上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Lecture 39: Sim-to-Real Challenges

Tao Huang

John Hopcroft Center, School of Computer Science, Shanghai Jiao Tong University

https://taohuang.info/cs3317

https://oc.sjtu.edu.cn/courses/89538

AI tools assisted in generating some figures in these slides. All such content has been reviewed, and the instructor is responsible for its accuracy.

Where We Are

- **L38 built the brain** — a vision-language-action policy that maps pixels straight to motor commands.
- But that brain learns from data — and **real robot data is the scarcest resource in AI.**
- ***L38 gave the robot a policy. L39 asks where the experience to train it comes from.***

Why Doesn’t Sim-Perfect Survive Reality?

A robot that walks flawlessly in simulation face-plants the moment it touches real ground. Why?

Years of experience, compressed into hours — but it's all happening inside a computer.

Objectives

After this lecture, you will be able to:

- Explain why simulation is the default source of robot training data.
- Describe the reality gap and its main physical sources.
- Distinguish the three sim-to-real strategy families.
- Evaluate which strategy fits a given task.
- Analyze a sim-to-real failure and name the likely culprit.

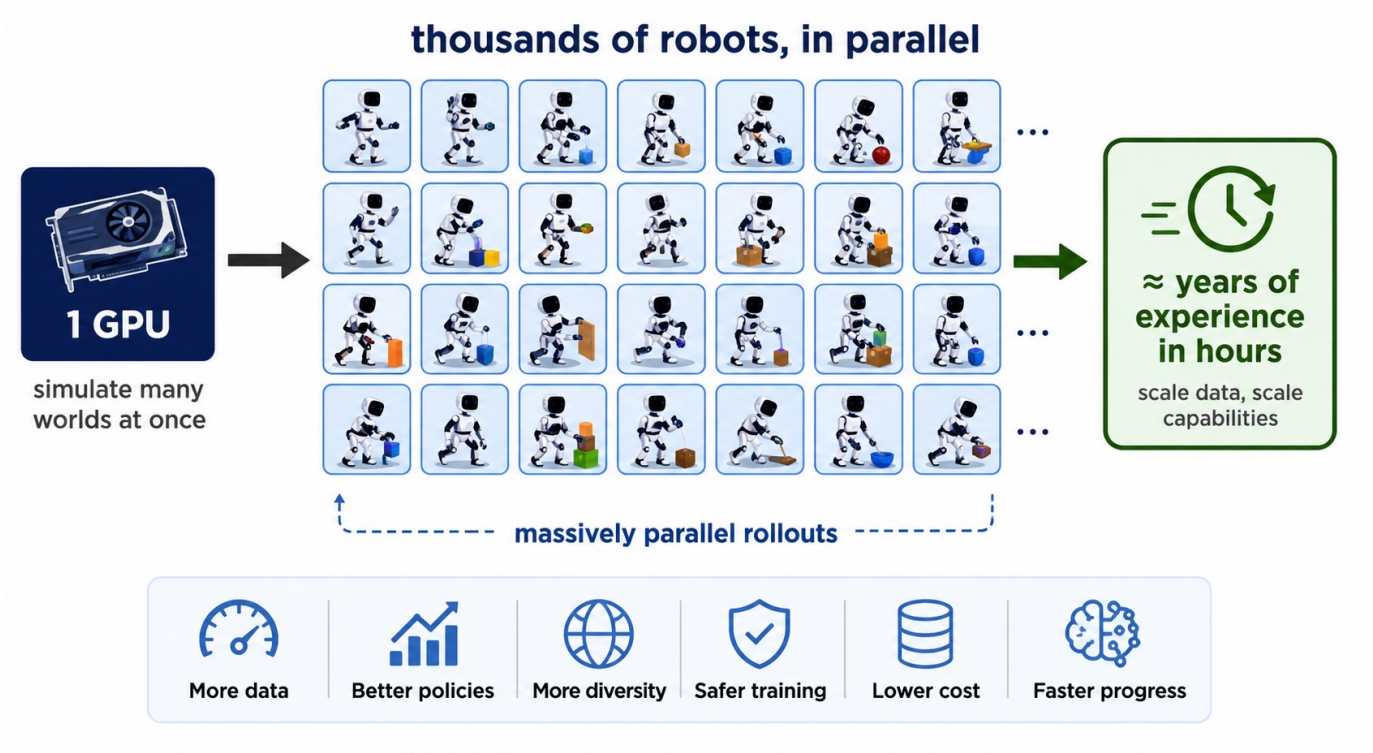
1. Why Simulate at All?

The Data Problem

- **Situation:** Deep RL and imitation learning need millions–billions of trial-and-error interactions.
- **Complication:** On a real robot, each interaction is slow, wears out hardware, can't be parallelized — and a fall can break a $100k machine.
- **Question:** *Where do we get that much experience cheaply and safely?*

Simulation is The Cheat Code

- Simulation is **fast** (faster than real-time), **free** (no hardware), **safe** (falls cost nothing), **perfectly labeled** (ground-truth state for free), and **massively parallel** (thousands of robots on one GPU).



Answer: don't collect real data — manufacture it in physics simulation.

The Catch

- Every simulator is a **model** of physics — not physics.
- Friction, contact, latency, motor dynamics, sensor noise: all approximate. A policy can exploit a simulator's quirks and learn behaviors that only work *there*.

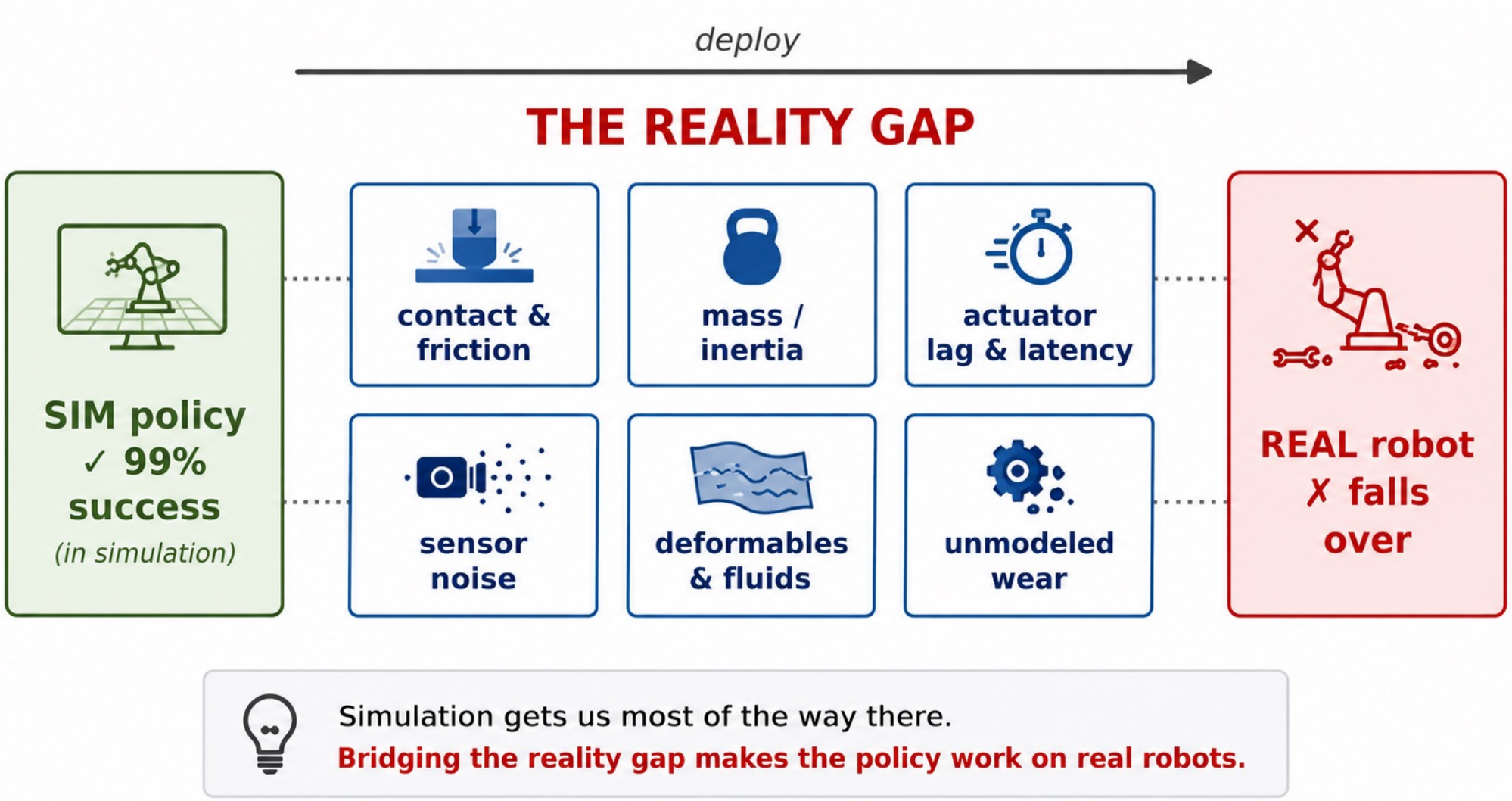
The cheat code has a bug: sim ≠ reality.

2. The Reality Gap

It Works in Sim, Fails in Real

- **Situation:** We trained a great policy in sim — 99% success.
- **Complication:** Deploy it on the real robot → it stumbles, oscillates, or freezes.
- **Question:** *What exactly is different between the two worlds?*

Anatomy of The Gap



Simulation gets us most of the way there.

Bridging the reality gap makes the policy work on real robots.

Worse than Ordinary Overfitting

- In supervised learning, test data is drawn from the **same distribution** as training.

- Here the test distribution (reality) is **systematically shifted** from the training distribution (sim). The policy didn't just under-generalize — it learned to rely on details that don't exist.

Sim-to-real is distribution shift you create on purpose — and then have to undo.

The Driving Question

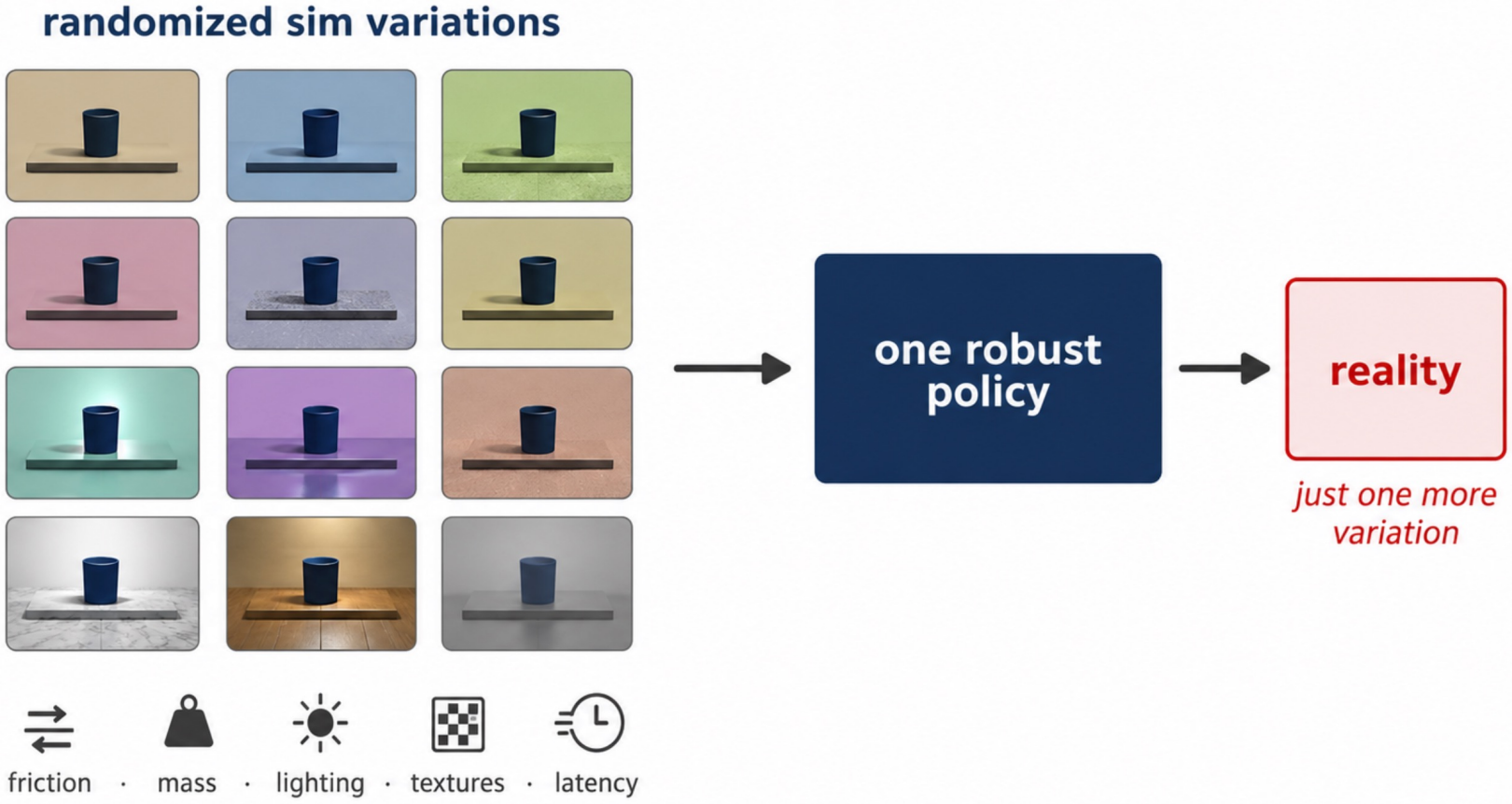
How do we make a sim-trained policy survive reality?

3. Strategy 1 — Domain Randomization

The Counter-Intuitive Idea

- **Situation:** We can't make the sim match reality exactly.
- **Complication:** Any single best-guess sim is wrong in some way we can't predict.
- **Question:** *If we can't hit reality, what should we aim at instead?*
- **Answer:** Don't aim — spray. Randomize sim parameters so wildly that reality looks like just one more variation the policy already handles.

How Domain Randomization Works



Randomize **physics** (friction, mass, damping) and **appearance** (textures, lighting, camera) → the policy must be robust to all of it.

The Dominant Approach

- Across 250+ surveyed sim-to-real papers, domain randomization is the **single most common technique**.

- It's simple, needs **no real data**, and gives **zero-shot** transfer when it works.

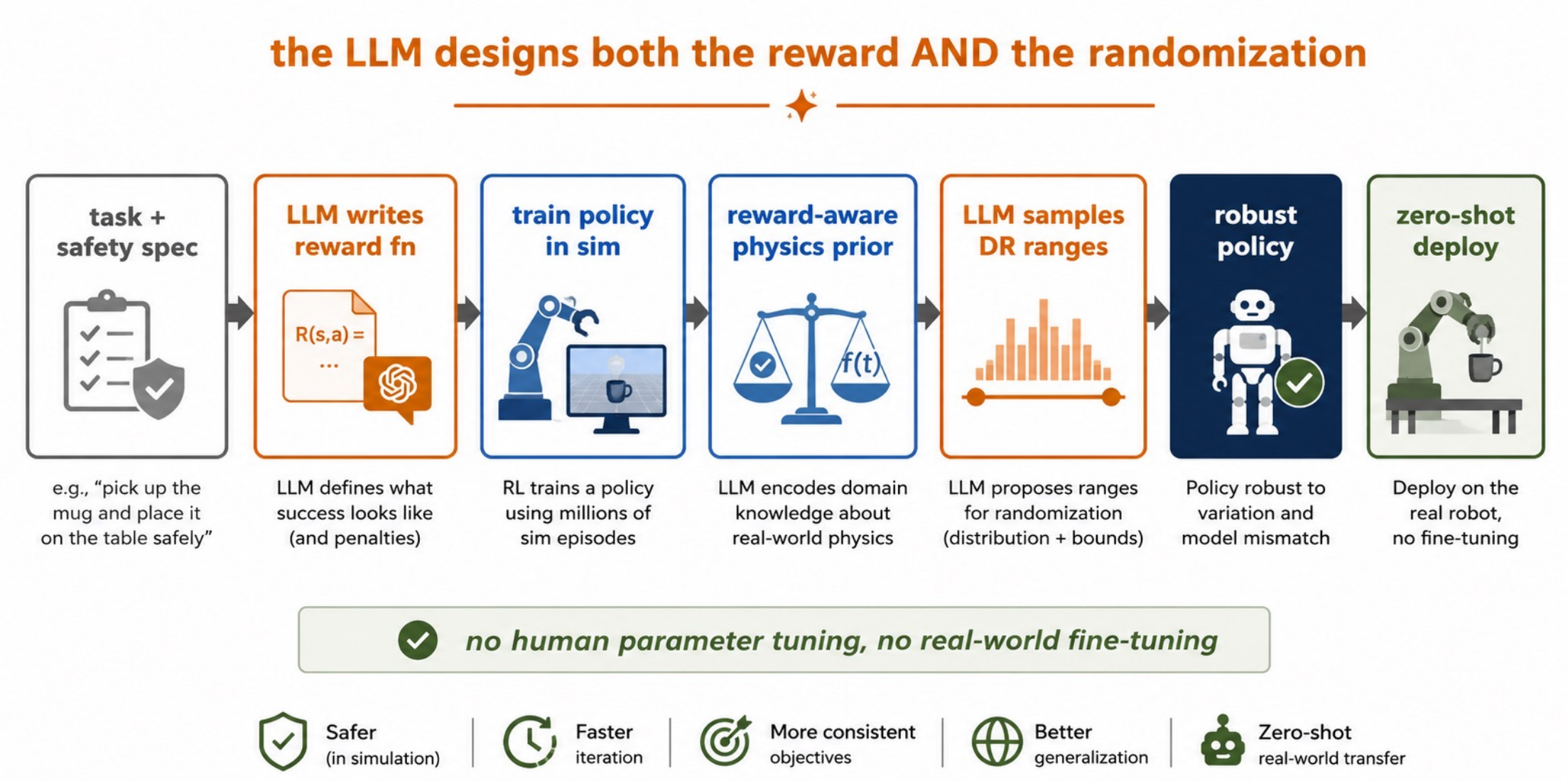
Most legged robots you've seen walk were trained this way.

Let an LLM Design It

- **Situation:** Choosing which parameters to randomize, and over what ranges, is expert hand-tuning.
- **Complication:** A roboticist can stare at dozens of parameters for weeks.
- **Answer:** LLMs already "know" physics common sense (friction matters on slippery floors). Let the LLM write the reward and propose the randomization ranges.

→ **DrEureka.**

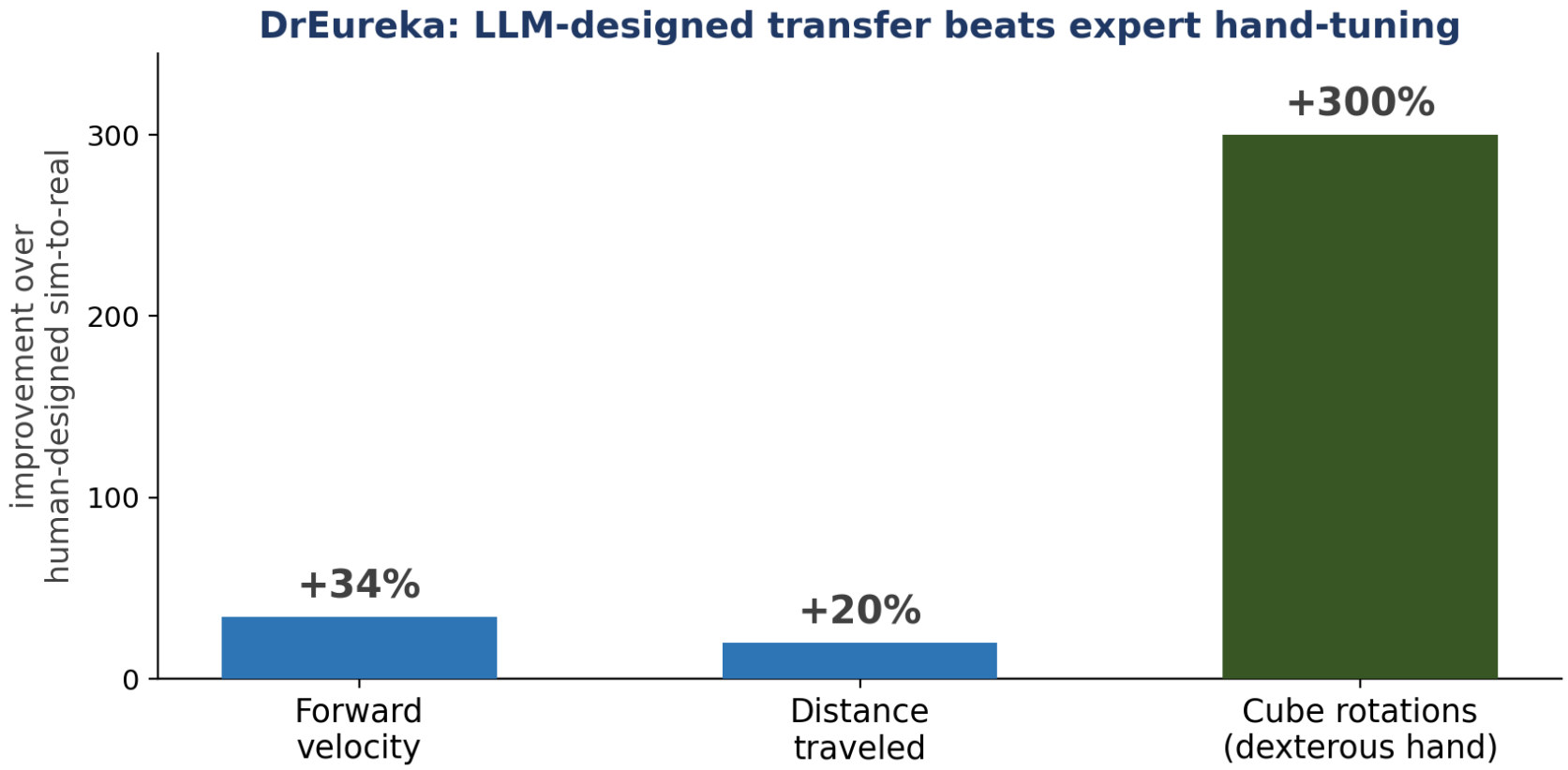
Worked Example: DrEureka



DrEureka: Zero-Shot Result

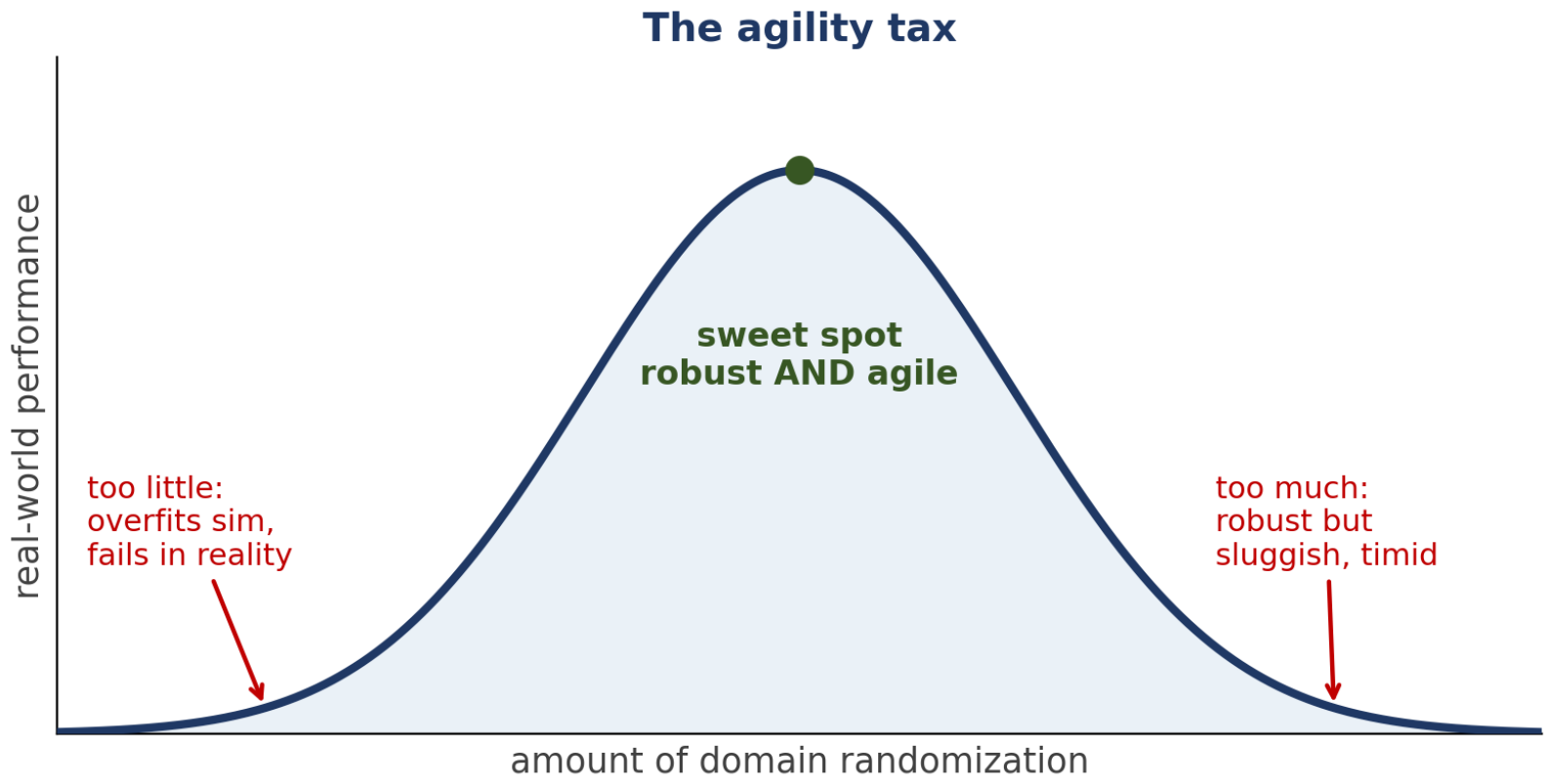
Trained entirely in simulation. No real-world fine-tuning. Transferred zero-shot — robust even to kicks and to deflating the ball.

Does It Beat Humans?



Automation now matches or beats expert hand-tuning at sim-to-real.

The Cost: The Agility Tax



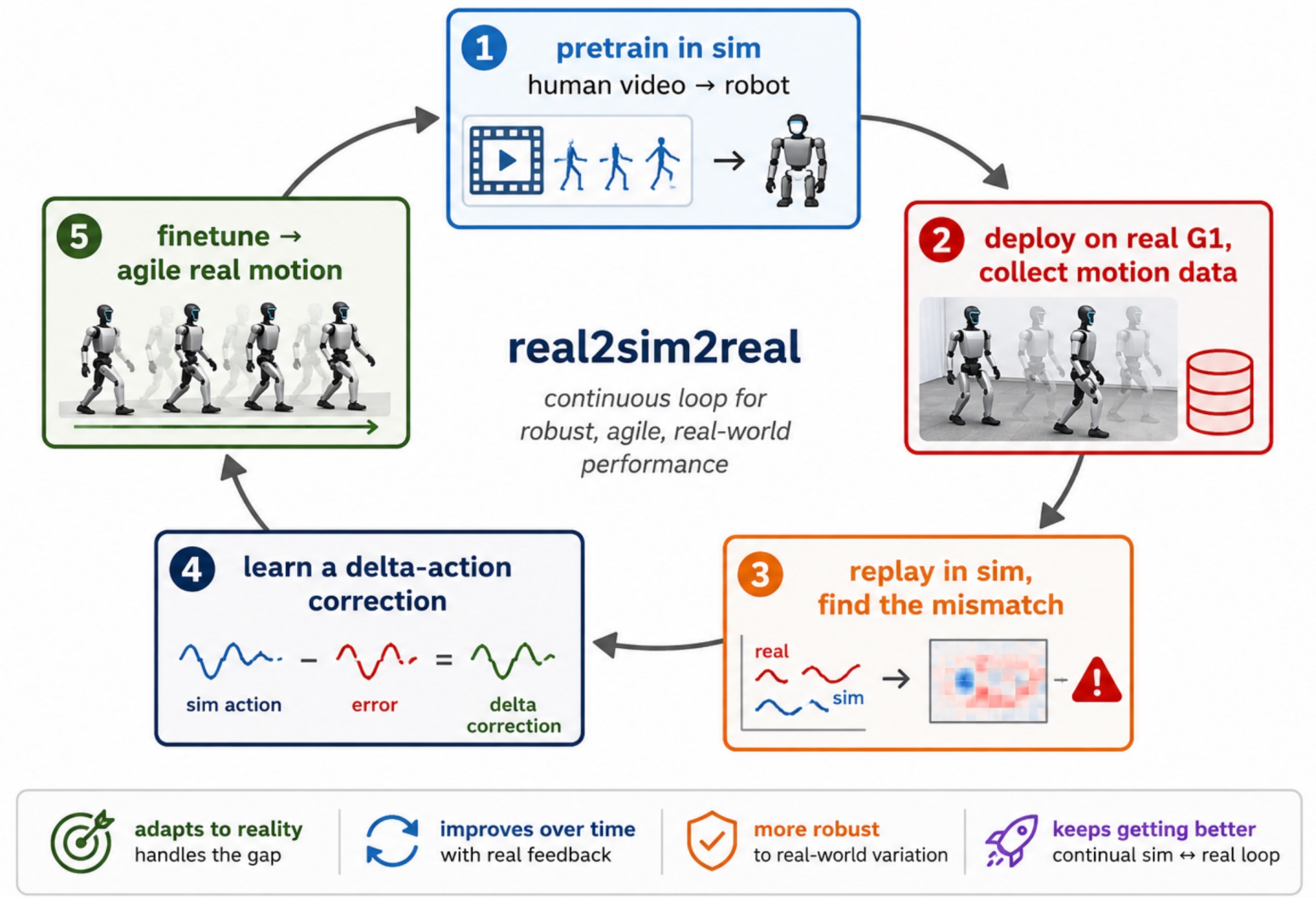
Randomize too hard and you get a robot that survives anything — by doing everything timidly.

4. Closing The Gap from The Real Side

Identify Reality, then Correct

- **Situation:** Domain randomization ignores the real robot entirely — it never looks.
- **Complication:** But we have the real robot. Throwing away that information is wasteful.
- **Question:** *What if we measure where sim and reality disagree, and patch exactly that?*
- **Answer:** System identification (measure real params → plug into sim) and real2sim2real (deploy, record, replay in sim, learn a correction).

Real2Sim2Real



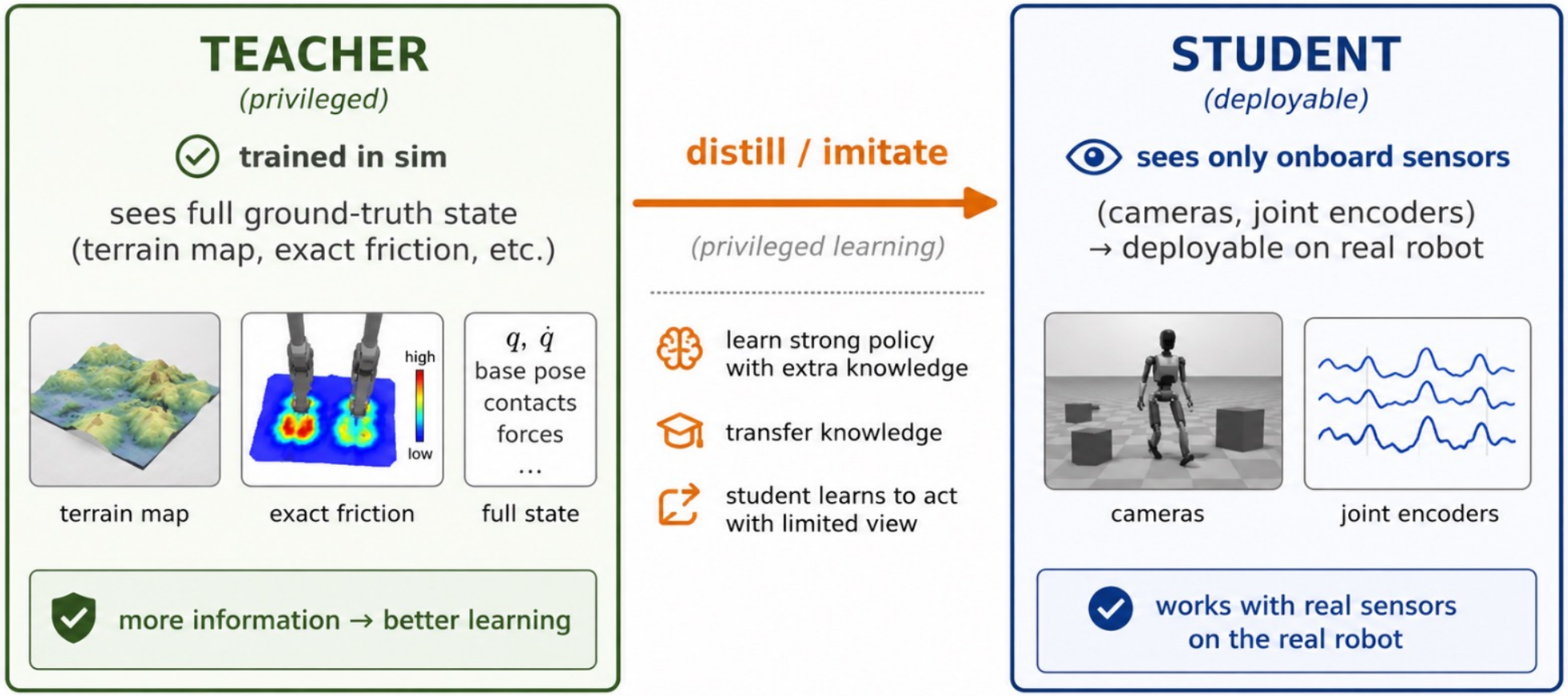
- Don't fix the physics equations.
- Learn a small **correction** that nudges sim behavior toward what the real robot actually does.
- **ASAP** = CMU + NVIDIA, 2025.

ASAP in action

Agile, human-like whole-body motion — learned in sim, corrected with a sip of real data.

Teacher and Student

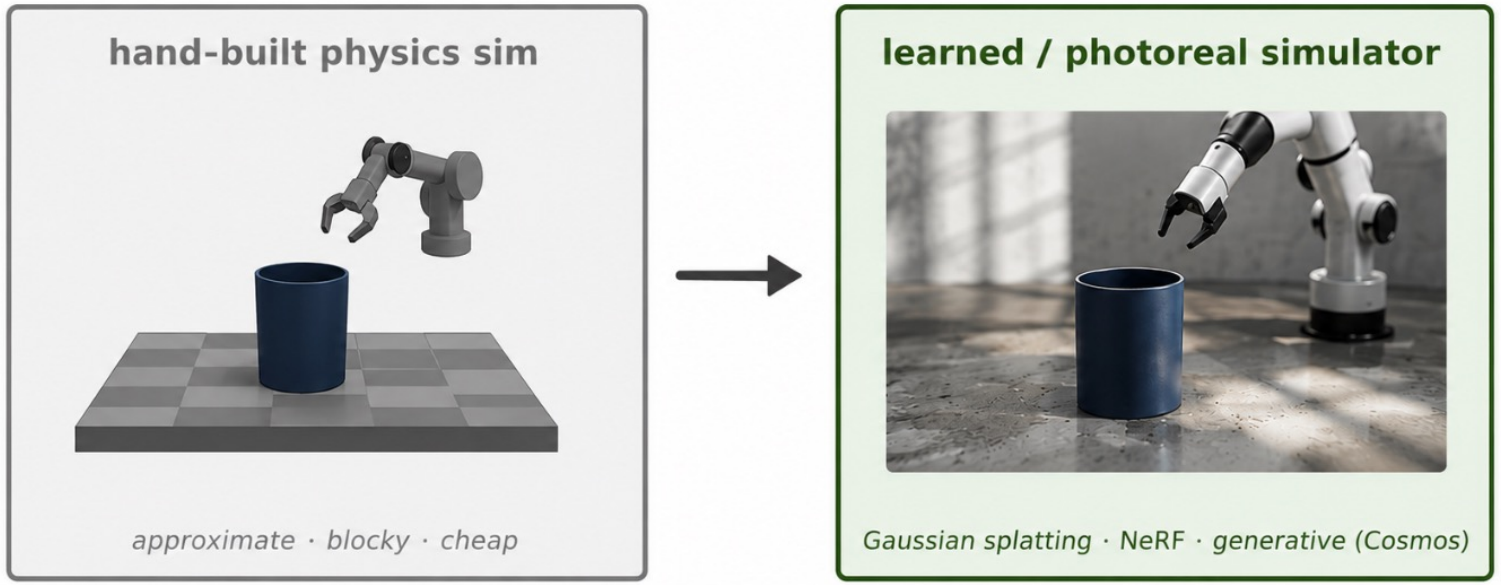
A third lever: train with information you can't deploy with — then distill it away.



Make the Simulator Itself Real

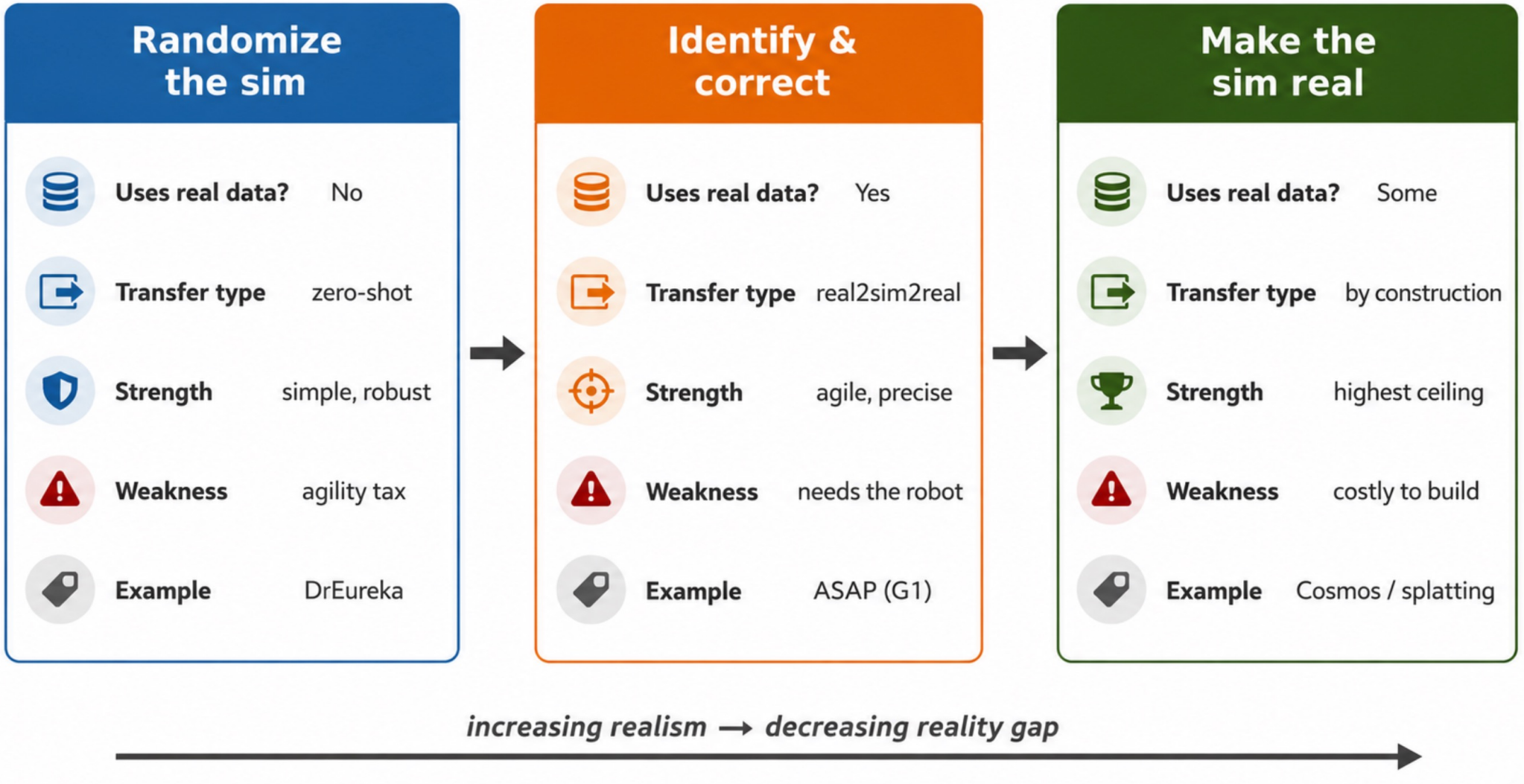
The deepest fix: build simulators reality can't tell apart. Two threads:

- **photoreal digital twins** (reconstruct a real scene, then simulate inside it)
- **learned / generative simulators** (neural models that generate plausible experience, e.g. NVIDIA Cosmos).



If the sim is indistinguishable from reality, the gap disappears by construction.

Three Families



Status

- **Locomotion is largely solved** by sim-to-real — quadrupeds and humanoids walk/run zero-shot.
- **Rich-contact manipulation lags** — grasping deformables, tools, liquids still transfers poorly.
- The gap narrows fastest where physics is well-modeled (rigid contact) and slowest for deformables and fluids.
- Zero-shot demos are real but **cherry-picked**; robustness across scenes is the open problem.

Sim-to-real is winning on legs, still losing on hands.

Summary

- Real robot data is too scarce; **simulation is the default training ground.**
- Every sim is approximate → the **reality gap** (contact, latency, sensors).
- **Domain randomization** drowns the gap — dominant, zero-shot, but pays an agility tax.
- **Real2sim2real & teacher-student** use real data to correct the gap precisely.
- **Learned / photoreal simulators** aim to erase the gap by construction.

Bridge to L40

We've taught one robot one skill in one simulator. L40 asks: can a single model hold every skill, on every robot?

- How do we train one policy that generalizes across tasks and robot bodies?
- What does a “GPT-3 moment” mean for robotics — and is it here?

L40: Robotics Foundation Models.