Мультимодальные и поведенческие модели

Sim-to-Real Transfer and Domain Adaptation

MIPT VLA 2026 | 2 ноября 2026

Алексей Староверов



МФТИ

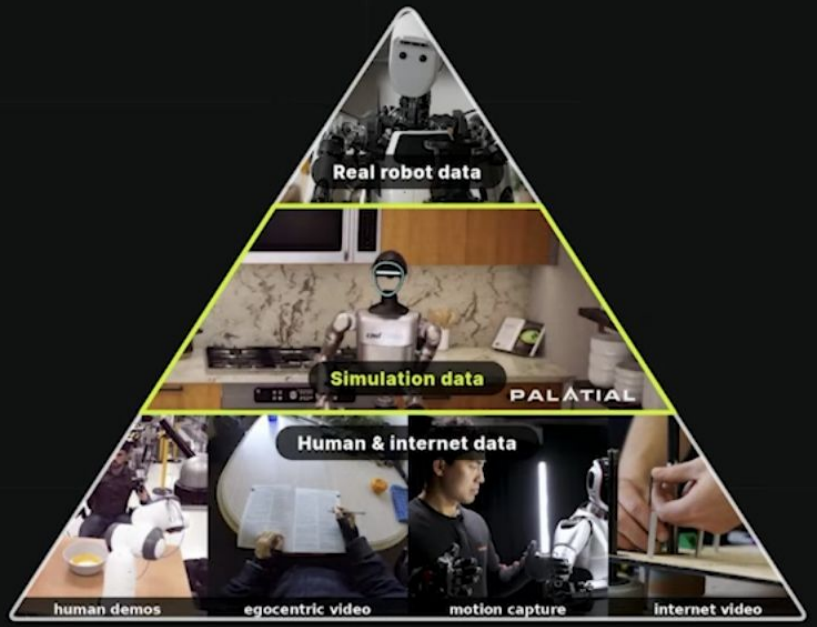
Center for Cognitive Modeling

План лекции

Generation of High Fidelity 3D assets for Sim2Real

Reinforcement Learning that can be transferred to a Real World

Physical AI has a Data problem



Real robot data scales 1:1 with hardware and operators. Highest fidelity, but it can't cover the edge cases.

Simulation data is the compute-backed way to scale, reproducible across tasks, robots, and environments. It closes the sim2real gap only when it's physically grounded.

Human and internet data is abundant and cheap, but low fidelity, and it carries the robot-embodiment gap.

Where simulation fits in robot development

Pre-training

Foundation policies learn from large, diverse data mixes.

Domain-randomized assets, at scale.



RL environments

Verifiable physics rewards: policies practice against ground truth.

Physics-tuned assets, codified motion and limits.



Deployment & validation

Generalization testing, and calibrated twins of the target system.

Physically grounded eval scenes.



Simulation carries every stage of the stack, **if the worlds it runs are physically trustworthy**.

Every “SimReady” generator today fails at least one of these

1 **Shape optimized for interaction**
Geometry built for articulation: parts open, slide, and turn without self-intersection, and fit together properly.

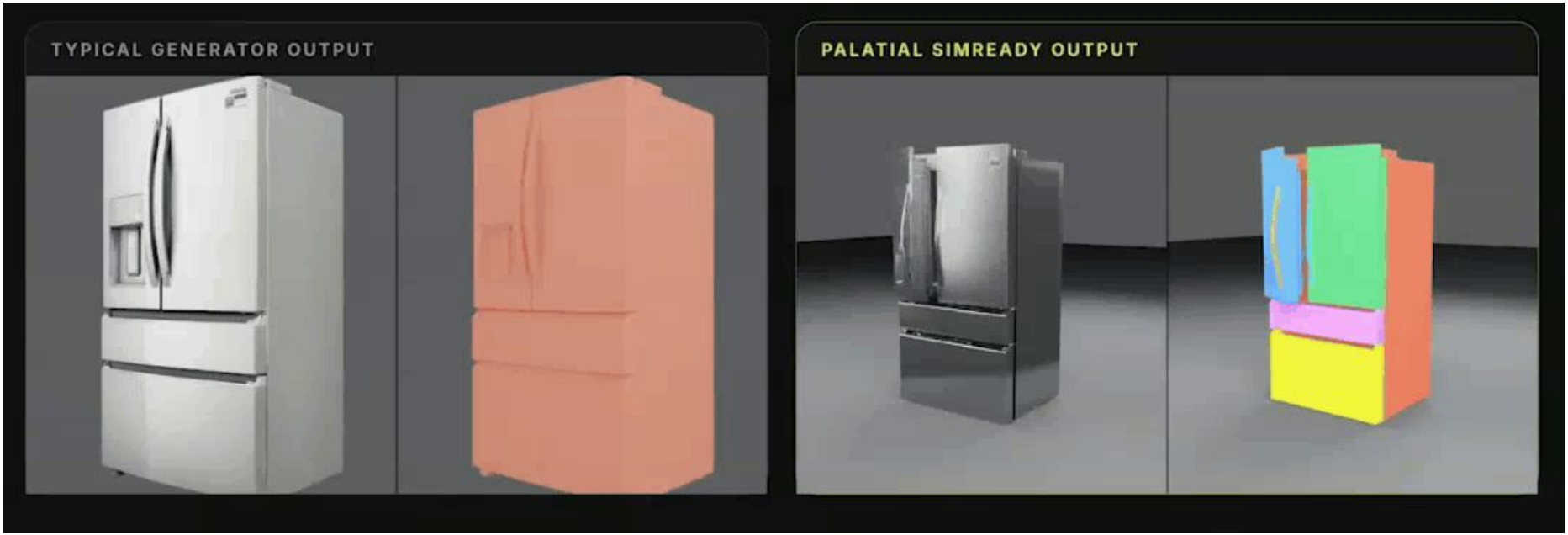
2 **Manipulation ready collisions**
Collision hulls accurate to the shape, without extra complexity that reduces performance.

3 **Physics optimized topology**
Solver-friendly topology: watertight, feature-aligned, and efficient, isotropic for soft bodies.

4 **Perception ready materials**
Bespoke textures and shaders: crisp text, correct handling of transparencies.

Getting all four right by hand takes a 3D modeler **days, per asset.**

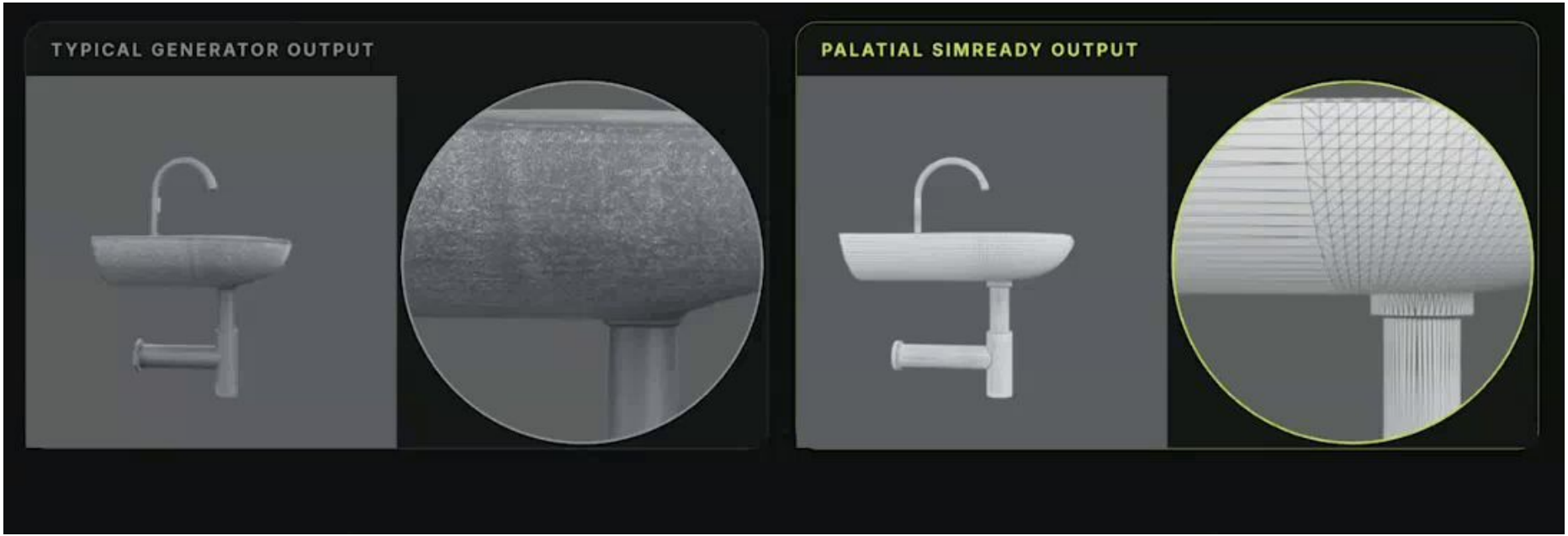
Shape optimized for interaction



Manipulation ready collisions



Physics optimized topology



Perception ready materials



The right asset for each scene



Assembly task

```
collisions: high-fidelity convex
joints: behavior scripts
physics: force-tuned
```



Layout planning

```
collisions: simplified proxies
physics: instanced, stable
variants: randomized
```

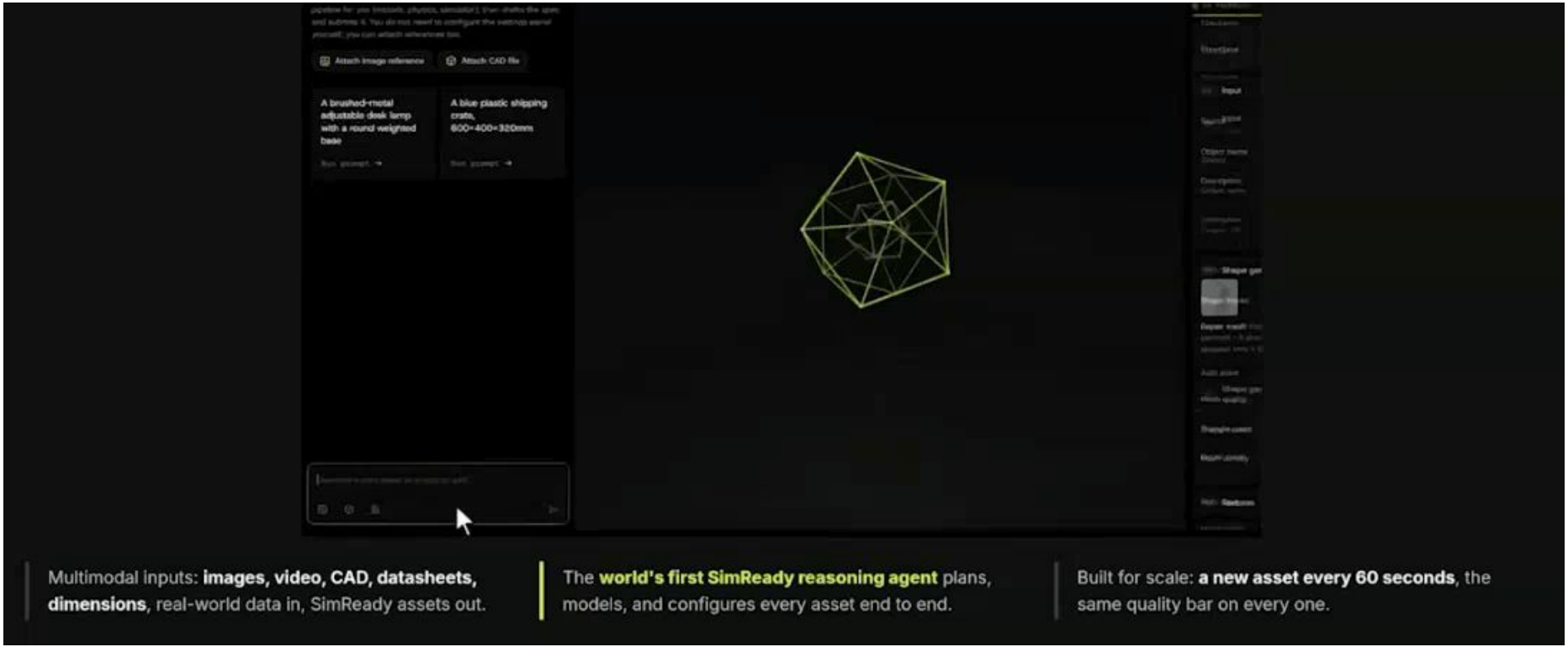


Deformable manipulation

```
collisions: signed distance field
physics: soft body (VDB)
topology: isotropic
```

Our agent **reasons about an asset's purpose**, and how it functions in the context of the scene, then delivers the right configuration through one API. No one-size-fits-all exports.

An agent-native platform that combines fidelity and scale

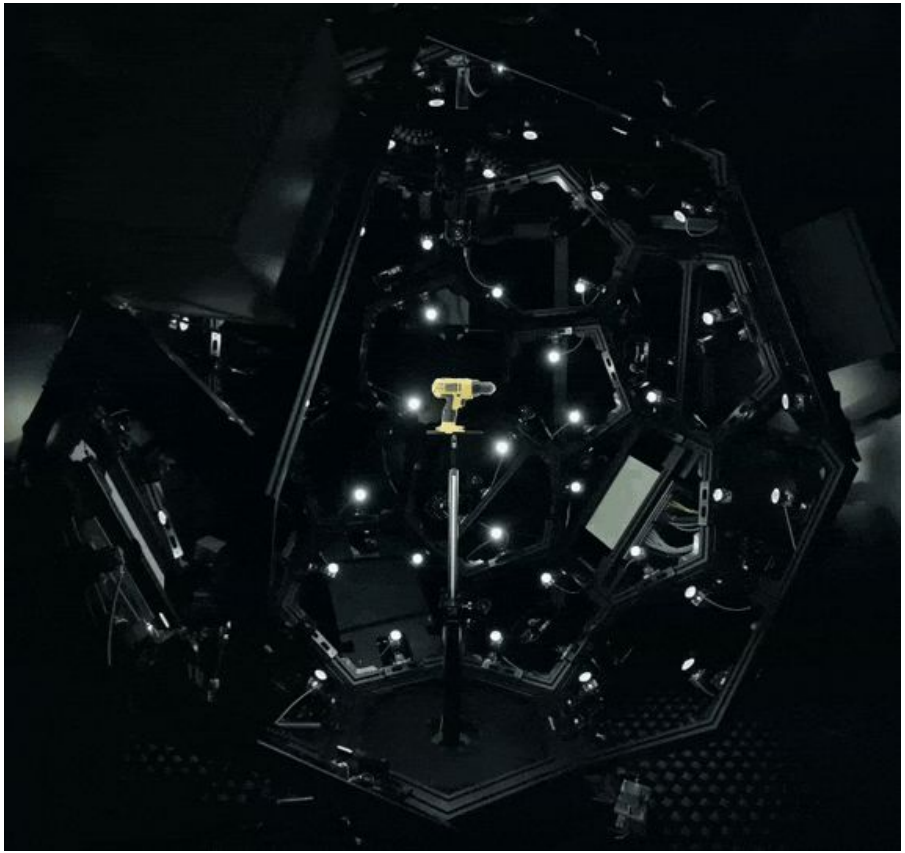


Multimodal inputs: **images, video, CAD, datasheets, dimensions**, real-world data in, SimReady assets out.

The **world's first SimReady reasoning agent** plans, models, and configures every asset end to end.

Built for scale: **a new asset every 60 seconds**, the same quality bar on every one.

Scanners that automatically produce physically grounded 3D Digital Twins



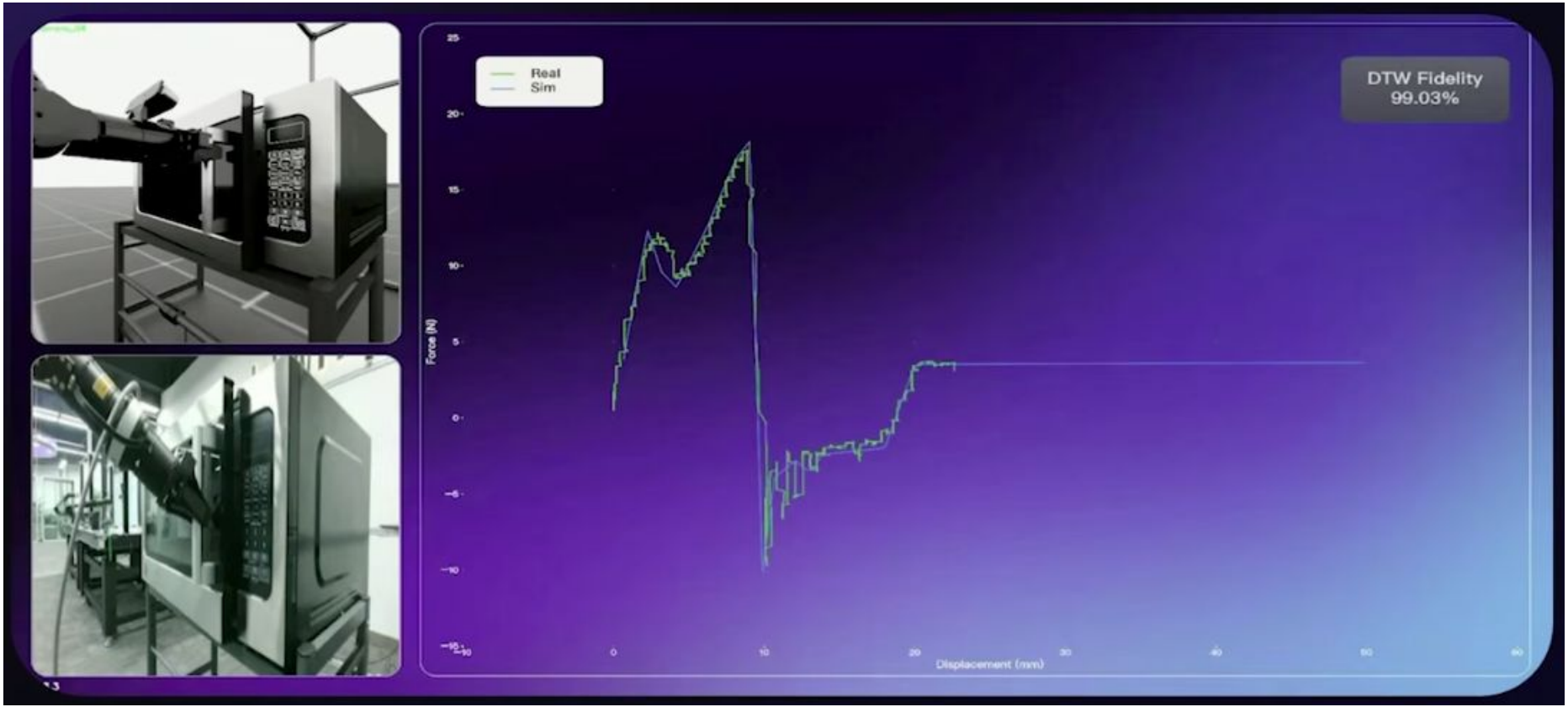
MEASURED

- ✓ GEOMETRY
- ✓ PBR MATERIALS
- ✓ MASS
- ✓ COLLIDERS

DERIVED & AGENTICALLY VALIDATED

- ≈ FRICTION COEFFICIENTS
- ≈ CENTER OF MASS & INERTIA TENSOR
- ≈ GRASP VECTORS
- ≈ RESTITUTION

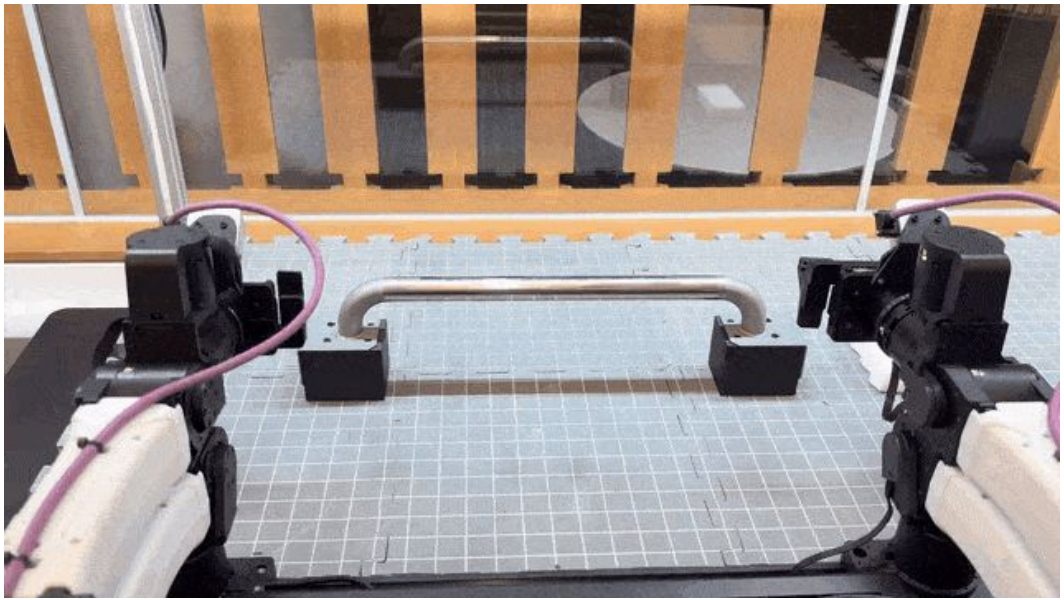
Physical Fidelity Benchmark



Sim-trained policies, early real-world results

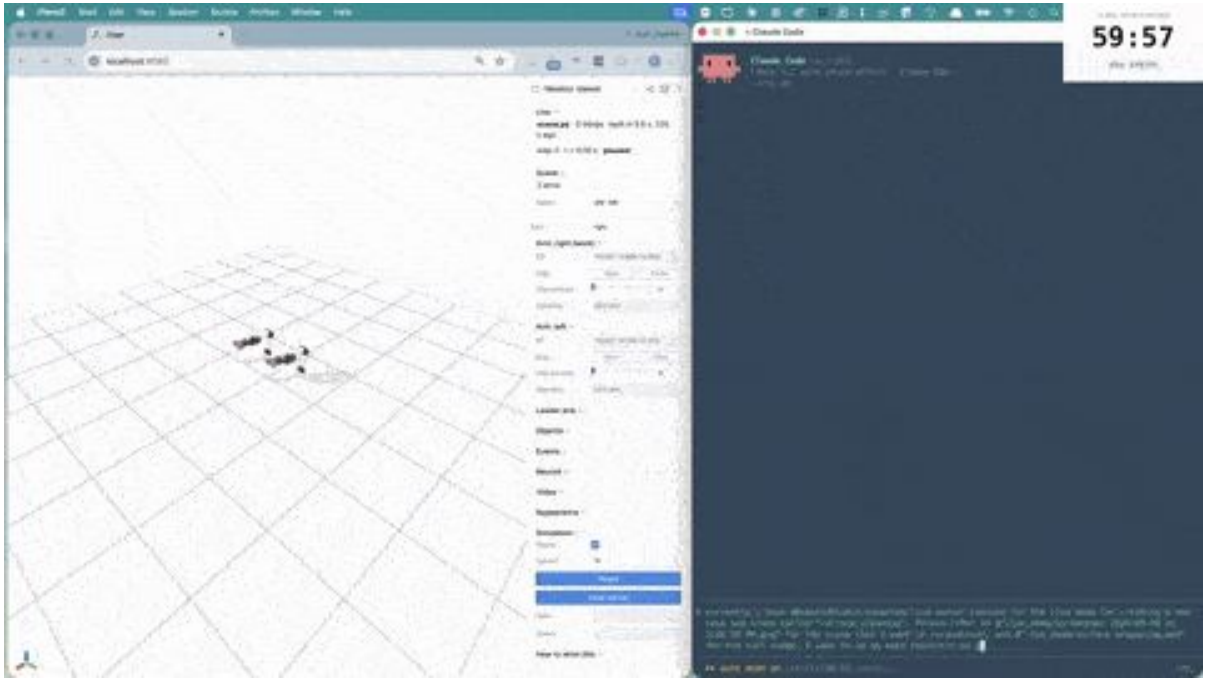


RoboSimStudio: An Interactive Toolkit for Rapid Real2Sim of Robot Manipulation Tasks



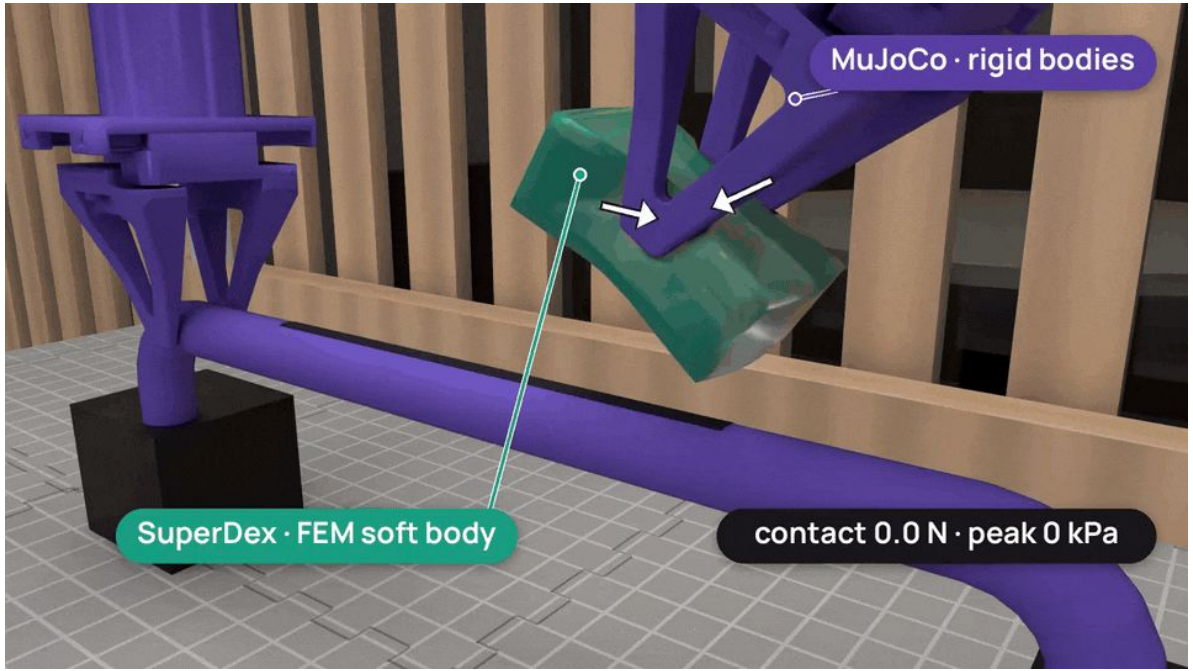
RoboSimStudio takes a smartphone video of a robot performing a task, here wiping a bar with a sponge, and a one-line prompt to a coding agent.

RoboSimStudio: An Interactive Toolkit for Rapid Real2Sim of Robot Manipulation Tasks



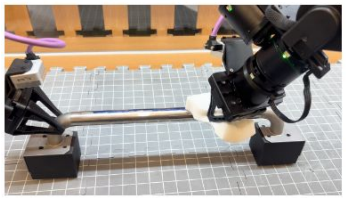
A Claude or Codex agent builds the simulation from the video: the scene, the camera and the physics, in about an hour (Real2Sim).

RoboSimStudio: An Interactive Toolkit for Rapid Real2Sim of Robot Manipulation Tasks

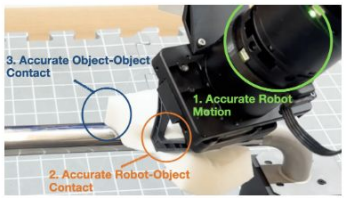


The output is a multi-physics simulation: rigid bodies in Newton coupled with the FEM sponge in SuperDex, which humans and agents inspect and tune until it matches the video.

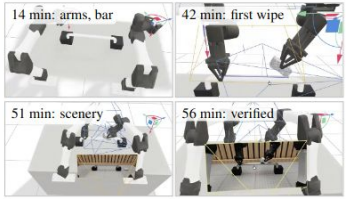
RoboSimStudio: An Interactive Toolkit for Rapid Real2Sim of Robot Manipulation Tasks



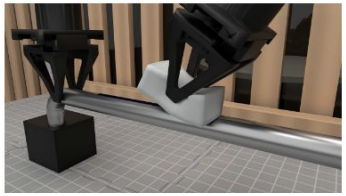
(a) Real-world input video.



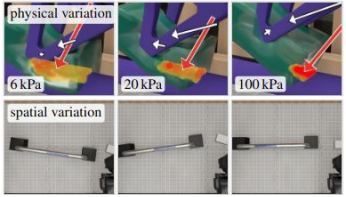
(b) Task challenges.



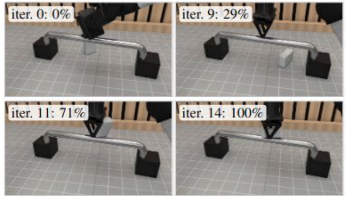
(c) Agentic Real2Sim construction.



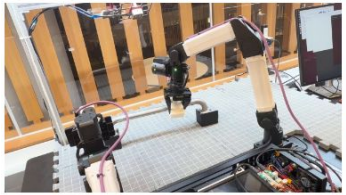
(d) Multi-physics simulation in RoboSimStudio.



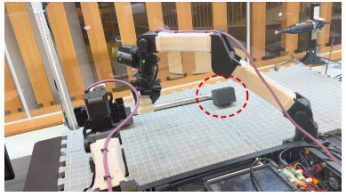
(e) Physical and spatial variations.



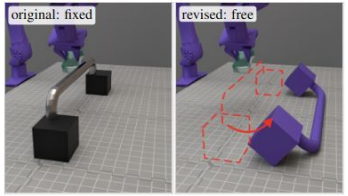
(f) Policy improvement with GaP.



(g) Sim2Real policy deployment.



(h) Real-world failure from unmodeled physics.



(i) Agentic simulation refinement.

What has reinforcement learning achieved?

- **AlphaGo** [1]: Learned to exceed what humans can do
- **Reasoning in LLMs** [2]: Coherent over long horizon
- **Dexterity in robotics** [rest of the talk]: Reliability

Zooming in on robotics

What enabled these success ?

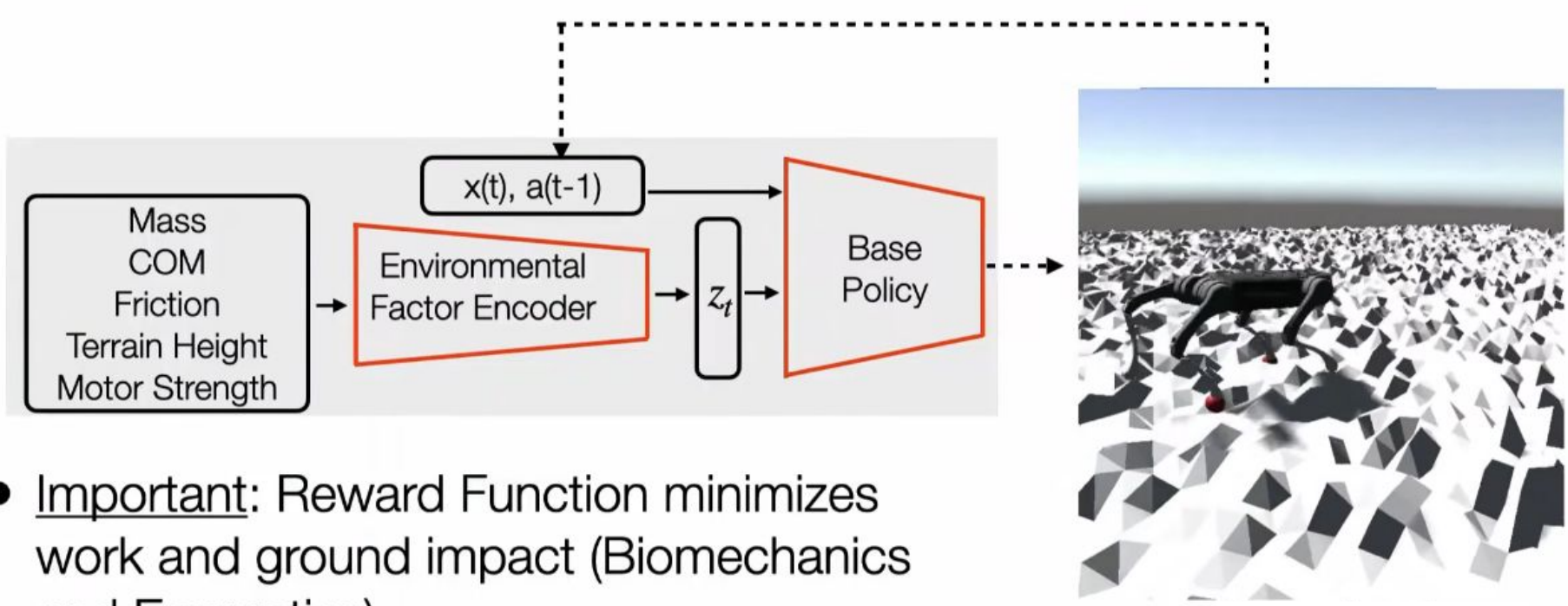
- Well specified rewards —> Programmatically calculate them
- Scalable evaluation of policies —> Simulation

Simulation to real?

Zooming in on robotics



Learn to Walk in Simulation



- Important: Reward Function minimizes work and ground impact (Biomechanics and Energetics)

Reward Function

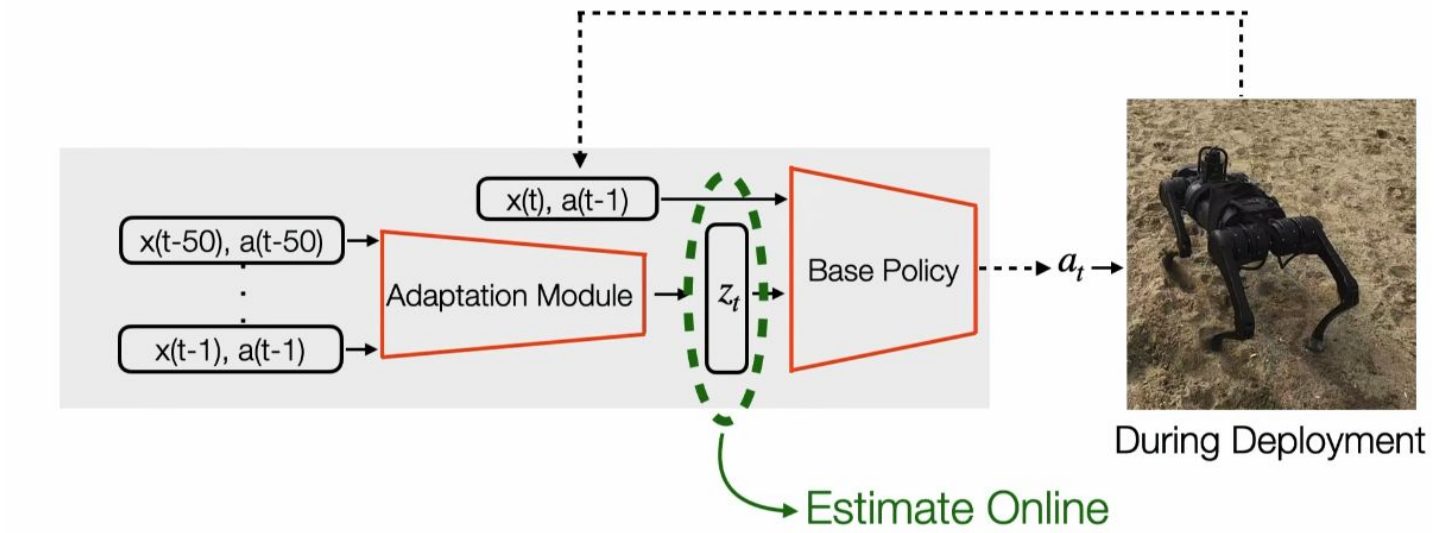
1) Forward: $\min(v_x^t, 0.35)$
2) Lateral Movement and Rotation: $-\|v_y^t\|^2 - \|\omega_{\text{yaw}}^t\|^2$
3) Work: $-|\boldsymbol{\tau}^T \cdot (\mathbf{q}^t - \mathbf{q}^{t-1})|$
4) Ground Impact: $-\|\mathbf{f}^t - \mathbf{f}^{t-1}\|^2$
5) Smoothness: $-\|\boldsymbol{\tau}^t - \boldsymbol{\tau}^{t-1}\|^2$
6) Action Magnitude: $-\|\mathbf{a}^t\|^2$
7) Joint Speed: $-\|\dot{\mathbf{q}}^t\|^2$
8) Orientation: $-\|\boldsymbol{\theta}^t_{\texttt{roll, pitch}}\|^2$
9) Z Acceleration: $-\|v_z^t\|^2$
10) Foot Slip: $-\|\text{diag}(\mathbf{g}^t) \cdot \mathbf{v_f}^t\|^2$

Forward Walking

Energetics

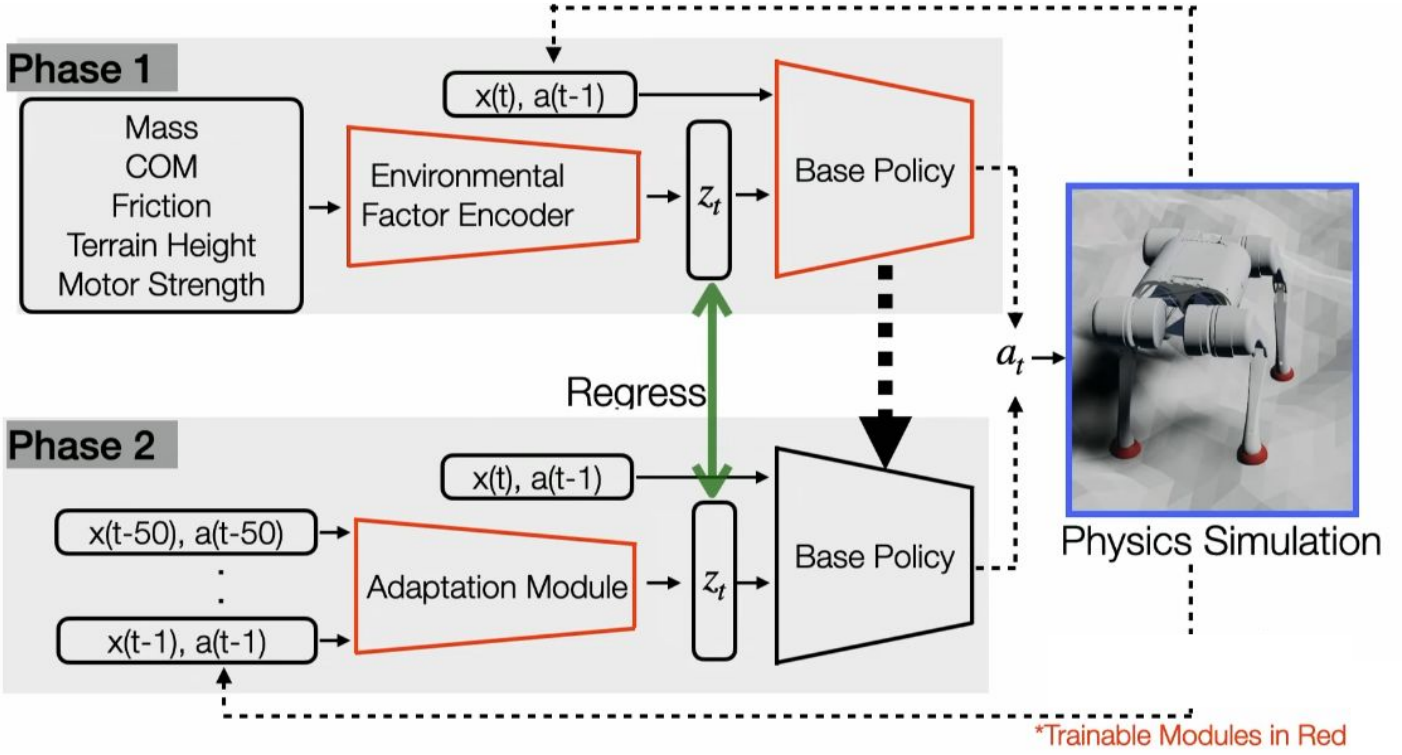
Stability + Minimize hardware damage

Key Insight - Extrinsics from Observation History

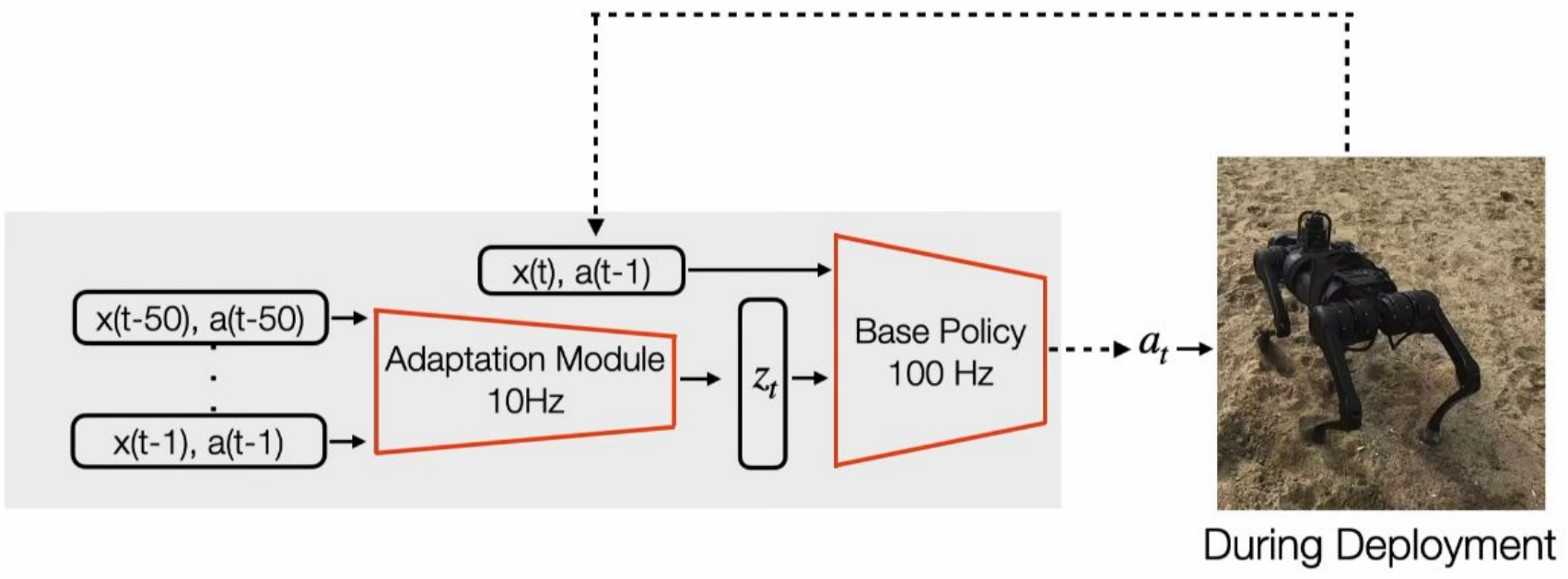


- Discrepancy b/w expected movement and actual measured movement

Training Summary



Test Time


x(t), a(t-1)
x(t-50), a(t-50)
x(t-1), a(t-1)
Adaptation Module
10Hz
z_t
Base Policy
100 Hz
a_t
During Deployment

Existing Attempts at Learning General Policies

	Success (%)	TTF	Reward	Distance (m)	Samples	Torque	Jerk	Ground Impact
Robust [49, 38]	62.4	0.80	4.62	1.13	0	527.59	122.50	4.20
SysID [54]	56.5	0.74	4.82	1.17	0	565.85	149.75	4.03
AWR [39]	41.7	0.65	4.17	0.95	40k	599.71	162.60	4.02
RMA w/o Adapt	52.1	0.75	4.72	1.15	0	524.18	106.25	4.55
RMA	73.5	0.85	5.22	1.34	0	500.00	92.85	4.27
Expert	76.2	0.86	5.23	1.35	0	485.07	85.56	3.90

- Domain Randomization (Robust)
- System Identification
- Fine tuning at test time in the real world

https://arxiv.org/abs/2107.04034

Legged Locomotion in Challenging Terrains using Egocentric Vision

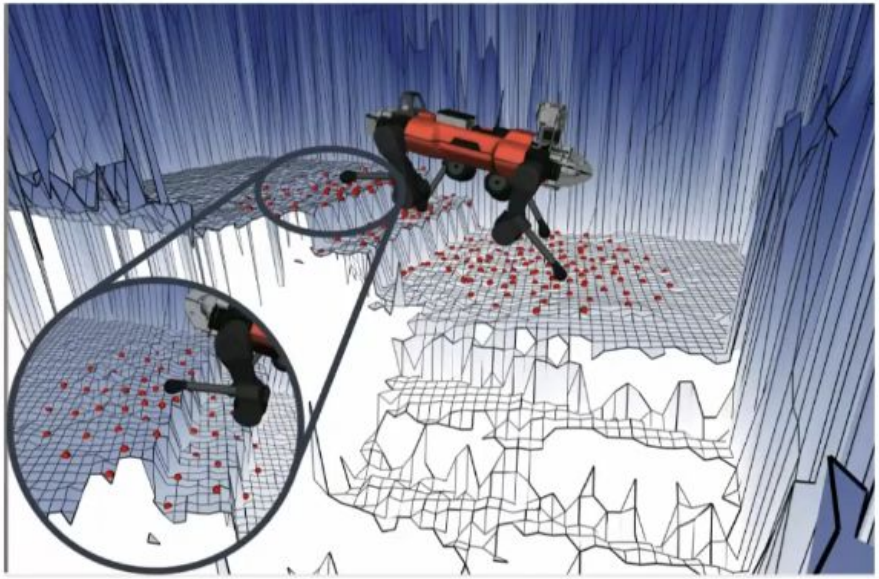
Ananye Agarwal*
CMU

Ashish Kumar*
UC Berkeley

Jitendra Malik†
UC Berkeley

Deepak Pathak†
CMU

Typical approach: build terrain maps from vision



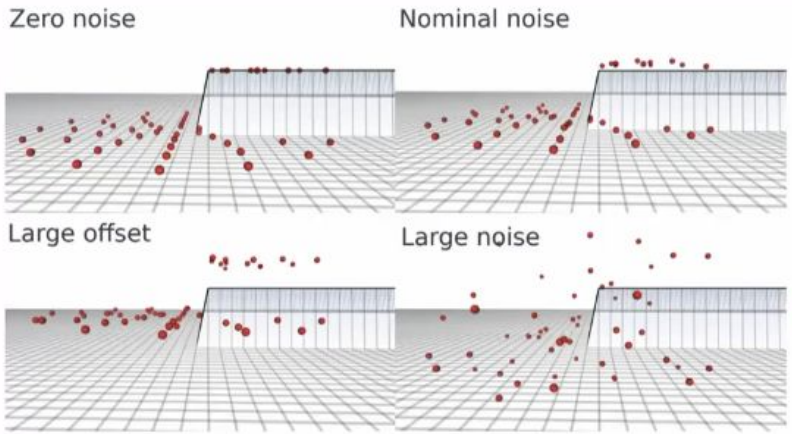


Miki, Takahiro, et al. "Learning robust perceptive locomotion for quadrupedal robots in the wild." *Science Robotics* 7.62 (2022)

Kim, Donghyun, et al. "Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot." ICRA 2020.

But terrain maps are very noisy!

Noisy maps wipe out signal ⇒ degraded performance



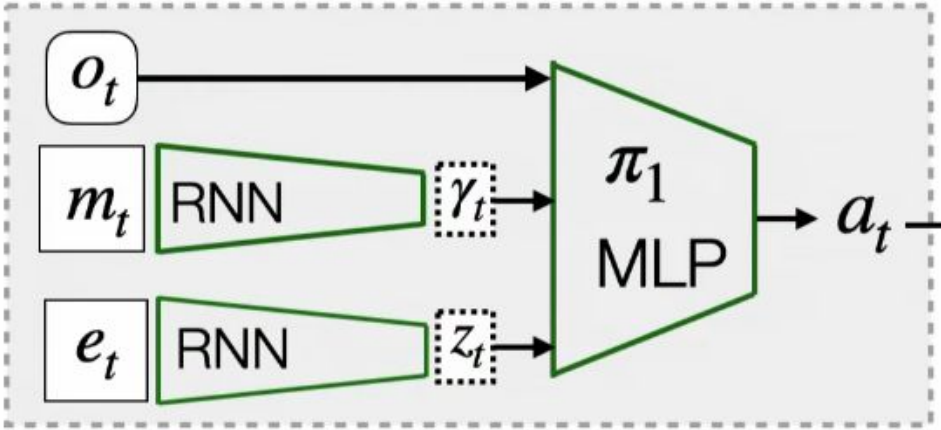
Miki, Takahiro, et al. "Learning robust perceptive locomotion for quadrupedal robots in the wild." Science Robotics 7.62 (2022)

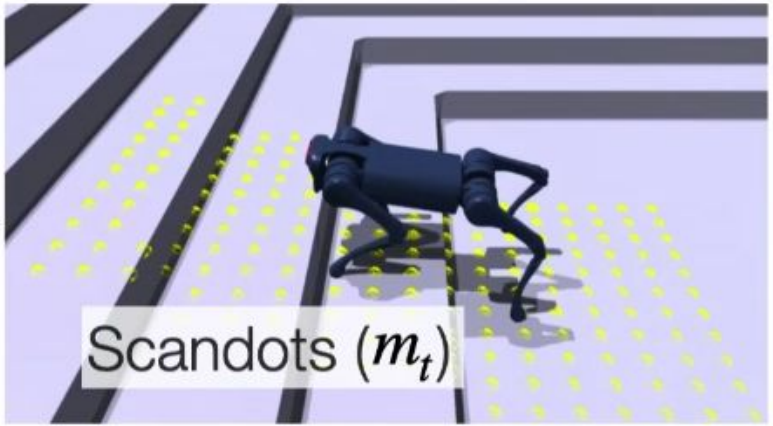
?




Stepping Stones (~15 cm apart)
Stanford

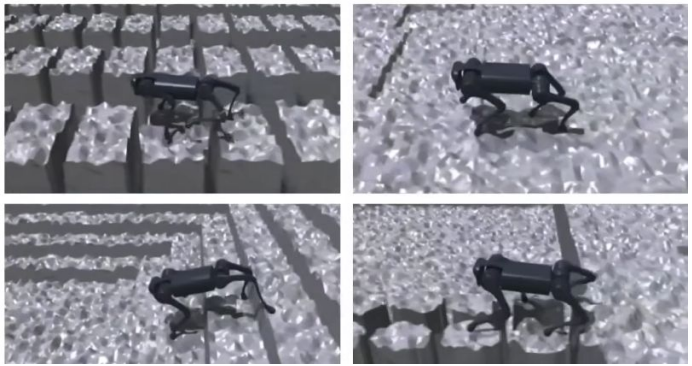
Phase 1: Learning to Walk with Privileged Terrain Information



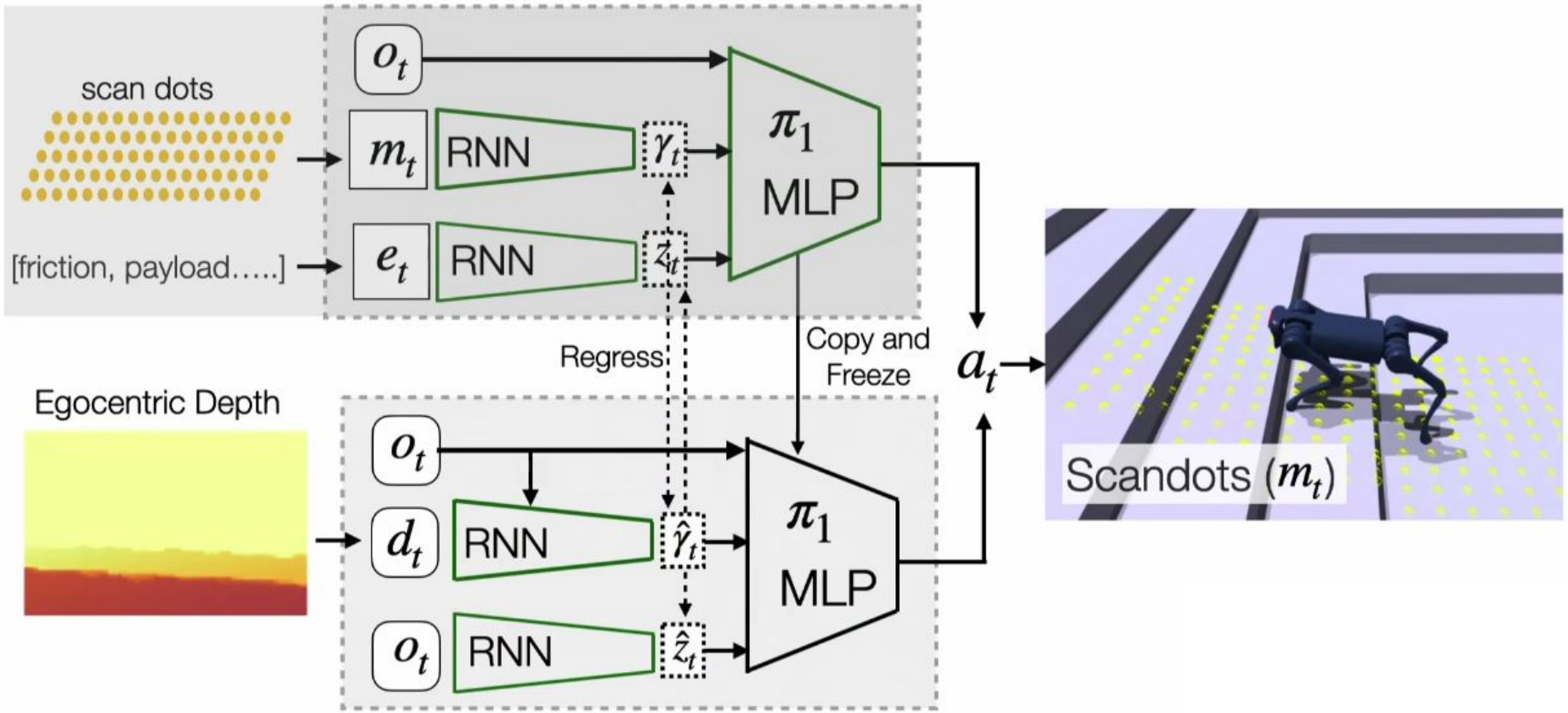


PPO in IsaacGym

Reward = (Track Velocity) + (Minimize Energy)



Phase 2: Learning to Walk with Egocentric Depth



Performance is better without maps

Average Distance

	Blind	Map Based	Ours
Slopes	34.72	36.14	43.98
Stepping Stones	1.02	1.09	18.83
Stairs	16.64	6.74	31.24
Discrete Obstacles	32.41	29.08	40.13