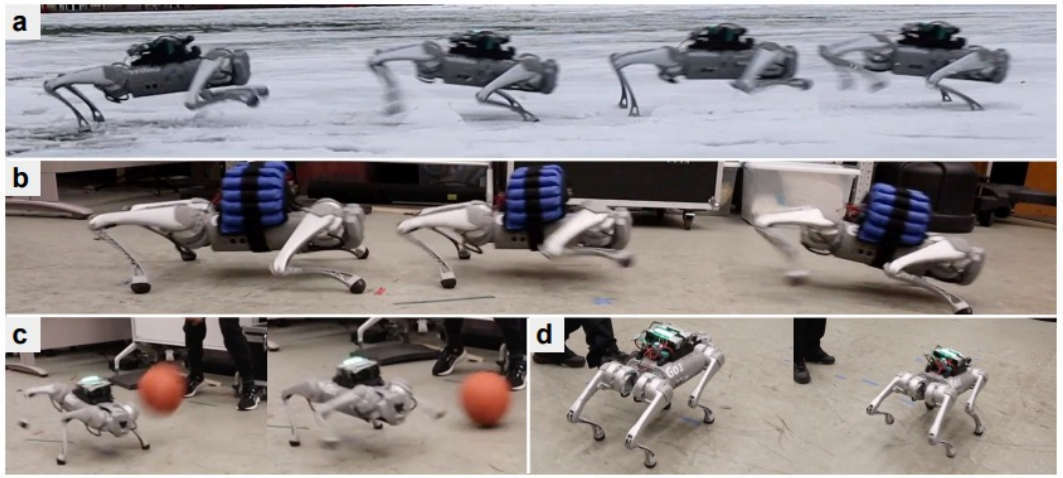
https://agile-but-safe.github.io/

Domain Randomization Example: Agile But Safe

- Randomize e in $x_{t+1} = f_{\text{sim}}(x_t, u_t, e)$. Train a *single* RL policy $\pi(x)$ that works for *many* e

TABLE II
DOMAIN RANDOMIZATION SETTINGS FOR AGILE POLICY TRAINING

Term	Value
Observation	
Illusion	Enabled
Joint position noise	$\mathcal{U}(-0.01, 0.01)$ rad
Joint velocity noise	$\mathcal{U}(-1.5, 1.5)$ rad/s
Angular velocity noise	$\mathcal{U}(-0.2, 0.2)$ rad/s
Projected gravity noise	$\mathcal{U}(-0.05, 0.05)$
log(ray distance) noise	$\mathcal{U}(-0.2, 0.2)$
Dynamics	
ERFI-50 [8]	0.78 N m× difficulty level
Friction factor	$\mathcal{U}(0.4, 1.1)$
Added base mass	$\mathcal{U}(-1.5, 1.5)$ kg
Joint position biases	$\mathcal{U}(-0.08, 0.08)$ rad
Episode	
Episode length	$\mathcal{U}(7.0, 9.0)$ s
Initial robot position	$x = 0, y = 0$
Initial robot yaw	$\mathcal{U}(-\pi, \pi)$ rad
Initial robot twist	$\mathcal{U}(-0.5, 0.5)$ m/s or rad/s
Goal Position	$x_{\text{goal}} \sim \mathcal{U}(1.5, 7.5)$ m $y_{\text{goal}} \sim \mathcal{U}(-2.0, 2.0)$ m
Goal Heading	$\arctan 2(y_{\text{goal}}, x_{\text{goal}}) + \mathcal{U}(-0.3, 0.3)$ rad



https://agile-but-safe.github.io/

Domain Randomization Example: RPL

[RPL: Learning Robust Humanoid Perceptive Locomotion on Challenging Terrains, Zhang et al.] *Work done at Amazon FAR*

Domain Randomization Example: RPL

❑ Randomize e in $x_{t+1} = f_{\text{sim}}(x_t, u_t, e)$. Train a *single* RL policy $\pi(x)$ that works for *many* e

TABLE III
DOMAIN RANDOMIZATION TERMS DURING DISTILLATION.

Term	Value
Dynamics Randomization	
Friction	$\mathcal{U}(0.5, 1.25)$
Link mass	$\mathcal{U}(0.9, 1.2) \times$ default kg
Base mass	$\mathcal{U}(-1.0, 3.0)$ kg
Base COM range	$x \sim \mathcal{U}(-0.025, 0.025)$m
	$y \sim \mathcal{U}(-0.05, 0.05)$m
	$z \sim \mathcal{U}(-0.05, 0.05)$m
Control delay	$\mathcal{U}(0, 20)$ms
Perception Randomization (Depth Cameras)	
Depth cam translation	$x \sim \mathcal{U}(-0.025, 0.025)$ m
	$y \sim \mathcal{U}(-0.025, 0.025)$ m
	$z \sim \mathcal{U}(-0.025, 0.025)$ m
Depth cam rotation (Euler)	$\phi \sim \mathcal{U}(-2.5, 2.5)^\circ$
	$\theta \sim \mathcal{U}(-3.0, 3.0)^\circ$
	$\psi \sim \mathcal{U}(-2.5, 2.5)^\circ$
Depth cam FOV	$\Delta\text{FOV} \sim \mathcal{U}(-2.0, 2.0)^\circ$
Pixel dropout	$p_{\text{drop}} = 0.05$
Depth noise	$\sigma_d = 0.1 \cdot \text{depth}$

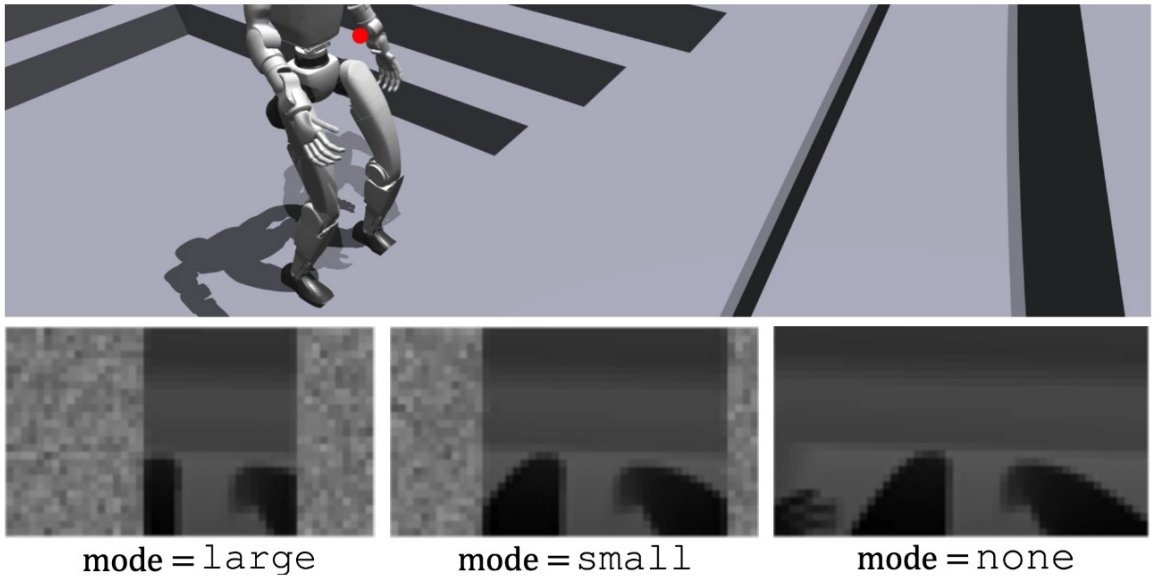


Fig. 6. **Illustration of RSM.** The red dot on the top figure is where the front camera is placed. The bottom three figures are different masking modes.

[RPL: Learning Robust Humanoid Perceptive Locomotion on Challenging Terrains, Zhang et al.] *Work done at Amazon FAR*

Outline

- ❑ What is robot simulators? Why is sim2real useful but challenging?
- ❑ Methods to reduce the sim2real gap:
  - Domain randomization
  - Learning to adapt / teacher-student
  - Real2sim2real
- ❑ Advanced topics:
  - Human data + sim2real learning
  - RL algorithms for sim2real policy learning
- ❑ Zooming out: Sim2Real 1.0 to 4.0

Learning to Adapt

- Key idea: randomize e in $x_{t+1} = f_{\text{sim}}(x_t, u_t, e)$
- Train an *adaptive* RL policy $\pi(x, e)$ that adapts to unknown e
- Essentially, **adaptive control**!
- It doesn't conflict with domain randomization. You can do both (robust adaptive control)!

- The problem of $\pi(x, e)$: e is often *unknown* in the real!
- Therefore, a common pipeline for learning to adapt: *learning from a privileged teacher:*
  - Sim: First train a teacher policy with privilege information $\pi(x, e)$
  - Sim: Then a student policy $\pi_s(x, \text{available info in the real})$ learns from $\pi(x, e)$
  - Real: Then deploy $\pi_s(x, \text{available info in the real})$ in the real
- The second phase is an imitation learning problem

Learning to Adapt Example: Locomotion

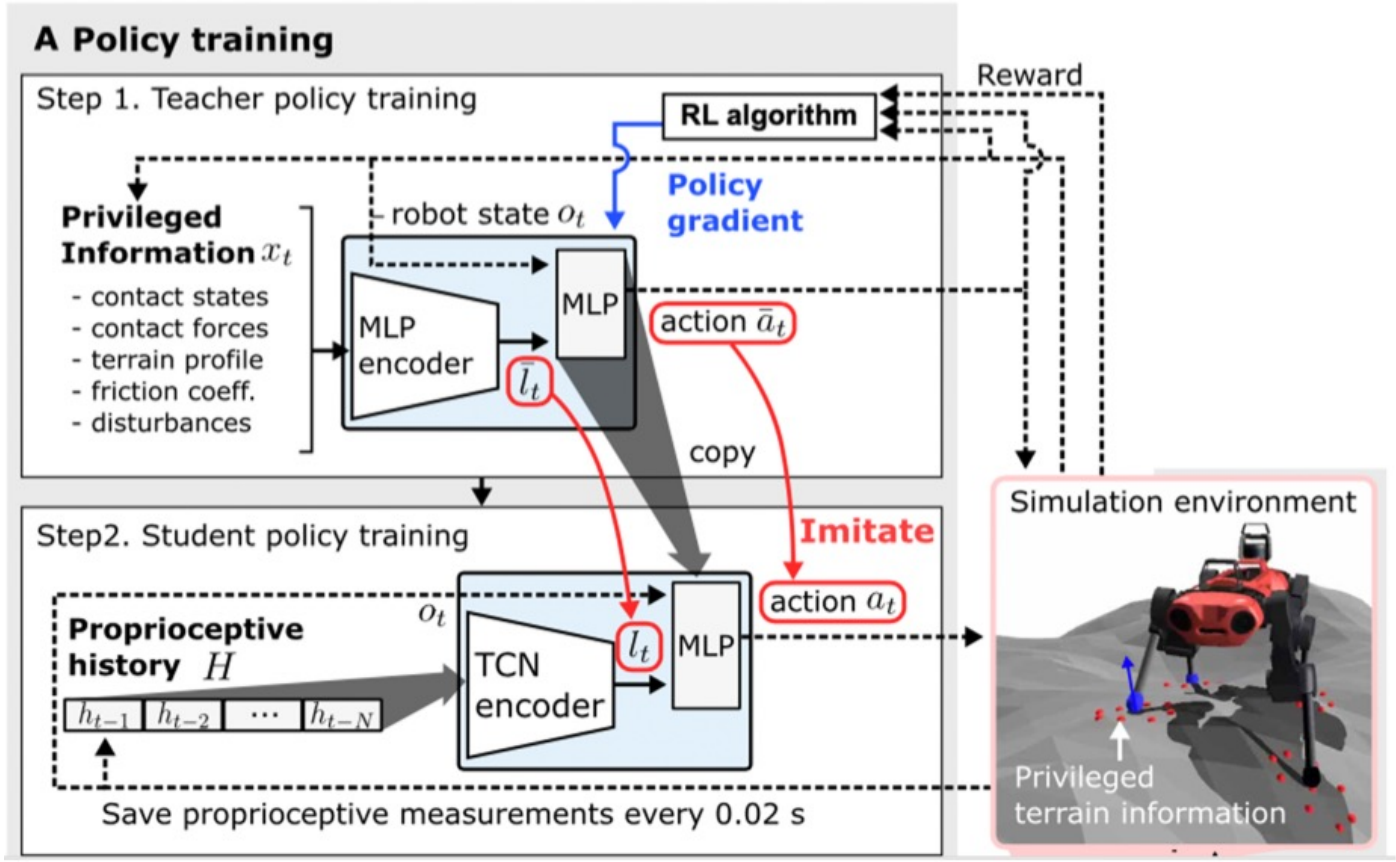
ANIMAL ROBOTS

Learning quadrupedal locomotion over challenging terrain

Joonho Lee[1]*, Jemin Hwangbo[1,2], Lorenz Wellhausen[1], Vladlen Koltun[3], Marco Hutter[1]

Learning to Adapt Example: Locomotion

- ❑ Privileged information contains almost everything available in sim (contact, terrain, fraction, disturbance, etc)
- ❑ Privileged teacher is trained using PPO
- ❑ Students only learn from proprioceptive history (IMU, joint angle, etc)

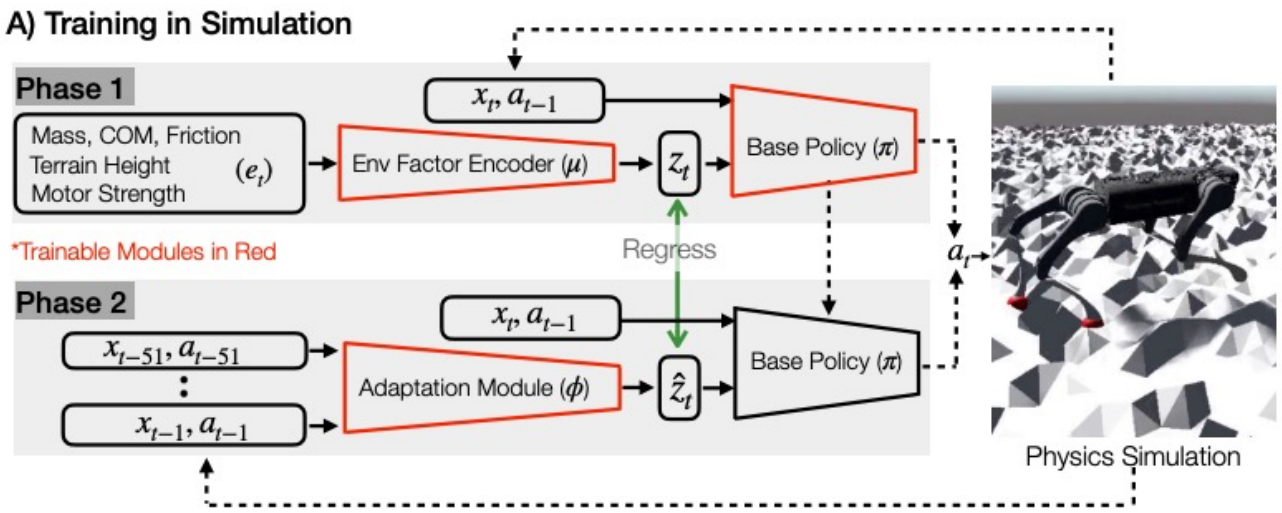


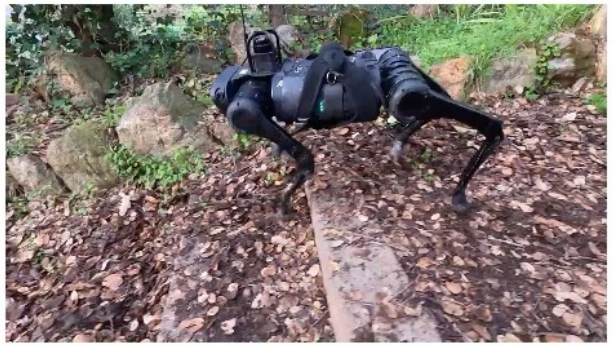
RMA: Student Learning in Latent Space

- ❑ Student learning doesn't have to happen in the action space
  - RMA: in latent space

RMA: Rapid Motor Adaptation for Legged Robots

Ashish Kumar, UC Berkeley
Zipeng Fu, Carnegie Mellon University
Deepak Pathak, Carnegie Mellon University
Jitendra Malik, UC Berkeley, Facebook





Variant: Asymmetric Actor-Critic

One-stage training:

- ❑ "Student" policy $\pi_s(x, \text{available info in the real})$
- ❑ "Teacher" critic $V(x, e)$
- ❑ Example: FALCON (humanoid loco-manip)
  - Actor: current time step's proprioception + 4-step history
  - Critic: Additionally have access to root vel & end effector force

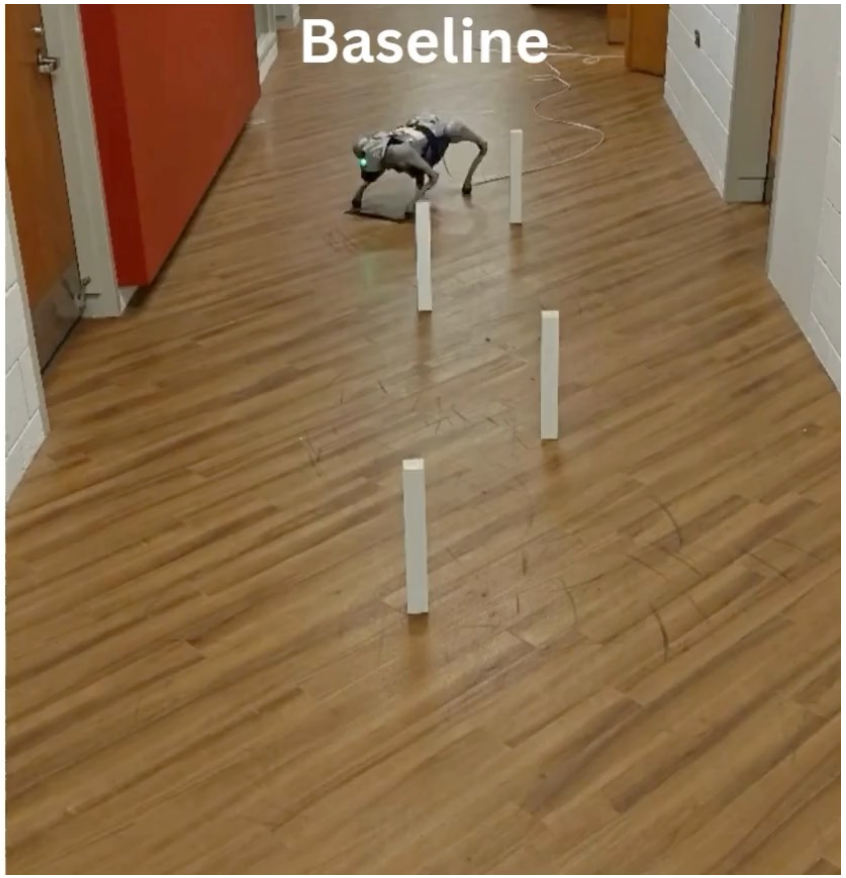
[FALCON: Learning Force-Adaptive Humanoid Loco-Manipulation, Zhang et al., L4DC'26]

Outline

- ❑ What is robot simulators? Why is sim2real useful but challenging?
- ❑ Methods to reduce the sim2real gap:
  - Domain randomization
  - Learning to adapt / teacher-student
  - Real2sim2real
- ❑ Advanced topics:
  - Human data + sim2real learning
  - RL algorithms for sim2real policy learning
- ❑ Zooming out: Sim2Real 1.0 to 4.0

Real2Sim by System ID

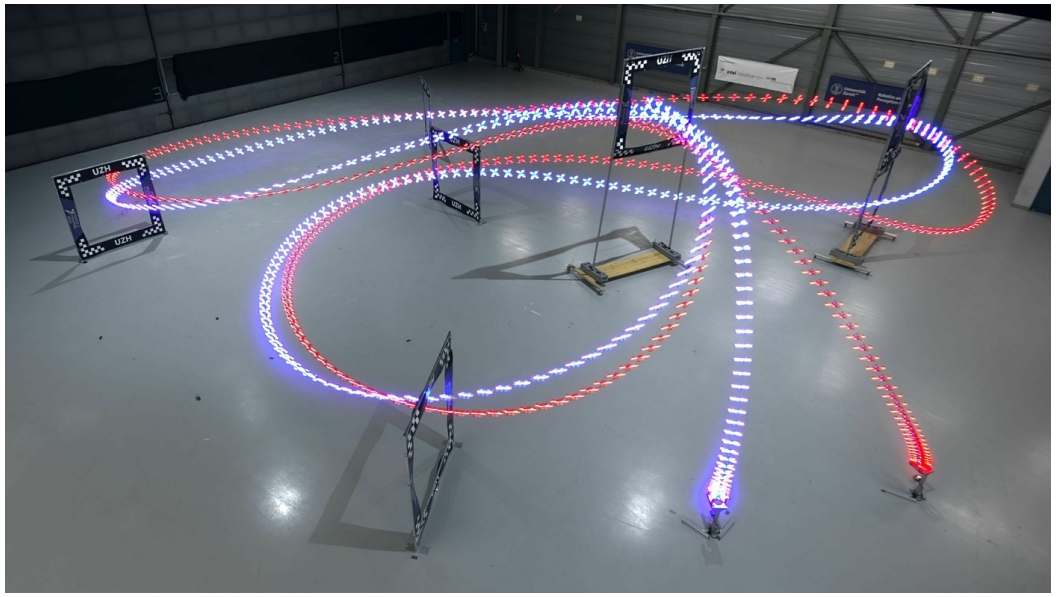
- **SPI-Active**: sampling-based system ID + active exploration
  - Use the policy that maximizes the Fisher Information to collect data
- https://lecar-lab.github.io/spi-active_/

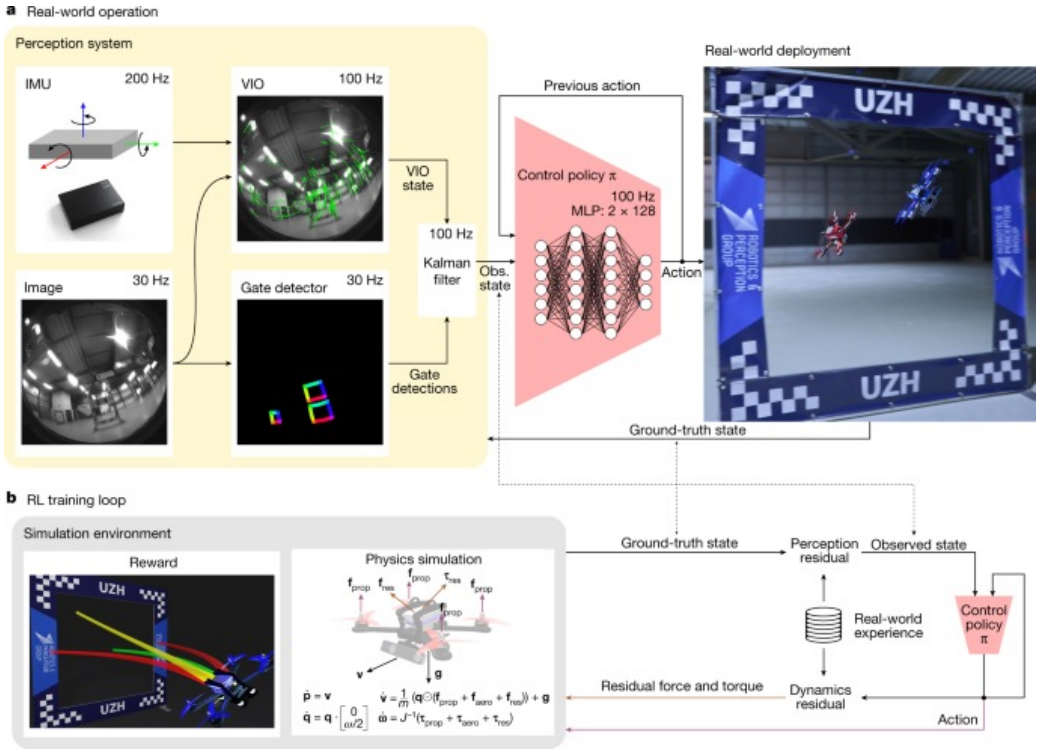


[Sampling-Based System ID with Active Exploration for Legged Robot Sim2Real Learning, Sobanbabu* and He* et al., CoRL'25]

Real2Sim by Learning Perception/Dynamics Residuals

- Pipelin: Use real data to “augment” a simulator, train DRL in simulation, deploy in the real world





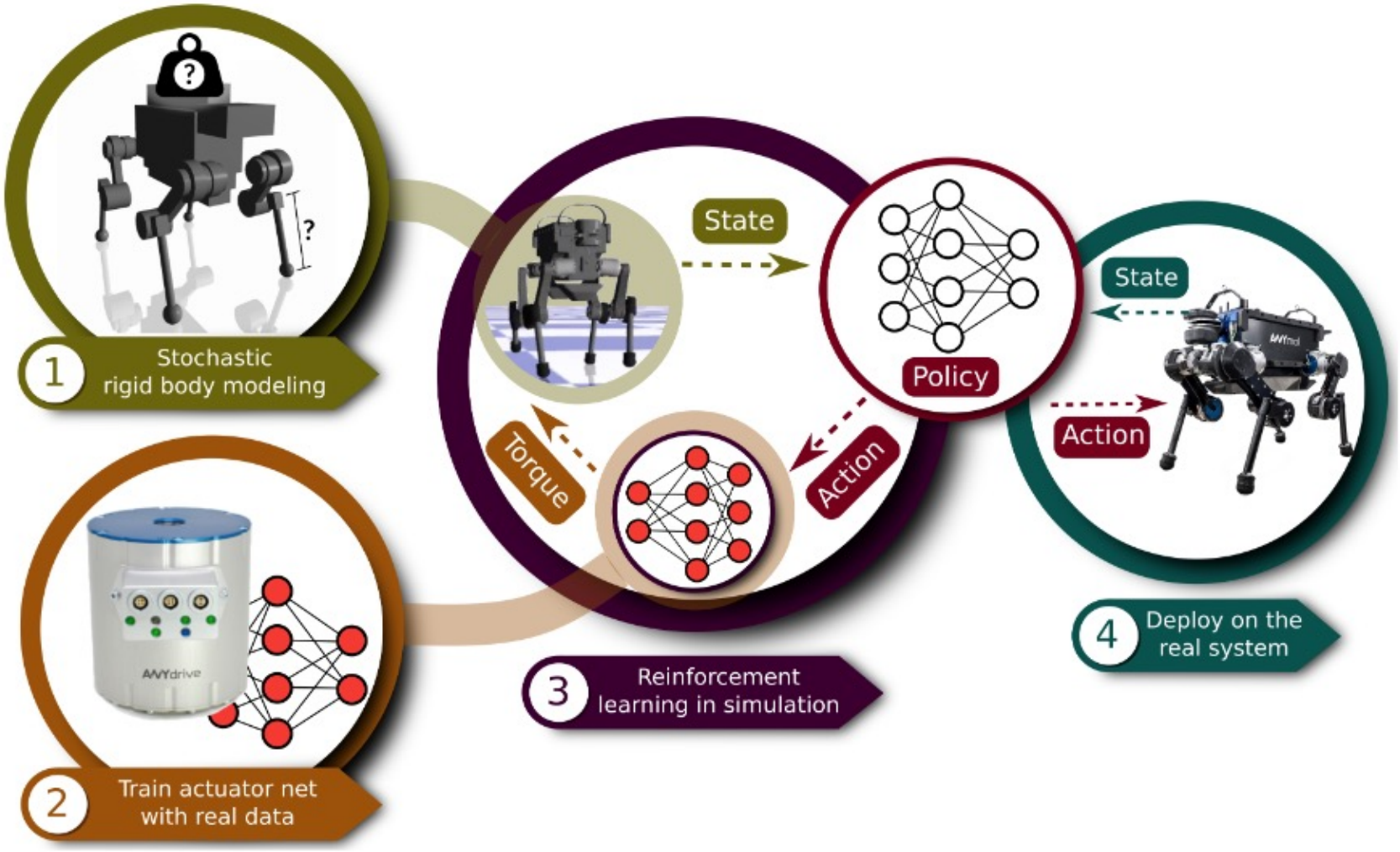
Champion-level drone racing using deep reinforcement learning

Elia Kaufmann ✉, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun & Davide Scaramuzza

Actuator Net for Locomotion

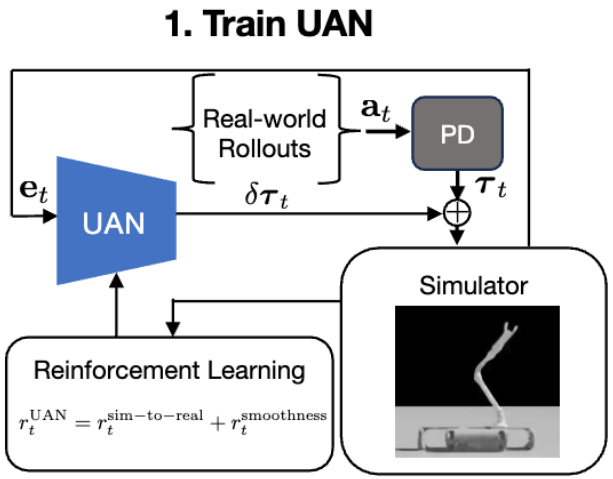
Learning agile and dynamic motor skills for legged robots

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



“Unsupervised” Actuator Net and Beyond

- ❑ Learning the actuator net needs torque labels
- ❑ Making it “unsupervised”: Use RL!
  - Train a “residual” torque model to match the real trajectory



Bridging the Sim-to-Real Gap for Athletic Loco-Manipulation

Nolan Fey, Gabriel B. Margolis, Martin Peticco, and Pulkit Agrawal
Improbable AI Lab
Massachusetts Institute of Technology, Cambridge, MA 02139

ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills

Tairan He[†1,2] Jiawei Gao[†1] Wenli Xiao[†1,2] Yuanhang Zhang[†1] Zi Wang[1] Jiashun Wang[1]
Zhengyi Luo[1,2] Guanqi He[1] Nikhil Sobanbab[1] Chaoyi Pan[1] Zeji Yi[1] Guannan Qu[1]
Kris Kitani[1] Jessica Hodgins[1] Linxi “Jim” Fan[2] Yuke Zhu[2] Changliu Liu[1] Guanya Shi[1]
[1]Carnegie Mellon University [2]NVIDIA [†]Equal Contributions
Page: https://agile.human2humanoid.com Code: https://github.com/LeCAR-Lab/ASAP

Outline

- ❑ What is robot simulators? Why is sim2real useful but challenging?
- ❑ Methods to reduce the sim2real gap:
  - Domain randomization
  - Learning to adapt / teacher-student
  - Real2sim2real
- ❑ Advanced topics:
  - Human data + sim2real learning
  - RL algorithms for sim2real policy learning
- ❑ Zooming out: Sim2Real 1.0 to 4.0

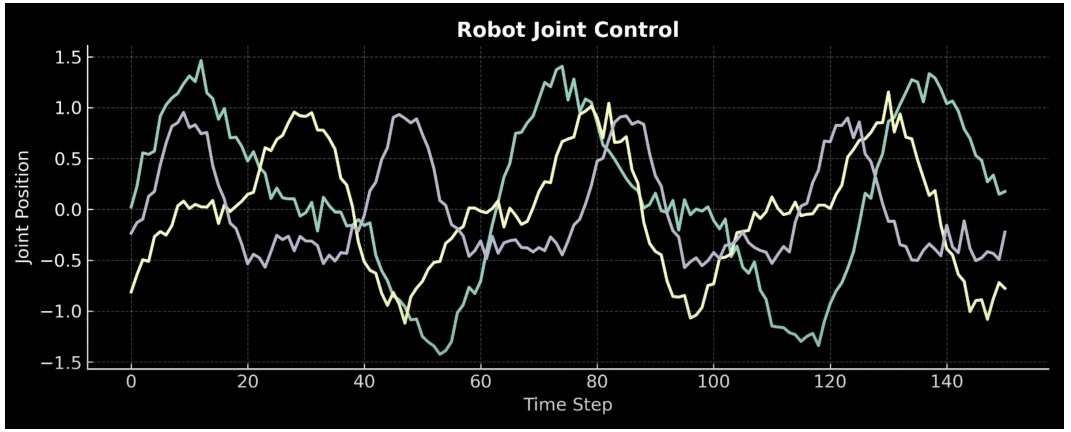
Sim2Real Skill Learning from Human Data

- ❑ Defining tasks/rewards in simulation could be very tricky
  - Relatively simple for locomotion
  - Hard for loco-manipulation & dexterous manipulation
- ❑ Why don't we leverage human data?
  - No free lunch! There is a "physics gap" between **human intents** and **robot actions**
  - Simulation can bridge this gap! I.e., physics grounding

"Human Intents"

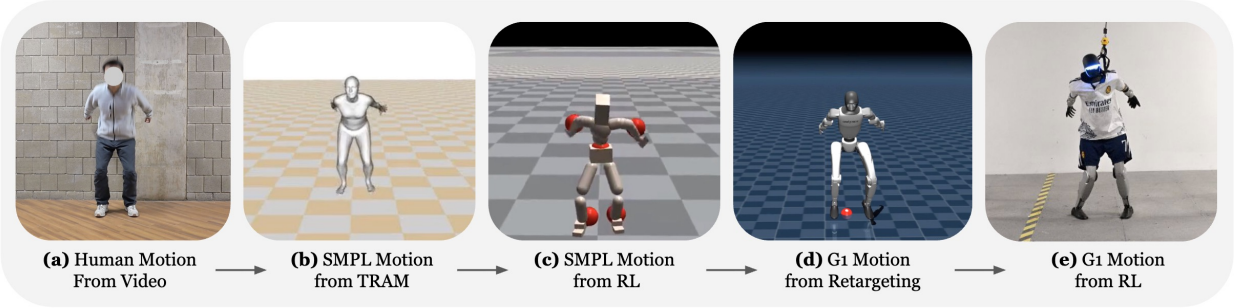
?

"Robot Actions"

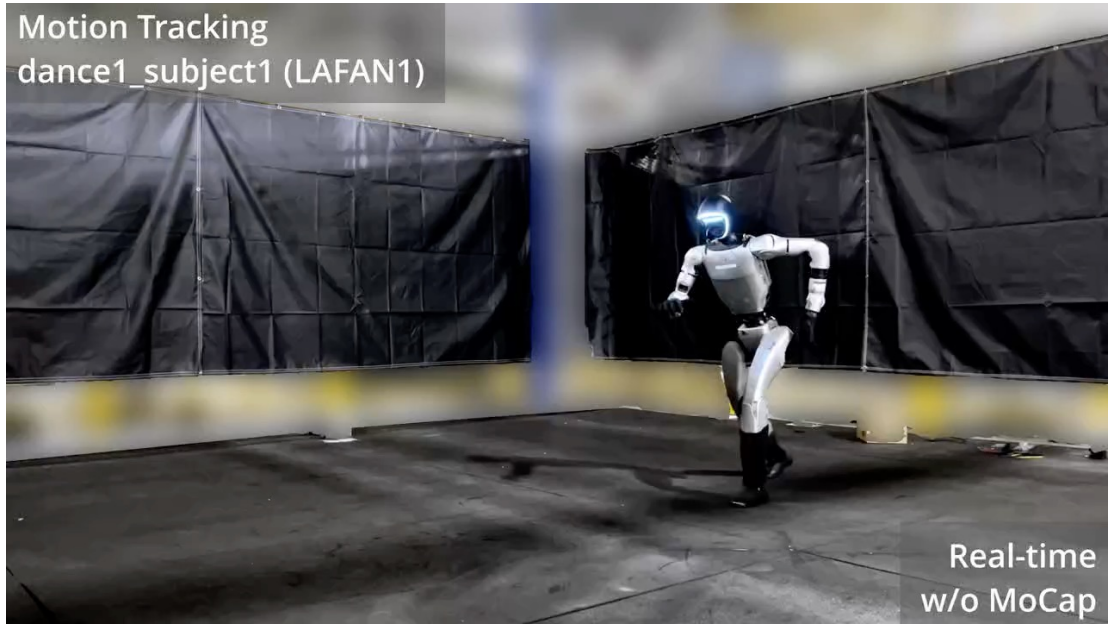


Retargeting + Policy Learning

- ❑ Step 1: Motion retargeting (typically kinematics-level)
- ❑ Step 2: Policy learning in simulation
- ❑ Example in whole-body tracking:



ASAP, Feb 2025 (RSS'25)
https://agile.human2humanoid.com/

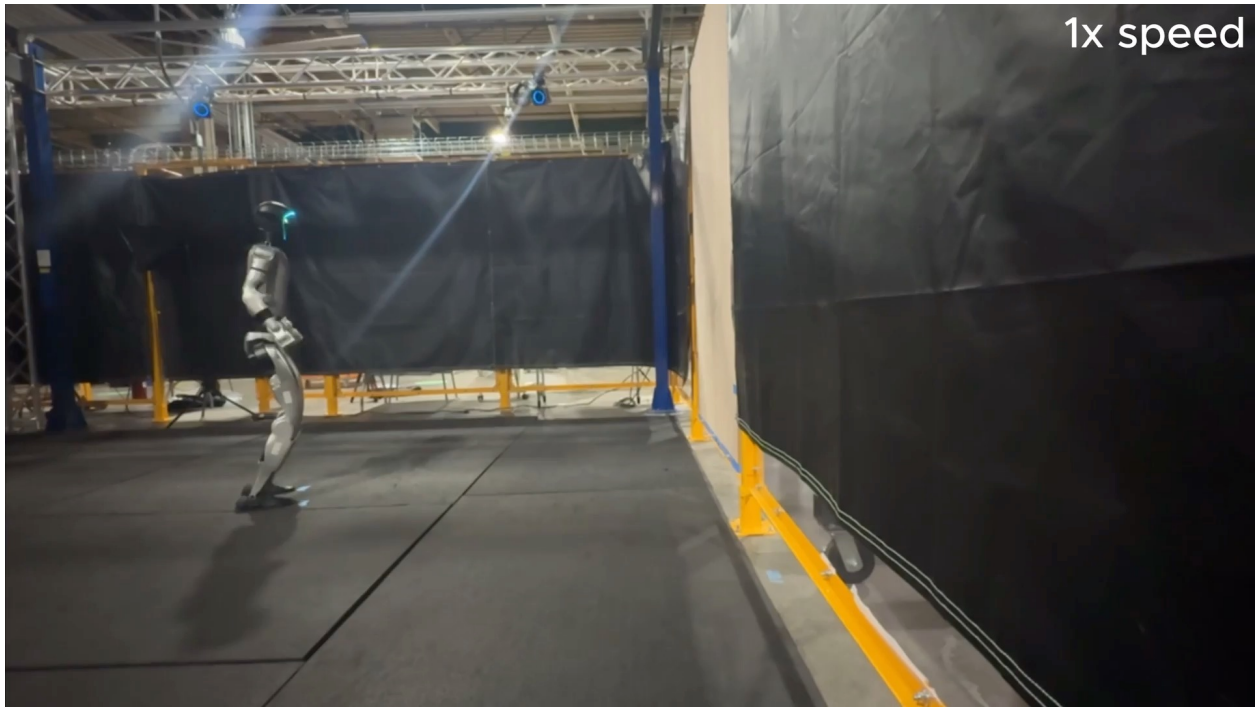


BeyondMimic, Aug 2025

Retargeting & Tracking for Loco-Manip & Scene Interaction: *OmniRetarget*

- ❑ More videos & datasets: https://omniretarget.github.io/
- ❑ In the video, robot only relies on proprioception

Parkour

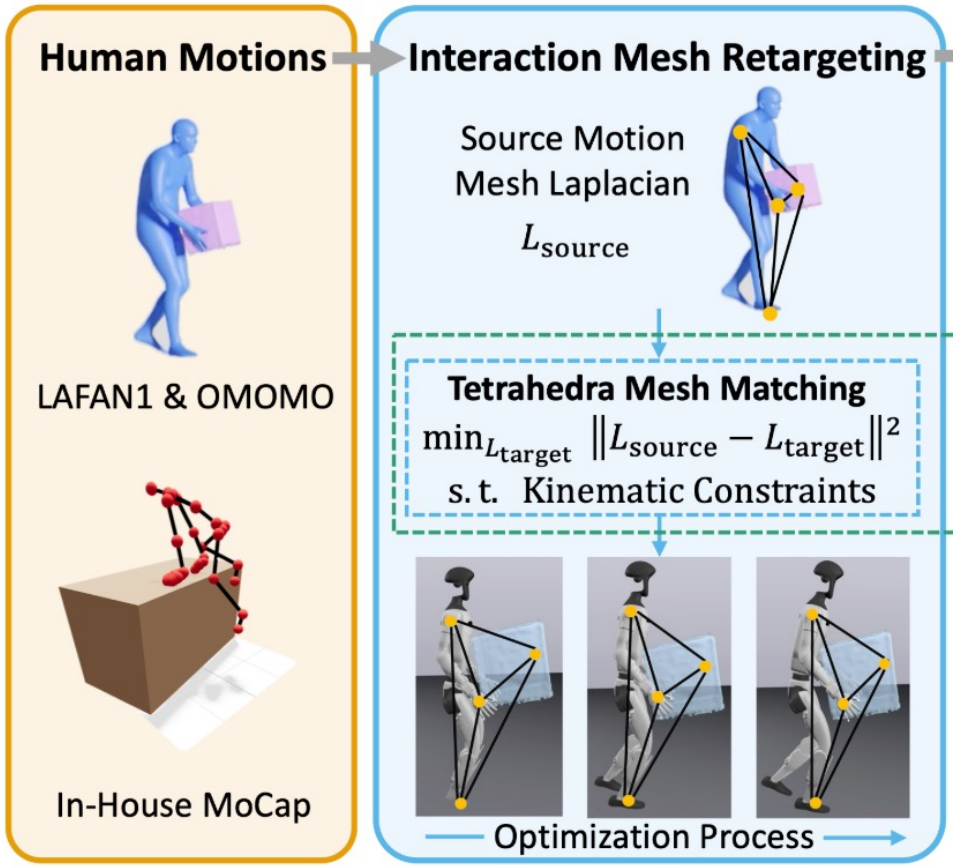


“Wall flip” (maximum angular rate 890 degrees / second)

[OmniRetarget: Interaction-Preserving Data Generation for Humanoid Whole-Body Loco-Manipulation and Scene Interaction, ICRA’26] *Work done at Amazon FAR*

Retargeting & Tracking for Loco-Manip & Scene Interaction: *OmniRetarget*

- ❑ Need to jointly consider the object and the robot!
- ❑ Interaction-mesh-based retargeting



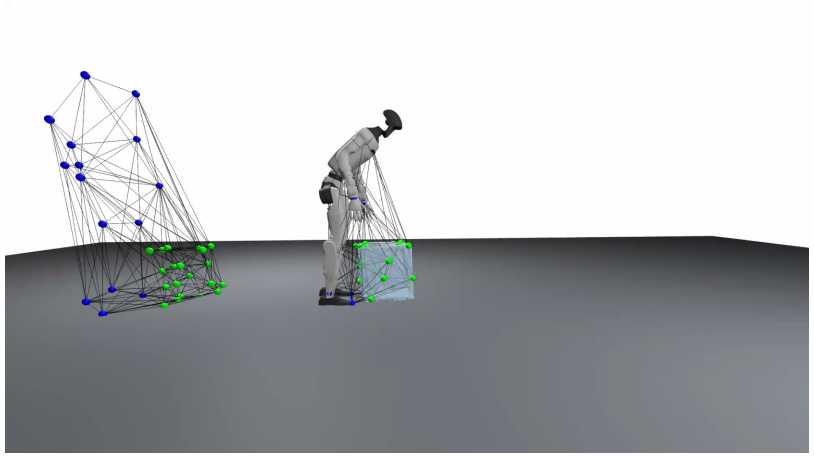
$$q_t^\star = \arg\min_{q_t} \sum_i \|L(p_{t,i}^{\text{source}}) - L(p_{t,i}^{\text{target}}(q_t))\|^2 + \|q_t - q_{t-1}\|_Q^2 \quad (3a)$$

$$\text{s.t.} \quad \phi_j(q_t) \geq 0, \forall j \quad (3b)$$

$$q_{\min} \leq q_t \leq q_{\max} \quad (3c)$$

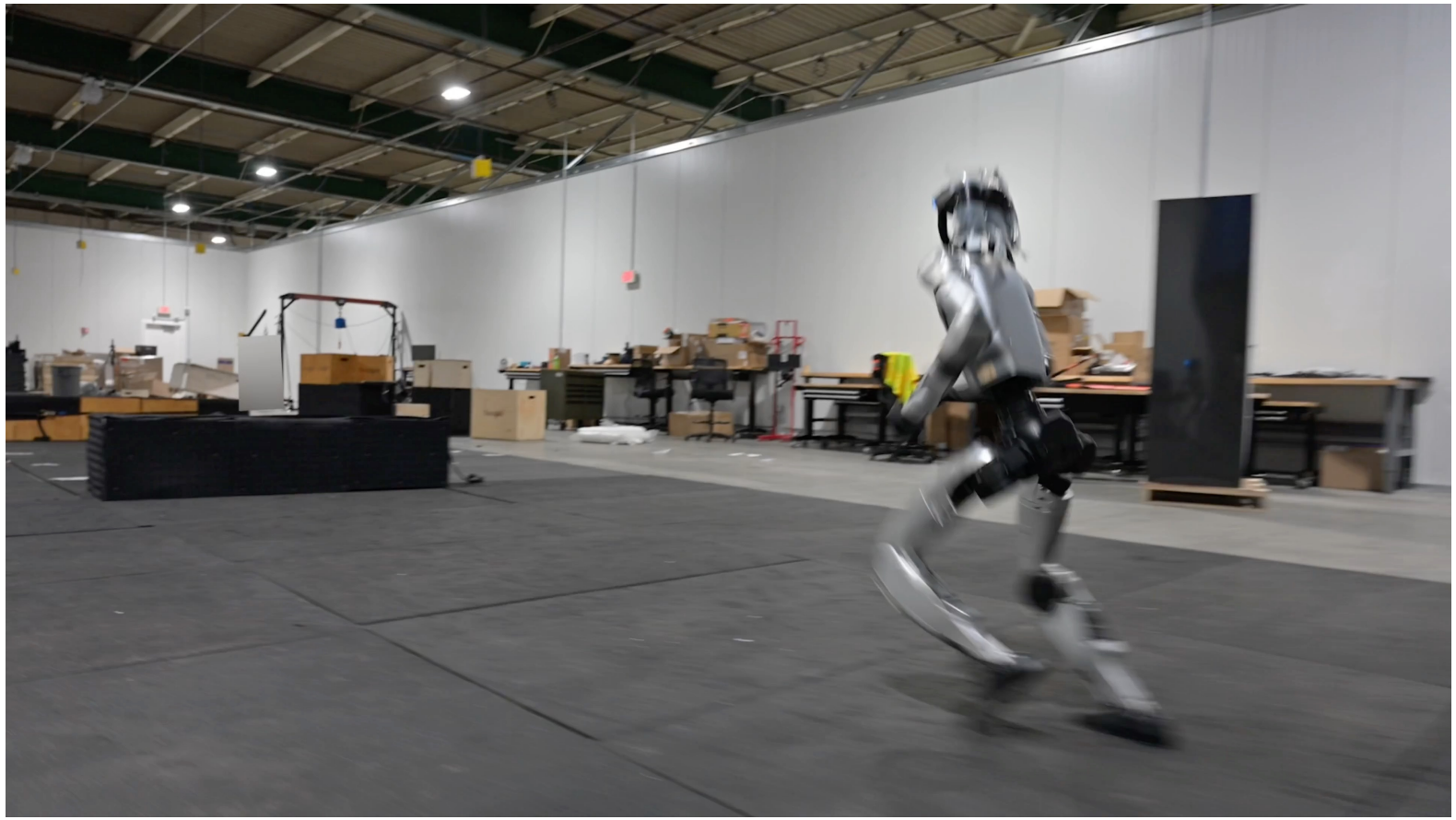
$$v_{\min} \cdot dt \leq q_t - q_{t-1} \leq v_{\max} \cdot dt \quad (3d)$$

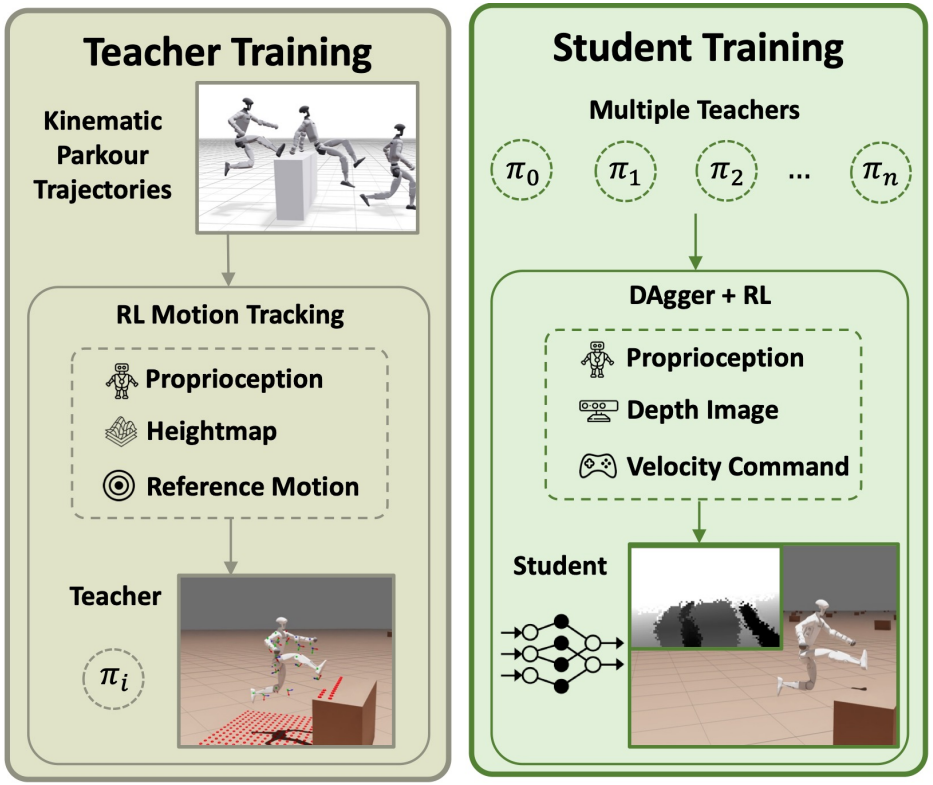
$$p_t^F = p_{t-1}^F, \forall \text{stance foot}, \quad (3e)$$



[OmniRetarget: Interaction-Preserving Data Generation for Humanoid Whole-Body Loco-Manipulation and Scene Interaction, ICRA'26] *Work done at Amazon FAR*

Extend to Perceptive Settings





[Perceptive Humanoid Parkour: Chaining Dynamic Human Skills via Motion Matching, Wu* & Yang* & Huang* et al., RSS'26]

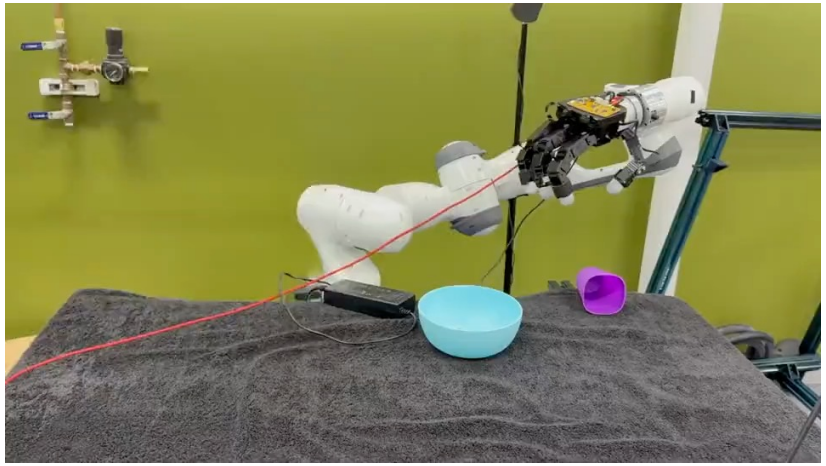
Work done at Amazon FAR

Dynamically Feasible Retargeting: *SPIDER*

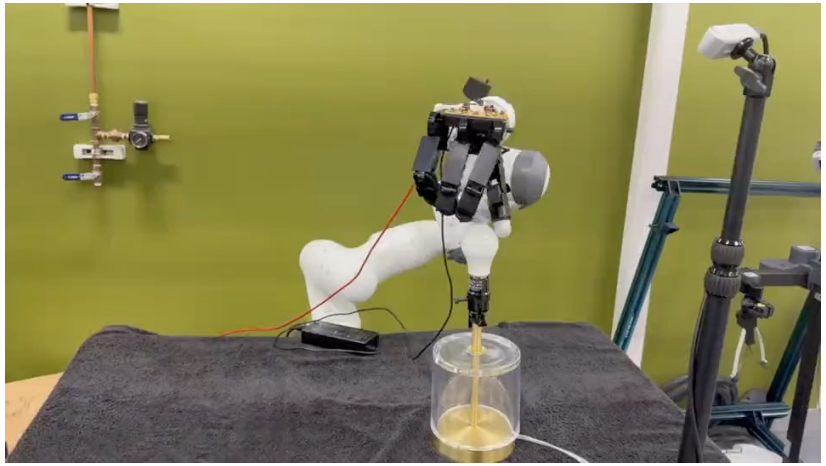
❑ *SPIDER*: Dynamics-level retargeting

- Retargeting as an optimal control problem and solve it by sampling-based methods

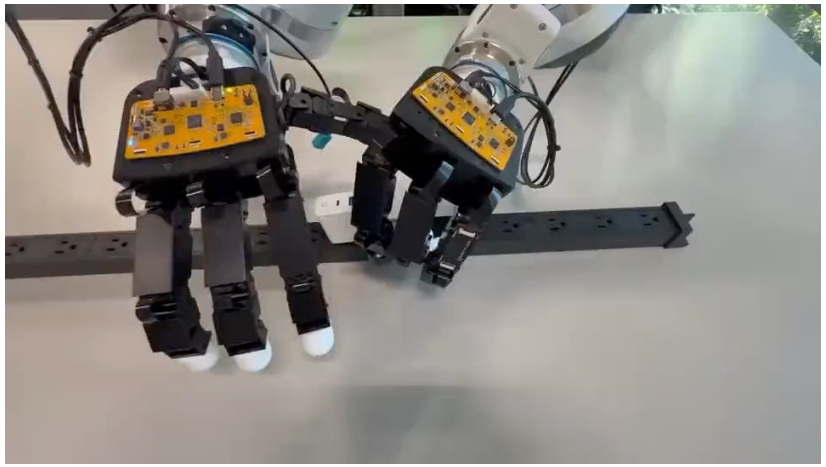
All these demos are direct open-loop replay of the retargeted action trajectories in real!



Pick cup



Rotate light bulb



Unplug charger

[SPIDER: Scalable Physics-Informed DExterous Retargeting, Pan et al.]

Outline

- ❑ What is robot simulators? Why is sim2real useful but challenging?
- ❑ Methods to reduce the sim2real gap:
  - Domain randomization
  - Learning to adapt / teacher-student
  - Real2sim2real
- ❑ Advanced topics:
  - Human data + sim2real learning
  - RL algorithms for sim2real policy learning
- ❑ Zooming out: Sim2Real 1.0 to 4.0

RL Algorithms for Sim2Real Policy Learning

- ❑ On-policy PG methods like PPO is very effective!

Other methods:

- ❑ Better leverage massively parallel envs: SAPG (split ang aggregate policy gradient)
- ❑ Off-policy methods for sim2real: FastTD3/FastSAC

Learning Sim-to-Real Humanoid Locomotion in 15 Minutes

Younggyo Seo* Carmelo Sferrazza* Juyue Chen
Guanya Shi Rocky Duan Pieter Abbeel

Amazon FAR (Frontier AI & Robotics)

RL Algorithms for Sim2Real Policy Learning

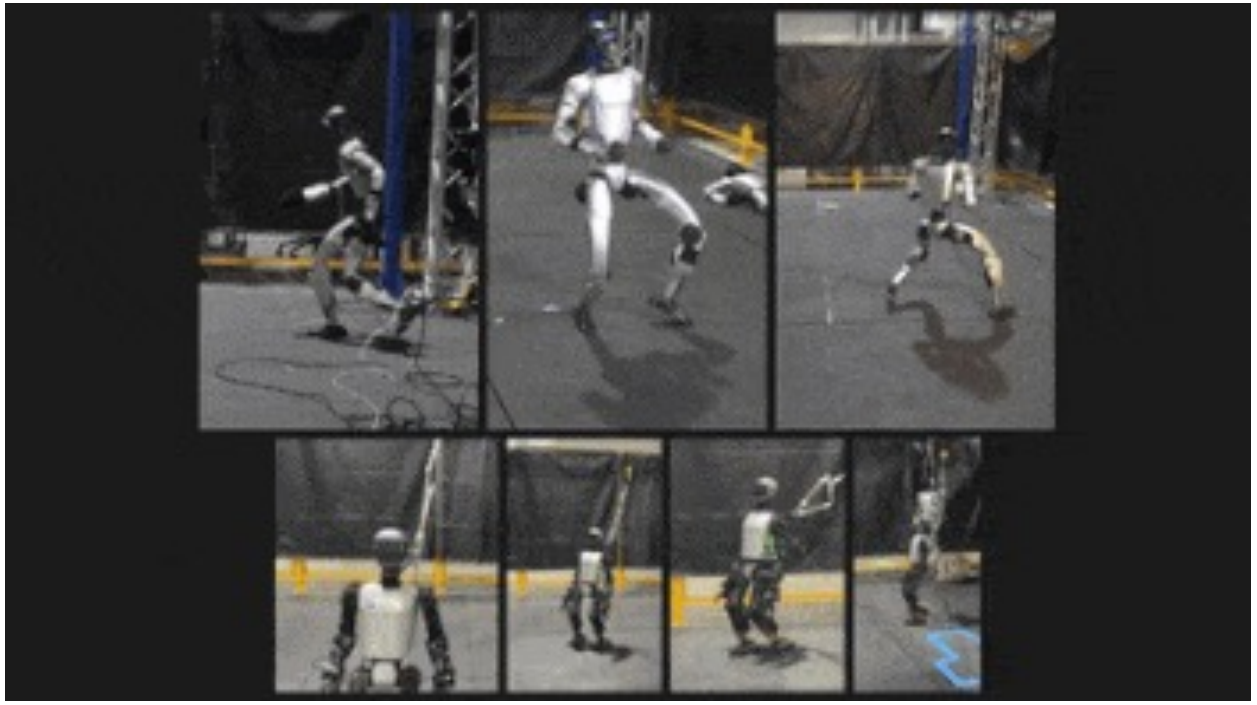
- ❑ PG for flow matching policies: FPO and FPO++

Flow Policy Gradients for Robot Control

Brent Yi[12†*] Hongsuk Choi[12†*] Himanshu Gaurav Singh[12†] Xiaoyu Huang[12†]
Takara E. Truong[13†] Carmelo Sferrazza[1] Yi Ma[24] Rocky Duan[1‡]
Pieter Abbeel[12‡] Guanya Shi[15‡] Karen Liu[13‡] Angjoo Kanazawa[12‡]

[1]Amazon FAR [2]UC Berkeley [3]Stanford [4]HKU [5]CMU

*Equal Contribution †Work done as an intern at Amazon FAR ‡Amazon FAR team co-lead



RL Algorithms for Sim2Real Policy Learning

- ❑ Unsupervised RL with forward-backward representation: BFM-Zero
- ❑ It can optimize any user-specified reward function at test time

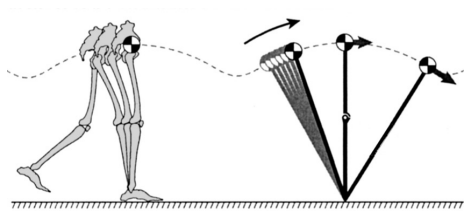
[BFM-Zero: A Promptable Behavioral Foundation Model for Humanoid Control Using Unsupervised Reinforcement Learning, Li* and Luo*, ICLR'26]

Outline

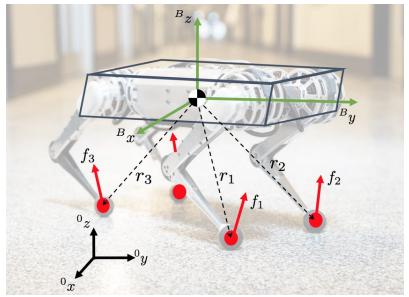
- ❑ What is robot simulators? Why is sim2real useful but challenging?
- ❑ Methods to reduce the sim2real gap:
  - Domain randomization
  - Learning to adapt / teacher-student
  - Real2sim2real
- ❑ Advanced topics:
  - Human data + sim2real learning
  - RL algorithms for sim2real policy learning
- ❑ Zooming out: Sim2Real 1.0 to 4.0

Sim2Real Has A Long History

- ❑ The control community has been doing sim2real for many decades!
- ❑ Sim2real 1.0:



Inverted pendulum model



Single rigid body model

"Simulator": Reduced-ordered models

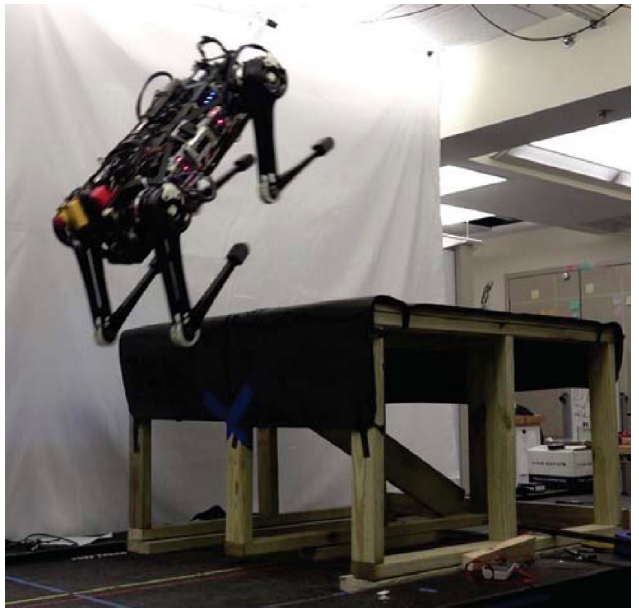
$$\min_{\boldsymbol{x}} \sum_{k=0}^{N-1} l(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) + l_N(\boldsymbol{x}_N, \boldsymbol{\theta})$$

subject to

$$\boldsymbol{x}_0 = \hat{\boldsymbol{x}}$$
$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) \; \forall \; k \in \{0, ..., N-1\}$$
$$\boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) \leq \boldsymbol{0} \; \forall \; k \in \{0, ..., N-1\}$$
$$\boldsymbol{g}_N(\boldsymbol{x}_N, \boldsymbol{\theta}) \leq \boldsymbol{0},$$

Online model predictive control

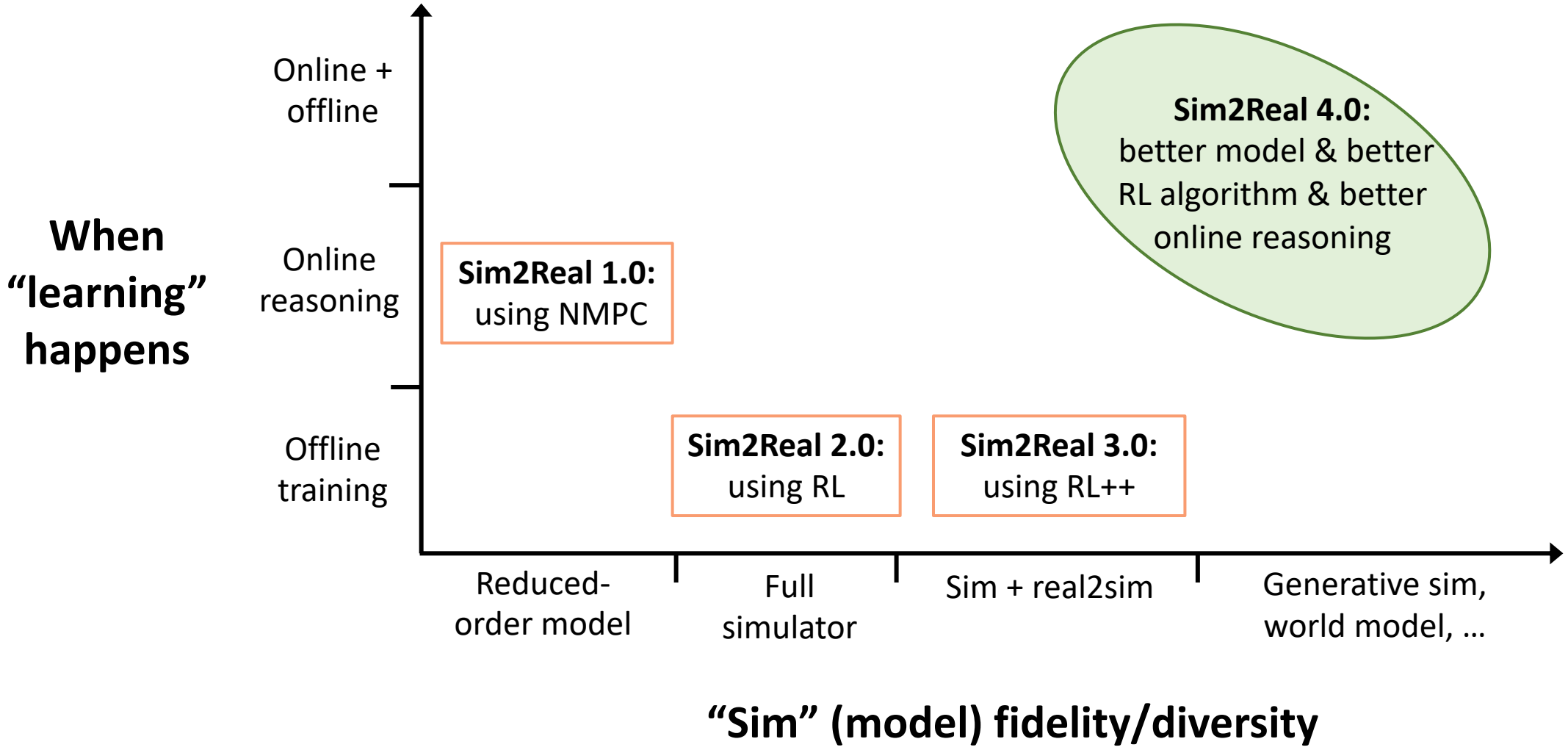
a_t



[Nguyen et al., ICRA'19]

- ❑ What is fascinating (but also strange): no "pretraining", 100% rely on very fast (>100Hz) online reasoning

Zooming Out: Towards Sim2Real 4.0



Thank You!

- ❑ Topics we didn’t cover today:
  - Sim & real co-training
  - Simulation for policy evaluation
  - Differentiable simulation
  - …
- ❑ All selected papers are open-sourced!
- ❑ Humanoid sim2real framework: https://github.com/amazon-far/holosoma
  - Regularly maintained and updated
  - Retargeting, policy training, deployment