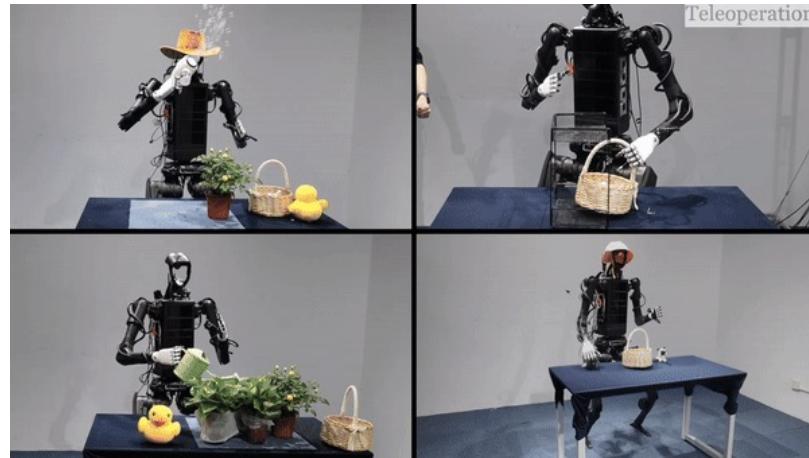


# From *Sim2Real* 1.0 to 4.0 for Humanoid Whole-Body Control and Loco-Manipulation



OmniH2O (CoRL'24)  
<https://omni.human2humanoid.com/>



ASAP (RSS'25)  
<https://agile.human2humanoid.com/>



FALCON (under review)  
<https://lecar-lab.github.io/falcon-humanoid/>

**Guanya Shi**

Assistant Professor, Robotics Institute, CMU

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

# Teleoperation or Learning from Videos Seems Really Promising

- ❑ Basic receipt: behavior cloning from labeled actions
  - Action space: typically joint angle or end-effector pose
  - Low-level control is easy and accurate (PD or IK + PD)

Physical Intelligence  $\pi$ 0.5

Learning from teleoperation data



Tesla Optimus

Learning from mixed teleoperation & human video data



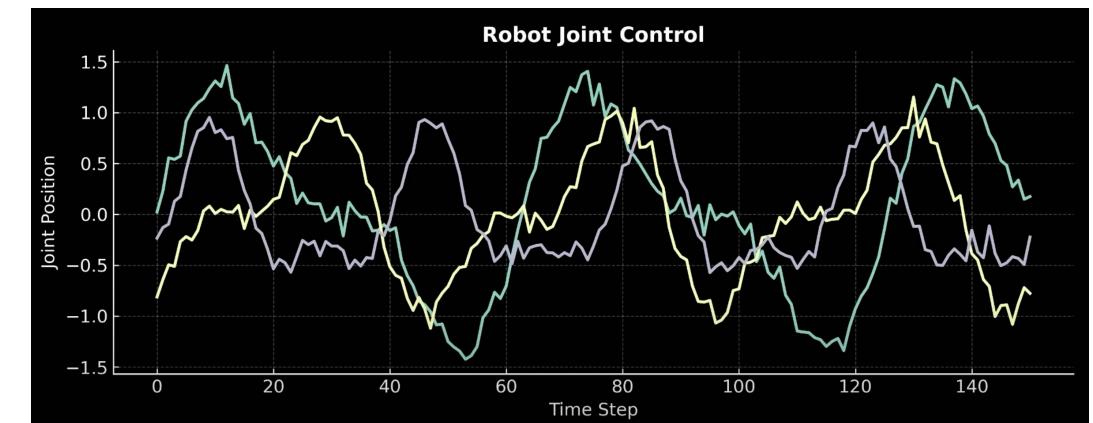
# Teleoperation or Learning from Videos Seems Really Promising



Humanoid Policy ~ Human Policy  
(human data and humanoid data co-training)  
<https://human-as-robot.github.io/>

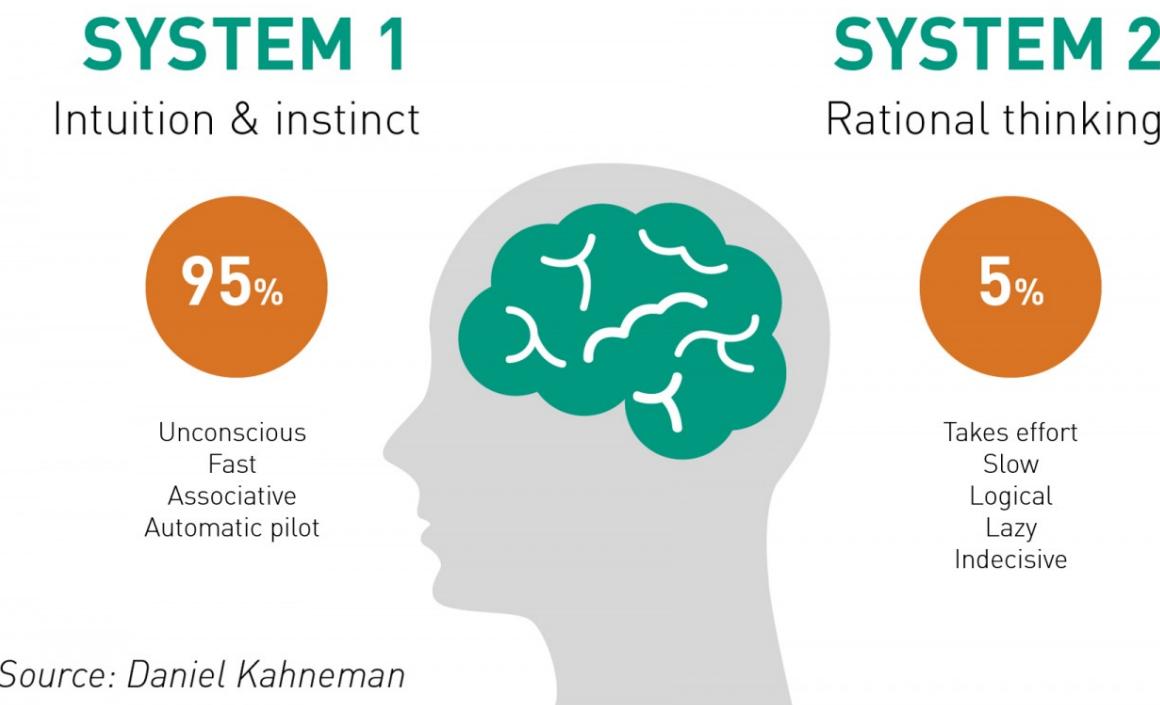
# ... How About Tasks Involving Whole-Body Agility?

- For those tasks, *impossible or extremely hard* to:
  - Teleoperate (if you can, you already solve the problem)
  - Get labeled action (imagine ask MJ: “I wanna learn fadeaway jumper. Could you tell me your joint trajectories?”)
  - Use simple low-level controllers for tracking



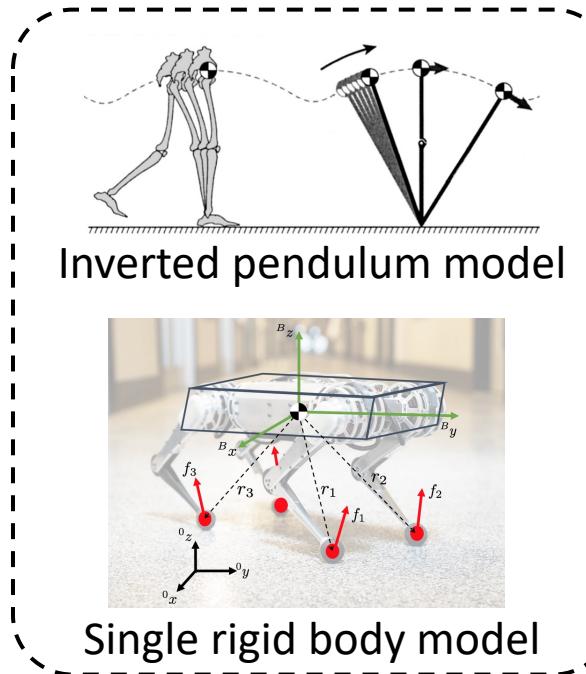
# ... How About Tasks Involving Whole-Body Agility?

- ❑ Most dexterity and agility (especially whole-body) come from **system 1**
- ❑ (I suspect) most human demonstrations involve little or 0 system 1
- ❑ How to learn **system 1** agility and dexterity?
  - We need a “model/simulator” and **sim2real** learning!



# Sim2Real 1.0: Simplified Model + Online Reasoning

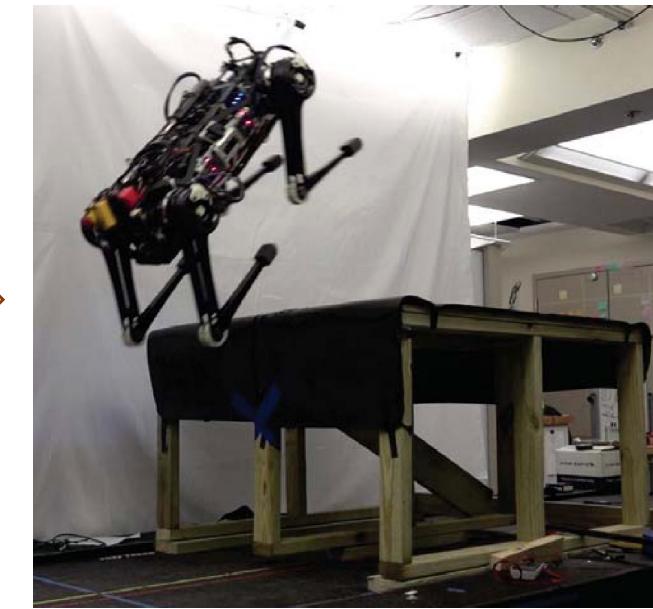
- The control community has been doing sim2real for many decades!



The diagram shows a flow from the simulator section to the control section. An orange arrow points from the "Simulator" section to the "Online model predictive control" section. Inside the "Online model predictive control" section, there is a mathematical optimization problem:

$$\begin{aligned} & \min_{\boldsymbol{x}} \sum_{k=0}^{N-1} l(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) + l_N(\boldsymbol{x}_N, \boldsymbol{\theta}) \\ \text{subject to} \\ & \boldsymbol{x}_0 = \hat{\boldsymbol{x}} \\ & \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) \quad \forall k \in \{0, \dots, N-1\} \\ & \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) \leq \mathbf{0} \quad \forall k \in \{0, \dots, N-1\} \\ & \boldsymbol{g}_N(\boldsymbol{x}_N, \boldsymbol{\theta}) \leq \mathbf{0}, \end{aligned}$$

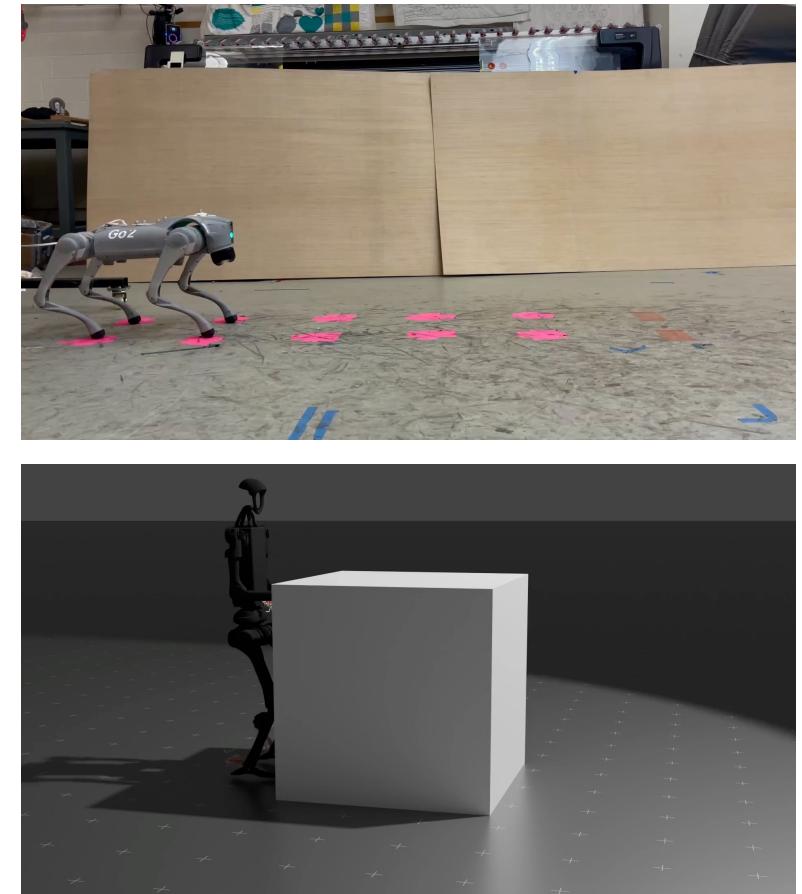
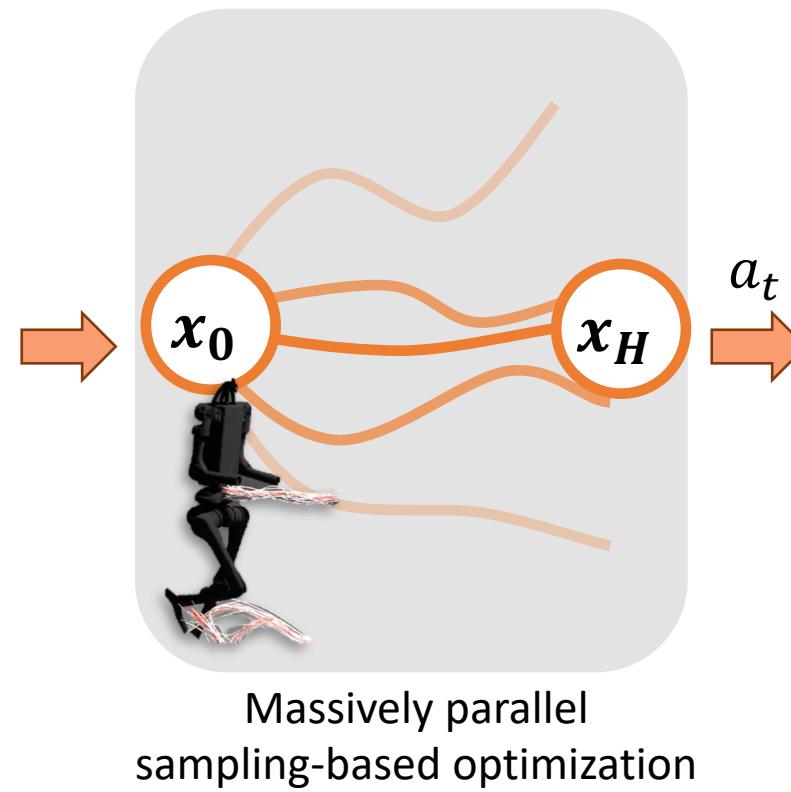
An orange arrow points from the "Online model predictive control" section to a photograph of a real-world robot.



- What is fascinating (but also foolish): no “pretraining”, 100% rely on very fast (>100Hz) online reasoning

# Sim2Real 1.5: Full-Order Simulators + Online Reasoning

- We can do full-order MPC now using advanced sampling-based methods (e.g., DIAL-MPC)
- However, slow and require state estimation



DIAL-MPC

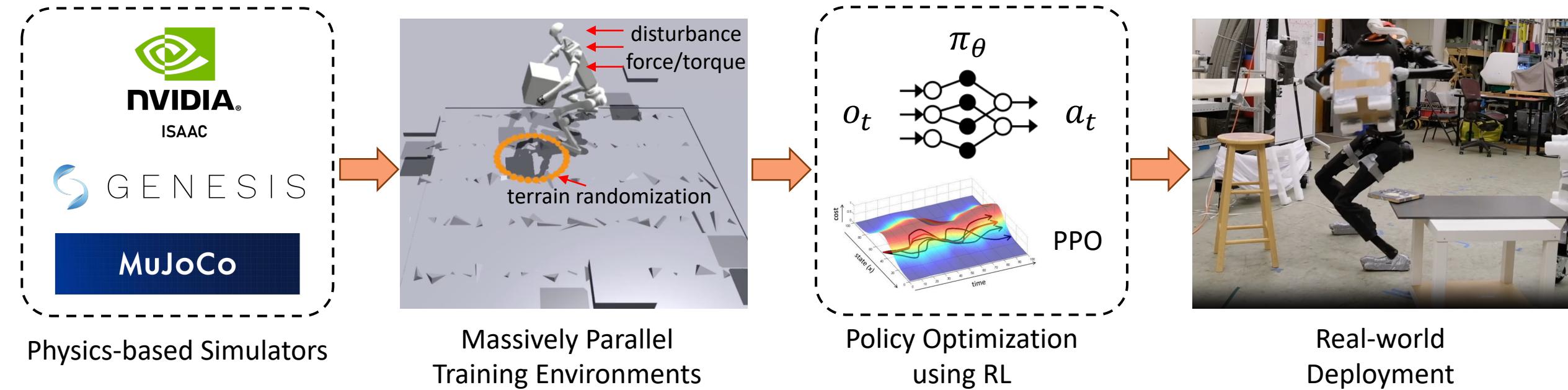
<https://lecar-lab.github.io/dial-mpc/>

[Xue\* and Pan\* et al., ICRA'25]

Best Paper Award Finalist

# Sim2Real 2.0: Sim2Real Reinforcement Learning (RL)

- ❑ Massively parallel policy gradient method (PPO) is such a strong policy optimizer
- ❑ No need for state estimation! Observation  $o_t$  is all you need



# H2O: Human-to-Humanoid Whole-Body Control

- **Goal:** Build an interface between whole-body human and humanoid motions
- Such an interface supports human whole-body teleoperation, imitation learning, integrating with VLMs, ...
- **Key idea of H2O:** Sim2Real 2.0 from large-scale retargeted human motion dataset



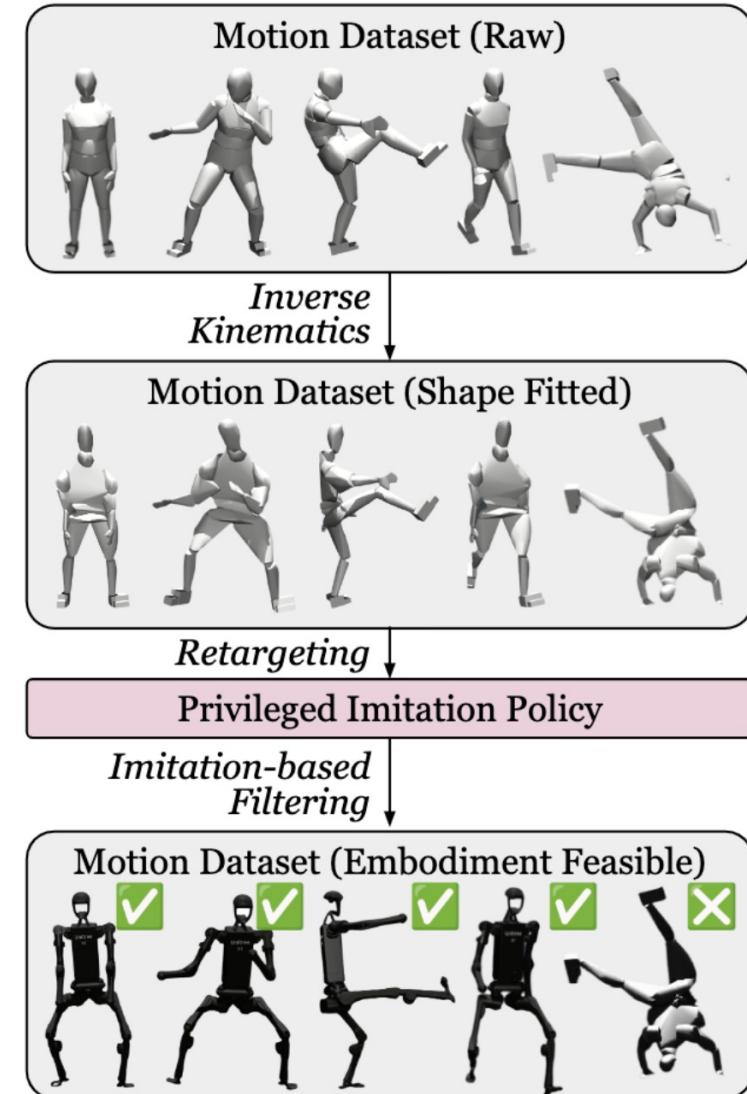
[Learning Human-to-Humanoid Real-Time Whole-Body Teleoperation, He\* & Luo\* et al., IROS'24 (**Oral**)]

[OmniH2O: Universal and Dexterous Human-to-Humanoid Whole-Body Teleoperation and Learning, He\* & Luo\* & He\*, et al., CoRL'24]

# H2O: Human-to-Humanoid Whole-Body Control

## Step 1: Create a large-scale humanoid-feasible motion dataset

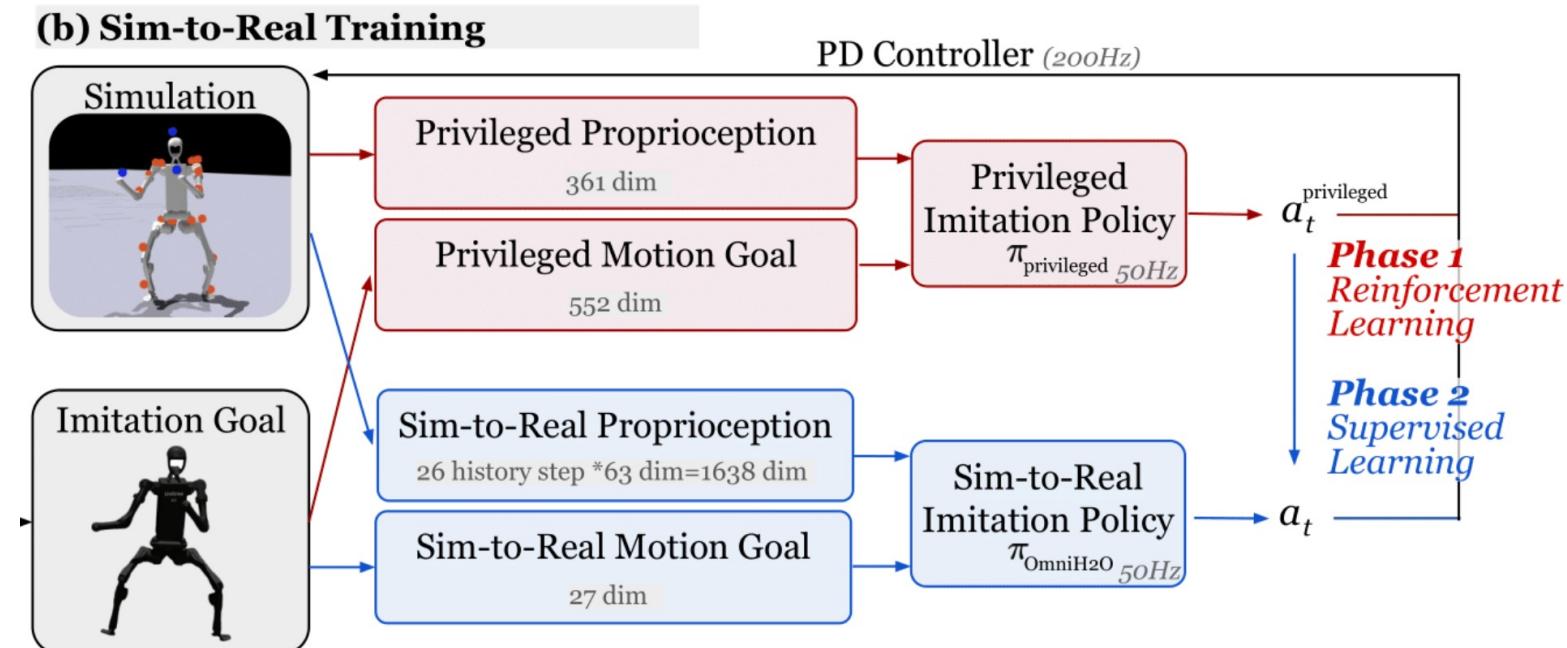
- >10K human motions from AMASS (ICCV'19)!
- Shape fitting using inverse kinematics
- Key: physics-based retargeting
  - Learn a **privileged tracking policy** to track all motions using RL
  - This policy knows all states
  - Generate *humanoid-feasible* motions and filter out impossible motions



# H2O: Human-to-Humanoid Whole-Body Control

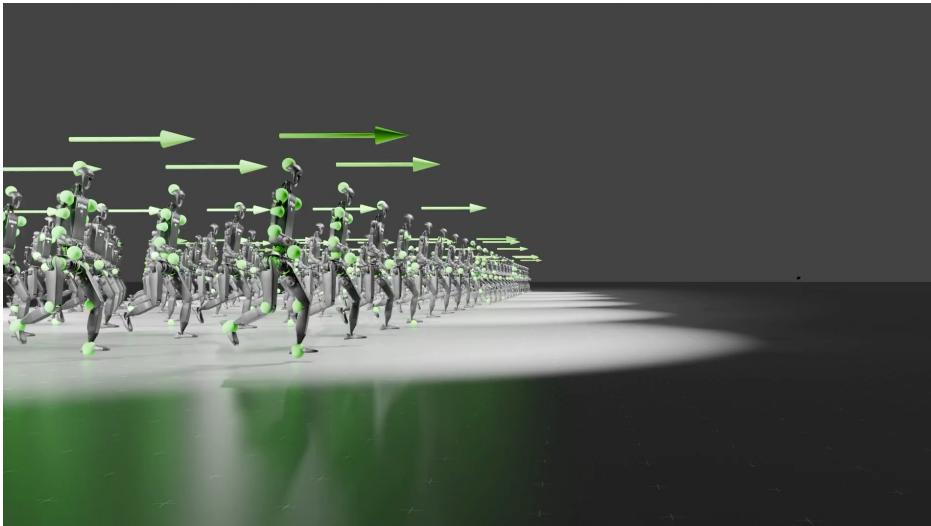
## Step 2: Sim2Real RL training

- ❑ Distill the **privileged tracking policy** to a **deployable student policy** in sim
  - The **student policy** only knows observations available in real
  - Key points as the motion goal (one head + two hands) for **student policy**
  - Domain randomization (DR) for robustness



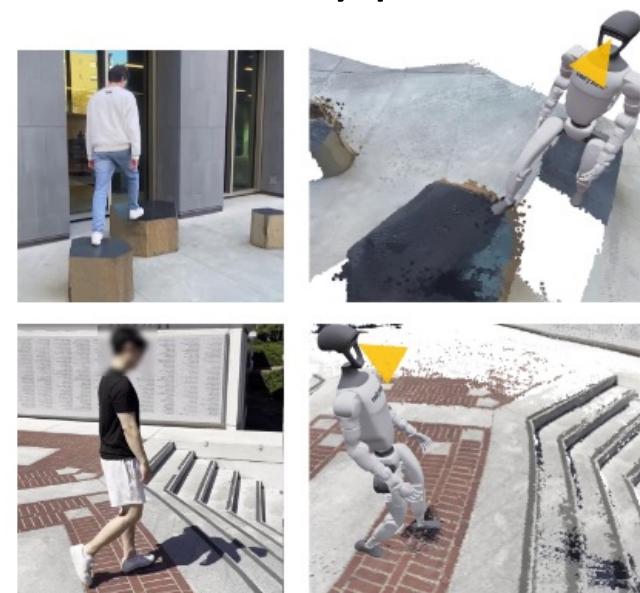
# H2O: Human-to-Humanoid Whole-Body Control

- The H2O pipeline is highly extendable
  - *Step 1:* motion retargeting
  - *Step 2:* privileged teacher policy learning for motion tracking
  - *Step 3:* distill to a student policy that can be deployed in the real
- Motion source in step 1 is flexible: MoCap (AMASS), videos, ...
- The student policy in step 3 is very flexible: Track different key points, vision-based, ...



One teacher -> multiple students

HOVER from NVIDIA [He\* and Xiao\* et al., ICRA'25]

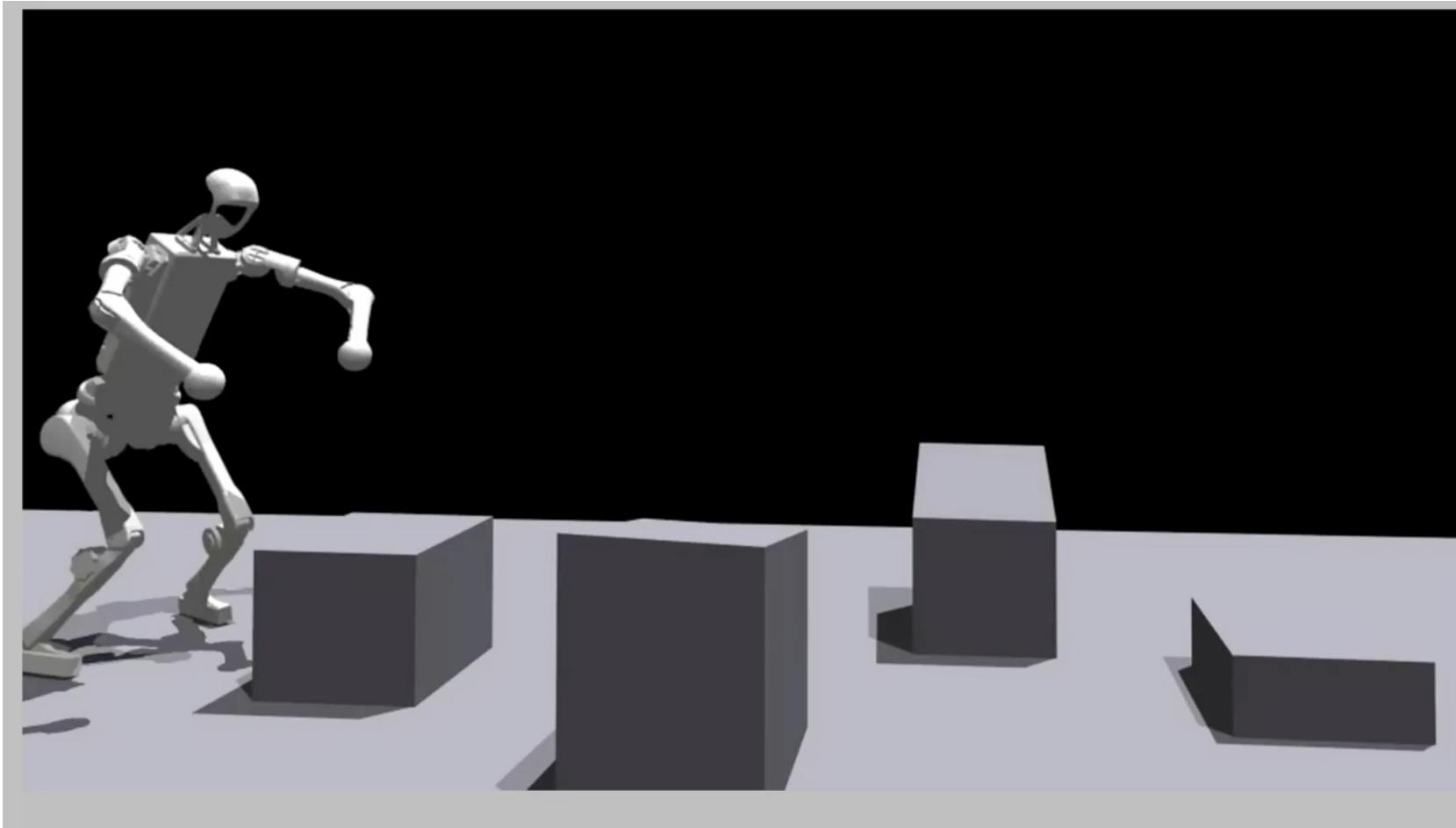


VideoMimic from Berkeley  
[Allshire\*, Choi\*, Zhang\*, McAllister\*, et al.]

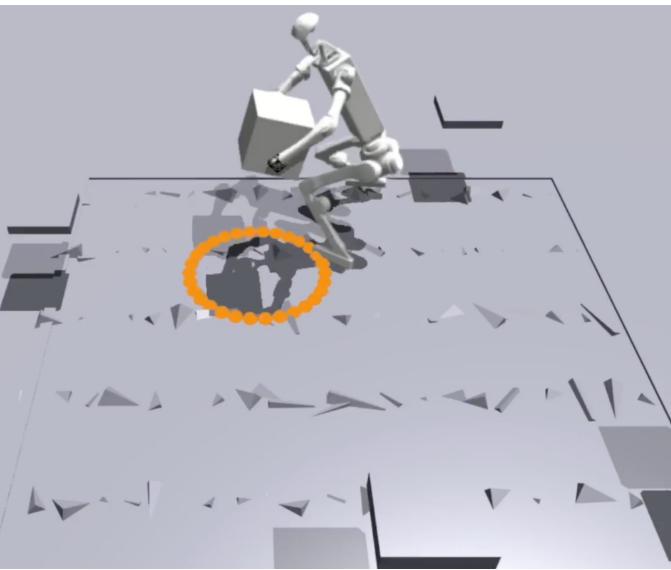
# WoCoCo: Learning Whole-Body Control with Sequential Contacts

---

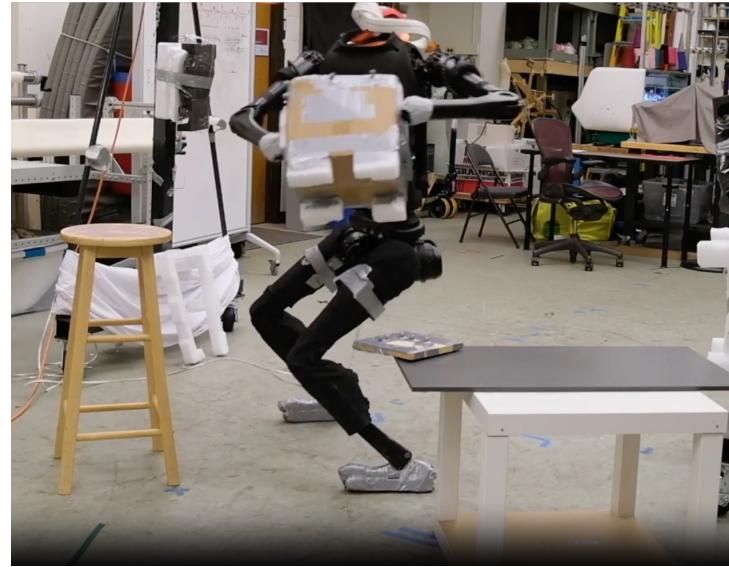
- **Goal:** learning versatile & agile & long-horizon whole-body skills *without* any motion priors
- **Key idea:** decompose a long-horizon skill into a sequence of contact goals and task goals



# What is Wrong with Sim2Real 2.0?

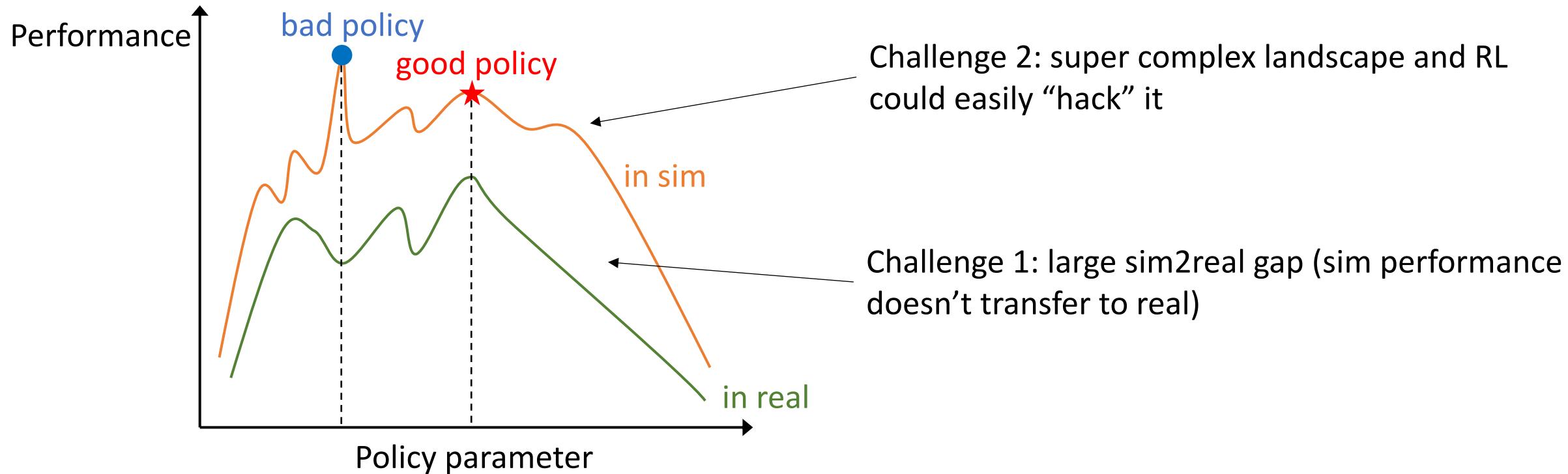


sim2real



- ❑ Sim2Real gap is large, unintuitive, and hard to quantify
- ❑ Tedious reward / curriculum / domain randomization tuning
- ❑ Hard to encode prior physics, poor sample complexity, unsafe
- ❑ No online reasoning: policies learned from sim are frozen in test time

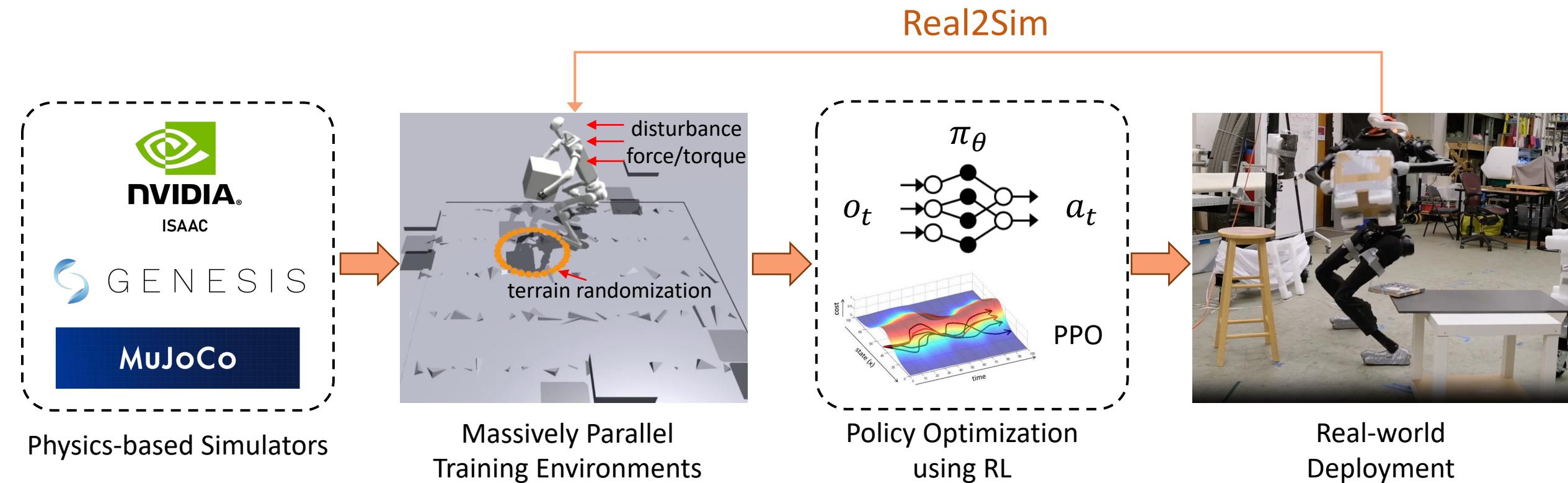
# From Sim2Real 2.0 to 3.0: Real2Sim and Structured RL



- Real2sim:** reduce the sim2real gap
- Structured RL:** add priors and inductive bias to have a smoother landscape

# Sim2Real 3.0: Real2Sim

- ❑ learning “residuals” to bridge the gap between real and sim



# Residual Dynamics Learning for Other Robotic Systems

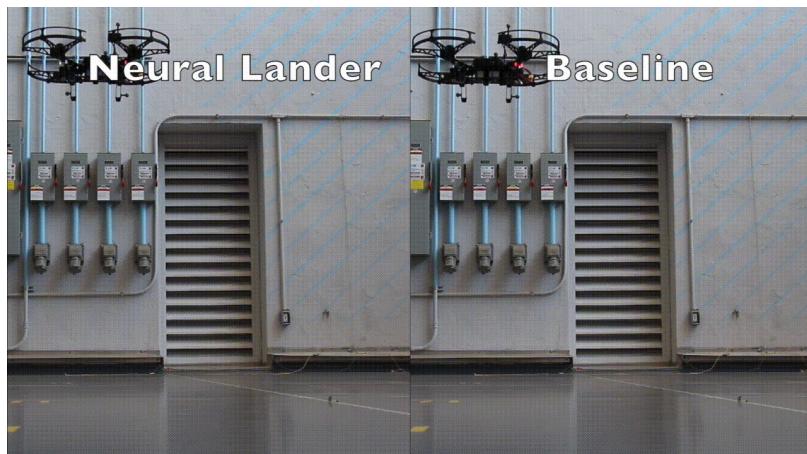
## □ Neural-Control Family

- Key idea: Collect data *in real* and use a DNN  $\hat{f}$  to approximate  $f$

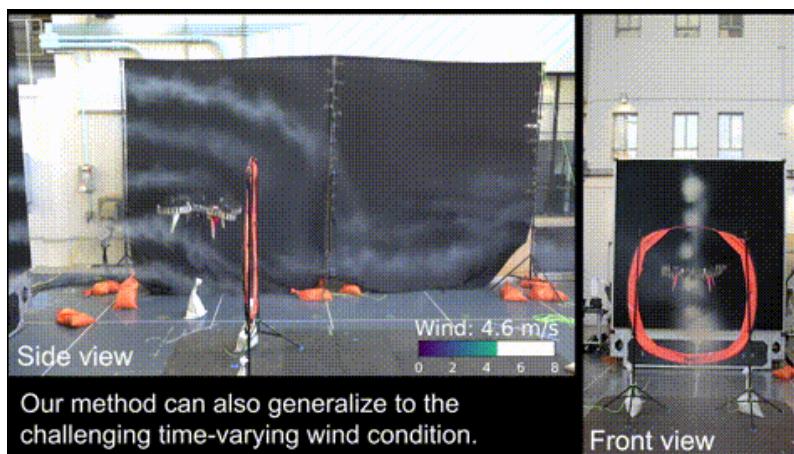
$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u + \hat{f}(q, \dot{q}, a, t)$$

→ unknown dynamics

- Then design a nonlinear controller  $u = \pi(q, \dot{q}, \hat{f})$
- Often need to regularize  $\hat{f}$  for Lyapunov stability guarantees



Neural-Lander  
[ICRA'19]



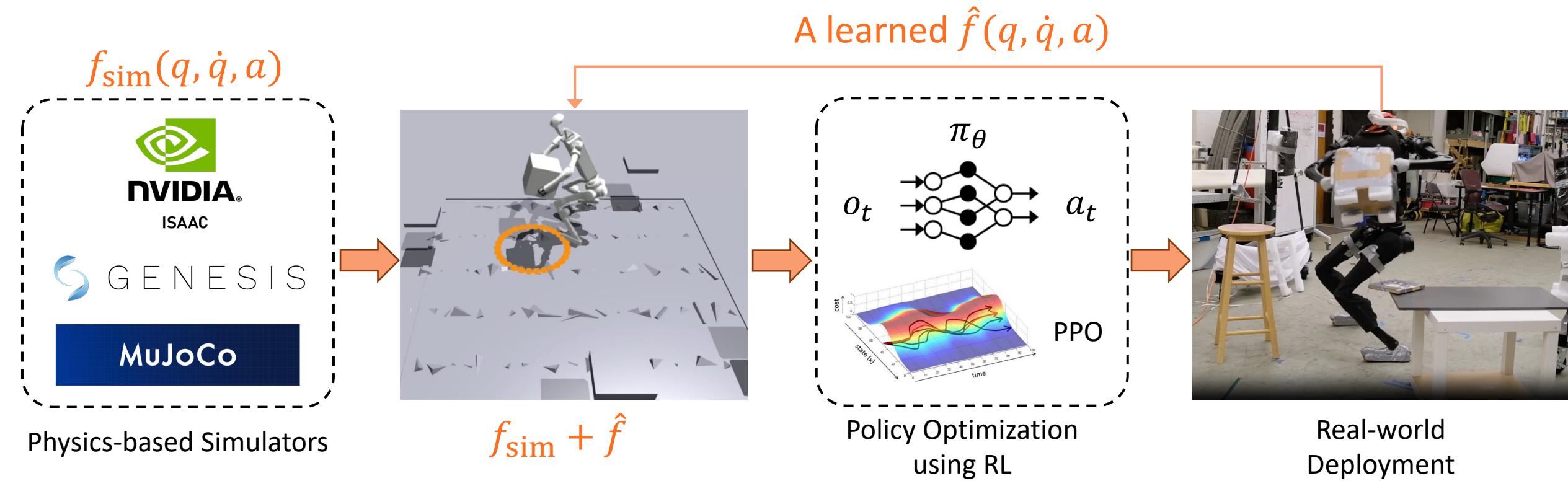
Neural-Fly:  $f$  is time-variant  
[NeurIPS'21][Science Robotics'22]



Aerial Manipulations  
[Guo\* and He\* et al., RAL'24]  
[He\* and Guo\* et al., RSS'25]

# Residual Dynamics Learning for Humanoids?

- Directly learning dynamics may not be a good idea for humanoids:
  - $\hat{f}$  needs to generalize well (requiring a lot of real-world data)
  - Need to regularize  $\hat{f}$  heavily to ensure  $f_{\text{sim}} + \hat{f}$  still “makes sense”
  - $\hat{f}$  will be exploited by RL

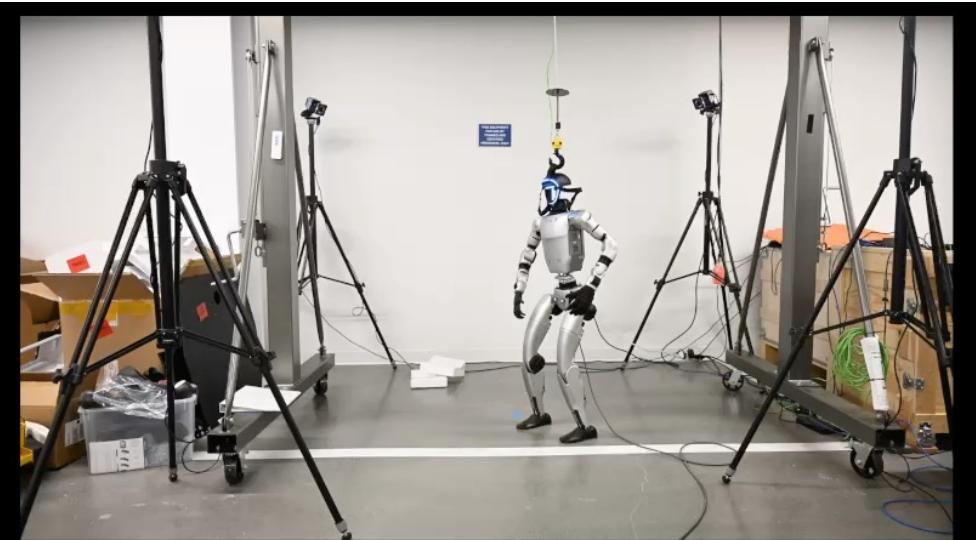


# An Alternative Solution: Learning a Delta Action Model

- The ASAP framework: learn a *delta action model* to match sim and real
  - Pretrain a policy  $\pi$  in sim, rollout in real:  $\{x_1^r, a_1^r, \dots, x_T^r\}$
  - Replay  $\{a_1^r, \dots\}$  in sim:  $x_{1:T}^s$ . Due to the sim2real gap,  $x_{1:T}^s \neq x_{1:T}^r$
  - Train a delta action model  $\Delta a(x, a, \dots)$  in sim such that  $a_t^r + \Delta a_t$  yields  $x_{1:T}^s \approx x_{1:T}^r$
  - Rollout  $\pi + \Delta a$  in sim to fine-tune  $\pi$ . Finally deploy  $\pi$  in real.



BeforeDeltaA



AfterDeltaA

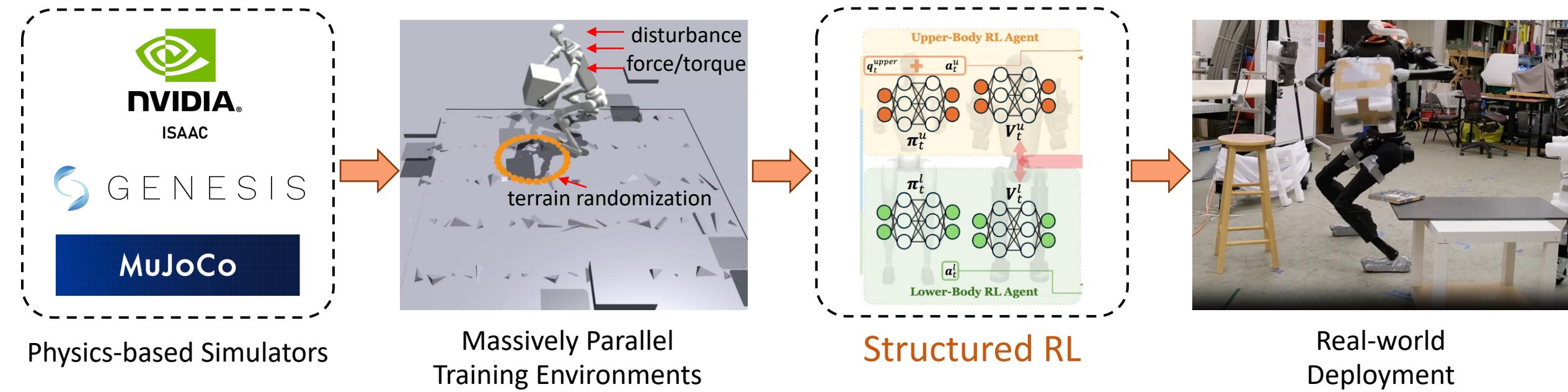
# Performance in Agile Whole-Body Control Tasks

- Similar to the human-to-humanoid pipeline but each policy focuses on one motion



# Sim2Real 3.0: Structured RL

- ❑ Leverage humanoid morphology to design better policy architecture
- ❑ Goal: make policy optimization easier



# FALCON: Dual-Agent RL for Force-Adaptive Loco-Manipulation

❑ Tasks: heavy-duty loco-manipulation



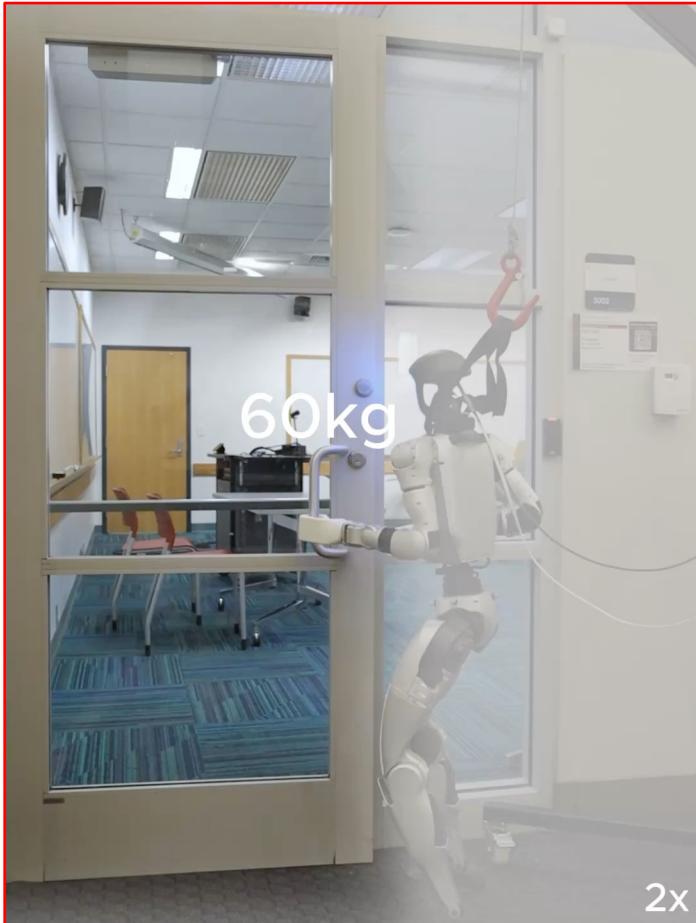
Baseline



FALCON

# FALCON: Dual-Agent RL for Force-Adaptive Loco-Manipulation

- ☐ Tasks: heavy-duty loco-manipulation



Baseline

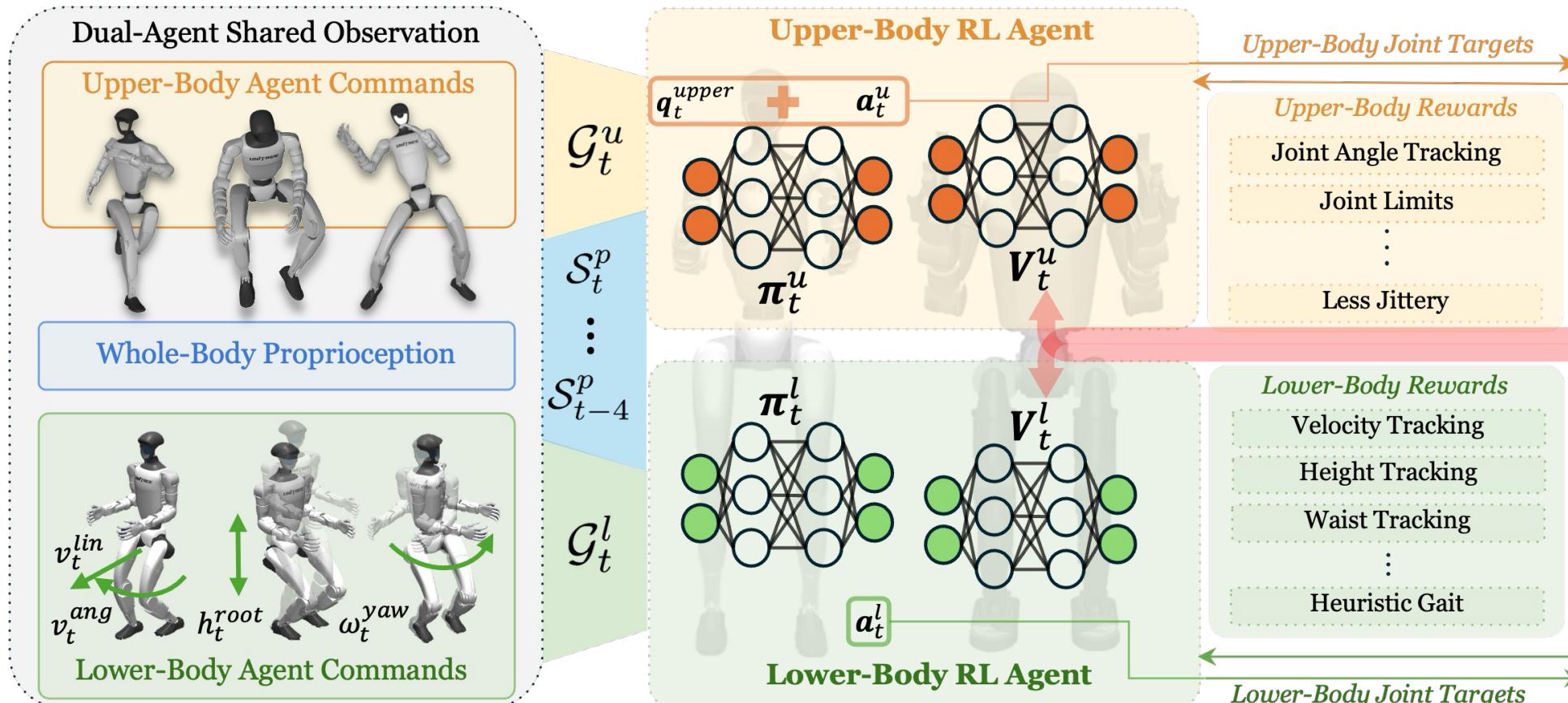


FALCON

# FALCON: Dual-Agent RL for Force-Adaptive Loco-Manipulation

## ❑ Key structure 1: dual-agent RL

- Two policies, two value functions (critics), two sets of rewards
- Jointly trained and both have whole-body proprioception input



# FALCON: Dual-Agent RL for Force-Adaptive Loco-Manipulation

## ❑ Key structure 2: adaptive and feasible 3D force curriculum on the end-effector

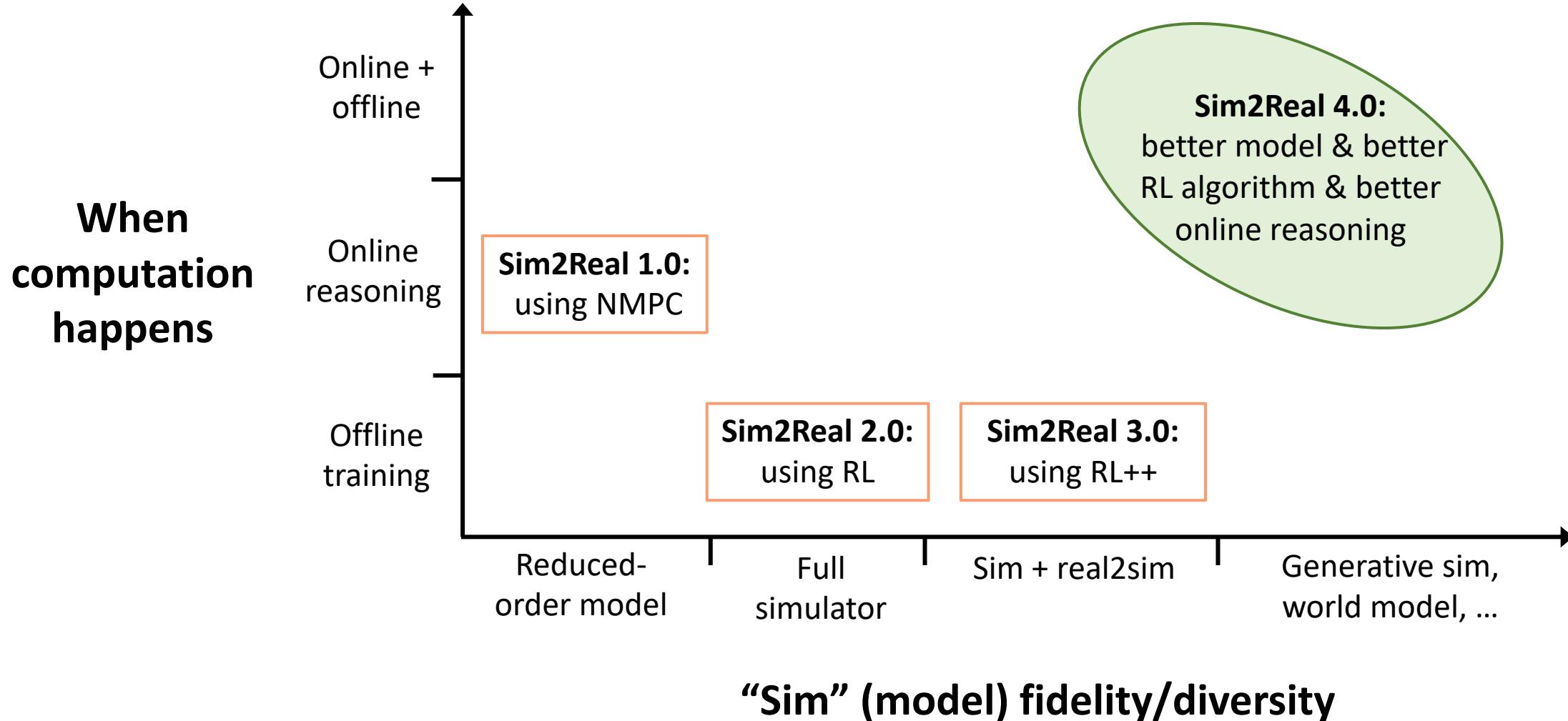
- Apply random external forces  $f^{ee}$  on two end-effectors
- Make sure  $f^{ee}$  is feasible with the motor torque limit



*Feasible 3D Force Curriculum*

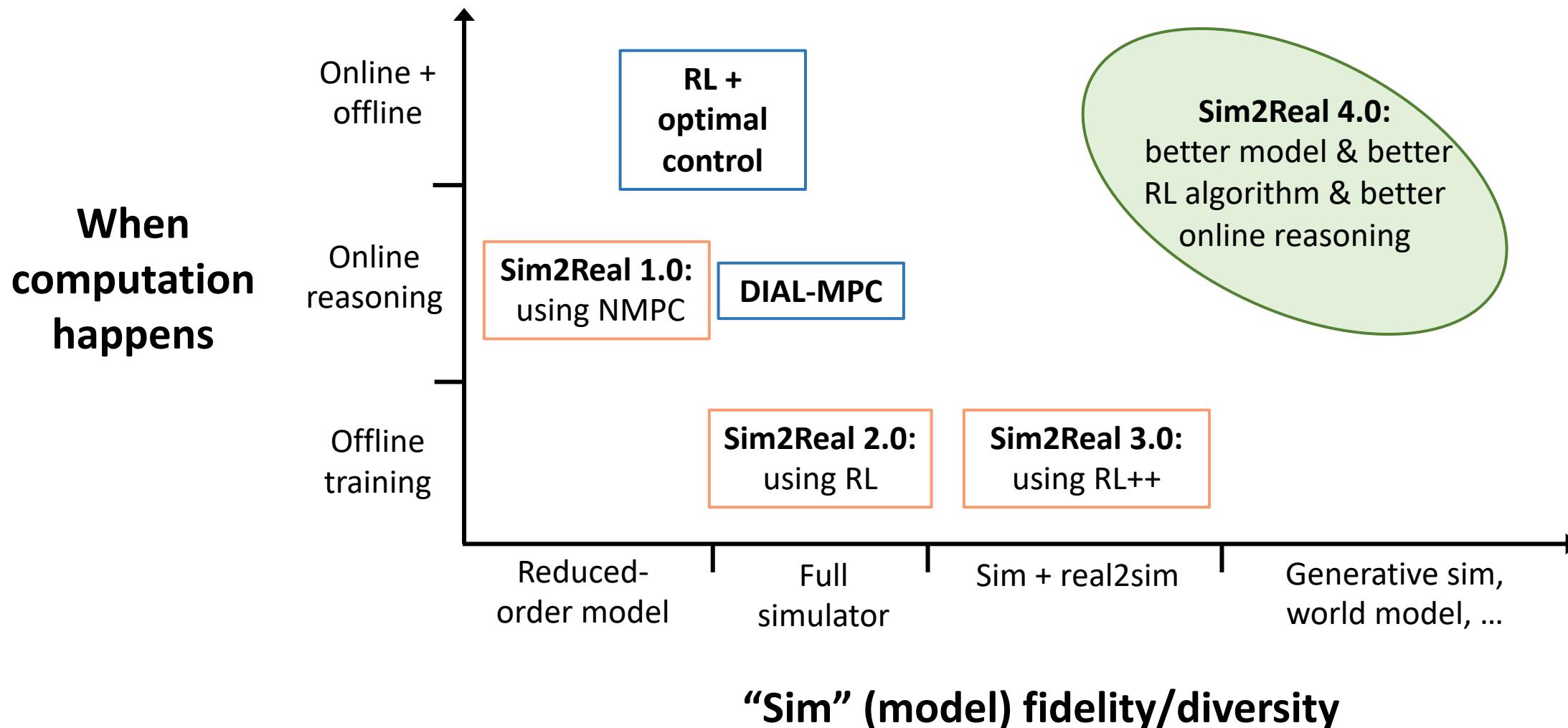
$$-\tau^{\lim} \leq \tau^g + J_{EE}^T f^{ee} \leq \tau^{\lim}$$
$$\tau^{\lim} \geq \mathbf{0}, \quad \tau^{\lim} - \tau^g \geq \mathbf{0}$$

# Zooming Out: Towards Sim2Real 4.0



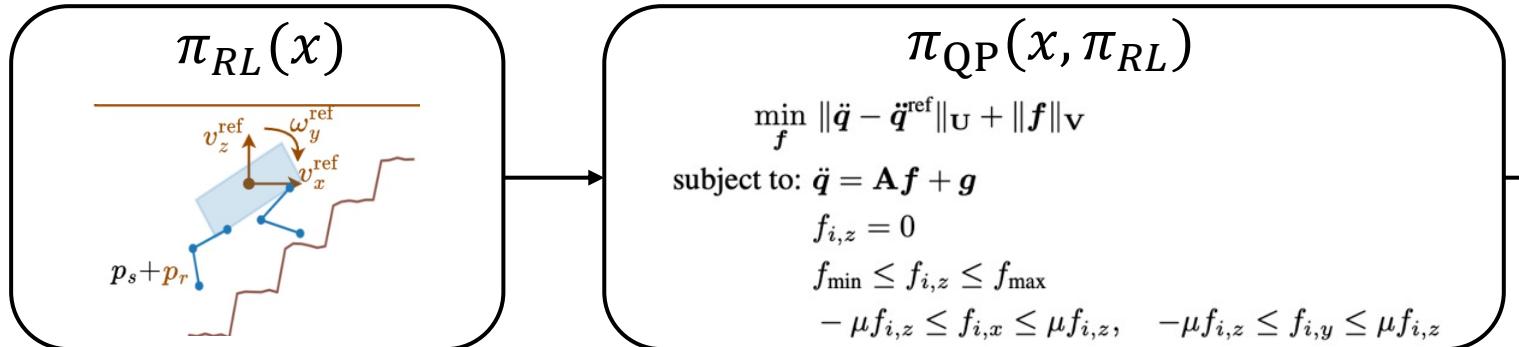
# Zooming Out: Towards Sim2Real 4.0

- ❑ Offline + online could be powerful!



# RL (full-order) + QP (Reduced-order)

- ❑  $\pi_{RL}$  outputs center of mass refs  $\ddot{q}^{\text{ref}}$ ;  $\pi_{QP}$  optimizes ground reaction force (GRF)
- ❑ Fully onboard & autonomous (depth camera for sensing)



[Agile Continuous Jumping in Discontinuous Terrains, Yang et al., ICRA'25]

[CAJun: Continuous Adaptive Jumping using a Learned Centroidal Controller, Yang et al., CoRL'23]

# Thank You!

Everything I presented is open-sourced:  
<https://lecar-lab.github.io/publications.html>