上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Lecture 38: Perception-to-Action Models

Tao Huang

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

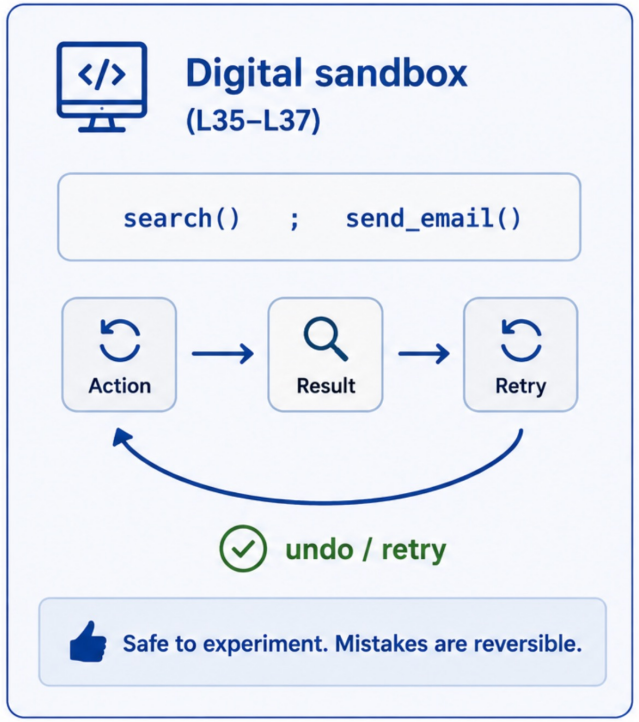
https://taohuang.info/cs3317

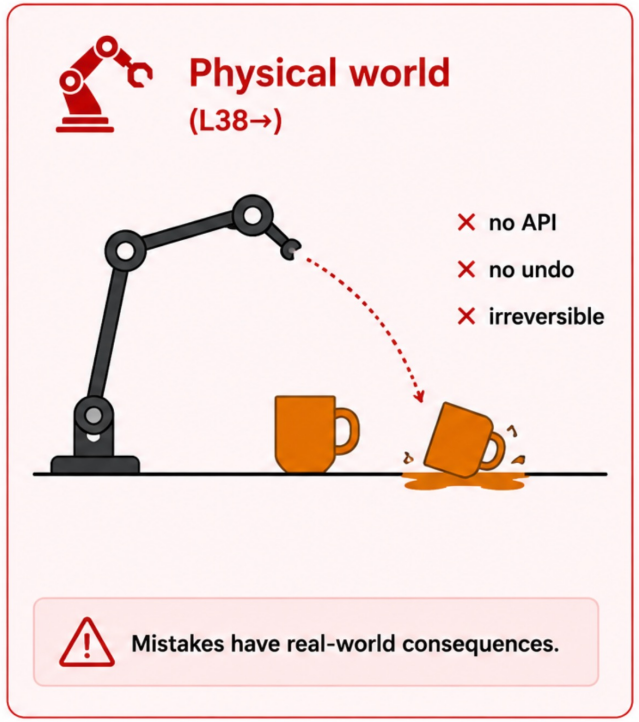
https://oc.sjtu.edu.cn/courses/89538

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: The Agent Grows a Body

- **L35** gave an agent a tool loop.
  L36 made it survive long horizons.
  L37 added many agents + a human.
- All of it lived in a **digital sandbox**.
  Every action was a token — an API call, a message, a line of code.





A digital agent can retry. A robot that knocks over the cup cannot.

What We're Explaining Today

Today: how a single neural network turns what a robot sees into what a robot does.

No hand-coded rules for "grasp the egg." It learned this. How?

Objectives

By the end of this lecture, you will be able to:

- Distinguish a digital action from a physical action — and explain why the physical case is harder.
- Explain why end-to-end learned policies replaced the classical perceive→plan→control pipeline.
- Describe how imitation learning turns human demonstrations into a robot policy.
- Explain the core idea of a Vision-Language-Action (VLA) model and the dual-system design.
- Evaluate a robot demo critically — separating capability from reliability.

1. The Embodiment Gap

The World Has No API

- **Situation:** Our L35–L37 agent acted by calling tools — search( ), send_email( ). Discrete, named, reversible.
- **Complication:** The physical world exposes no API.
  - Actions are continuous (joint torques) and fast (10–50×/sec);
  - perception is raw (millions of pixels);
  - the world is partial, noisy, unforgiving.
- **Question:** *What does an "action" even look like here — and what produces it?*

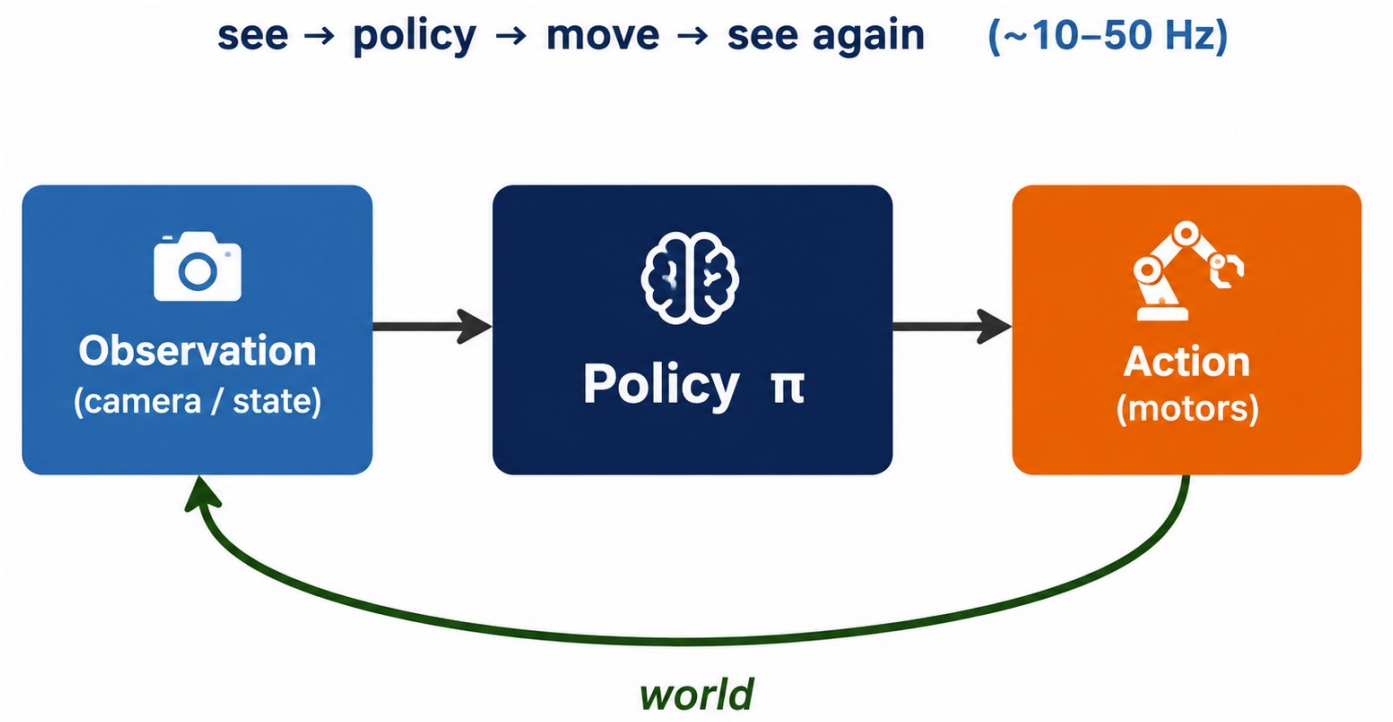
Two Kinds of Action, Side by Side

	Digital agent (L35–L37)	Physical robot
Action	`book_flight( )`	7 joint torques, 50×/sec
Feedback	text result	camera + force, noisy
Mistakes	retry / undo	irreversible
Timing	seconds OK	milliseconds

Same idea — perceive, decide, act — but every column got harder.

It's All One Function: The Policy

- The object of this whole module is a policy π that maps observation → action, run in a tight loop.

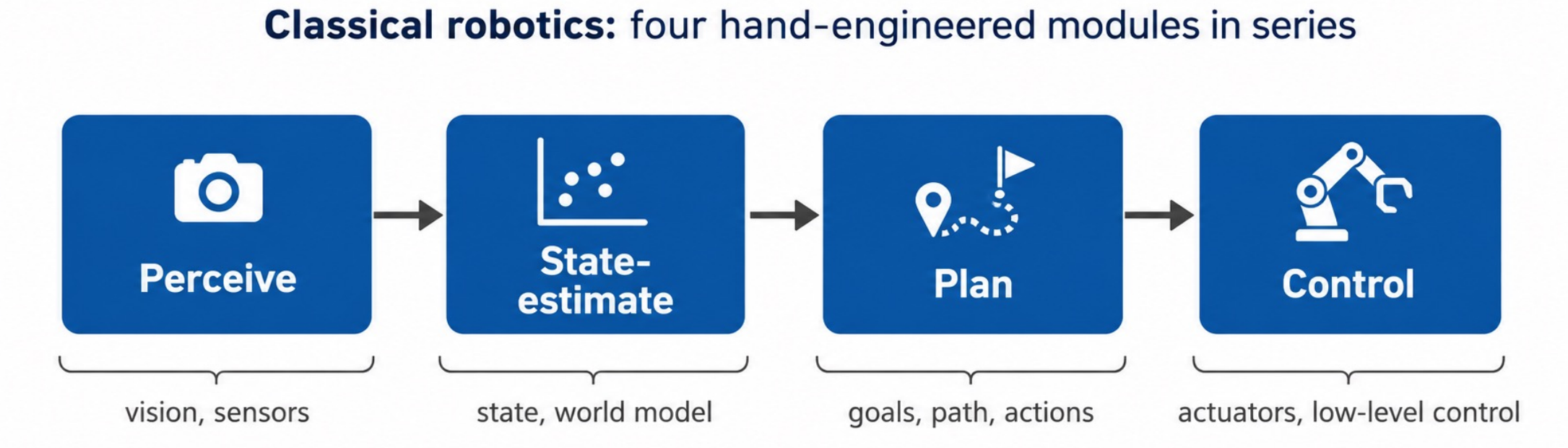


Perception-to-action model = the learned π that closes the loop, sensors → motors.

2. The Classical Pipeline

How Robots Were Built for 40 Years

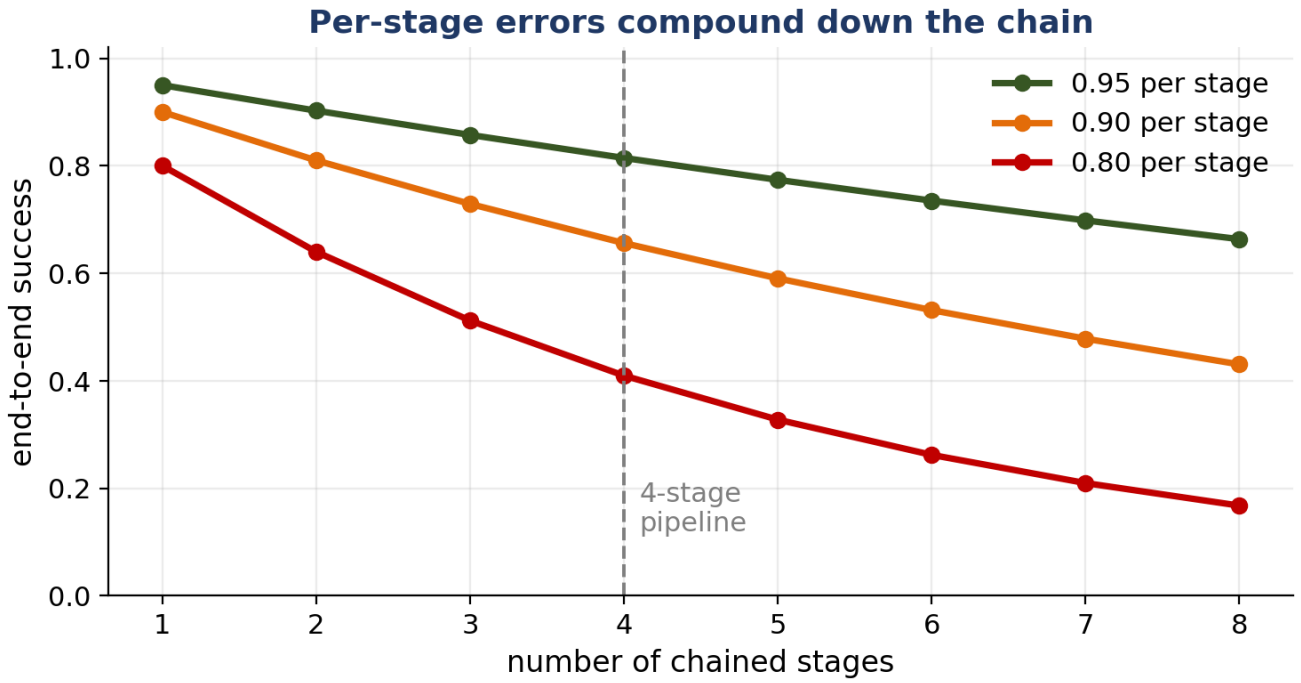
- The classical stack: four hand-engineered modules in series, each separately tuned.



Example "pick up the red block": detect block → estimate pose → plan a path → drive the motors.

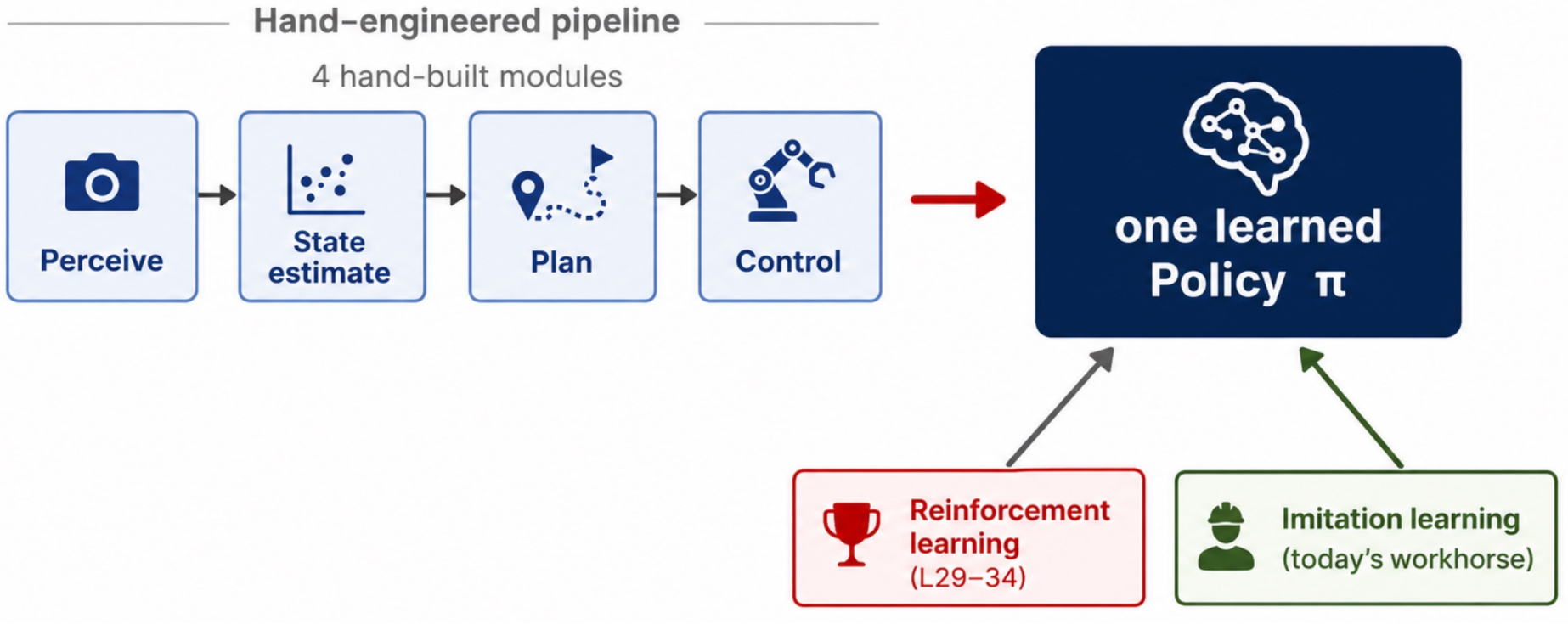
Brittle — and Errors Compound

- Each module assumes the previous one was right. A new object, light, or bit of clutter breaks one stage — and the failure cascades.
- Callback to L36: the same error-compounding intuition — small per-stage error, multiplied down a chain, becomes failure.



Learn The Whole Map End-to-End

- **Question:** *Can one learned function replace the entire pipeline?*
- **Answer:** Yes — an end-to-end policy trained from data. Two ways to get that data:



For manipulation, showing beats rewarding.

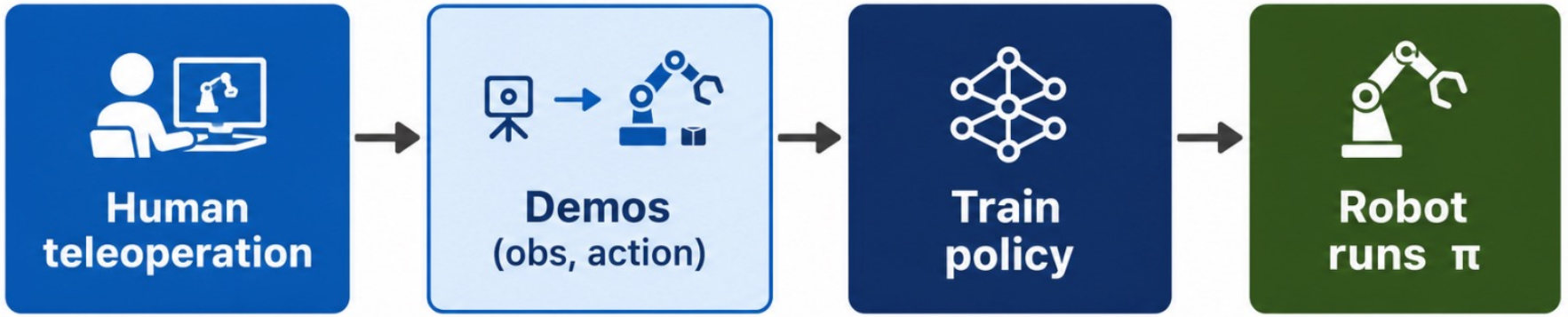
Learning from Demonstration

~50 human demonstrations per task. No reward function, no hand-coded controller — just imitation.

3. Demonstrations In, Behavior Out

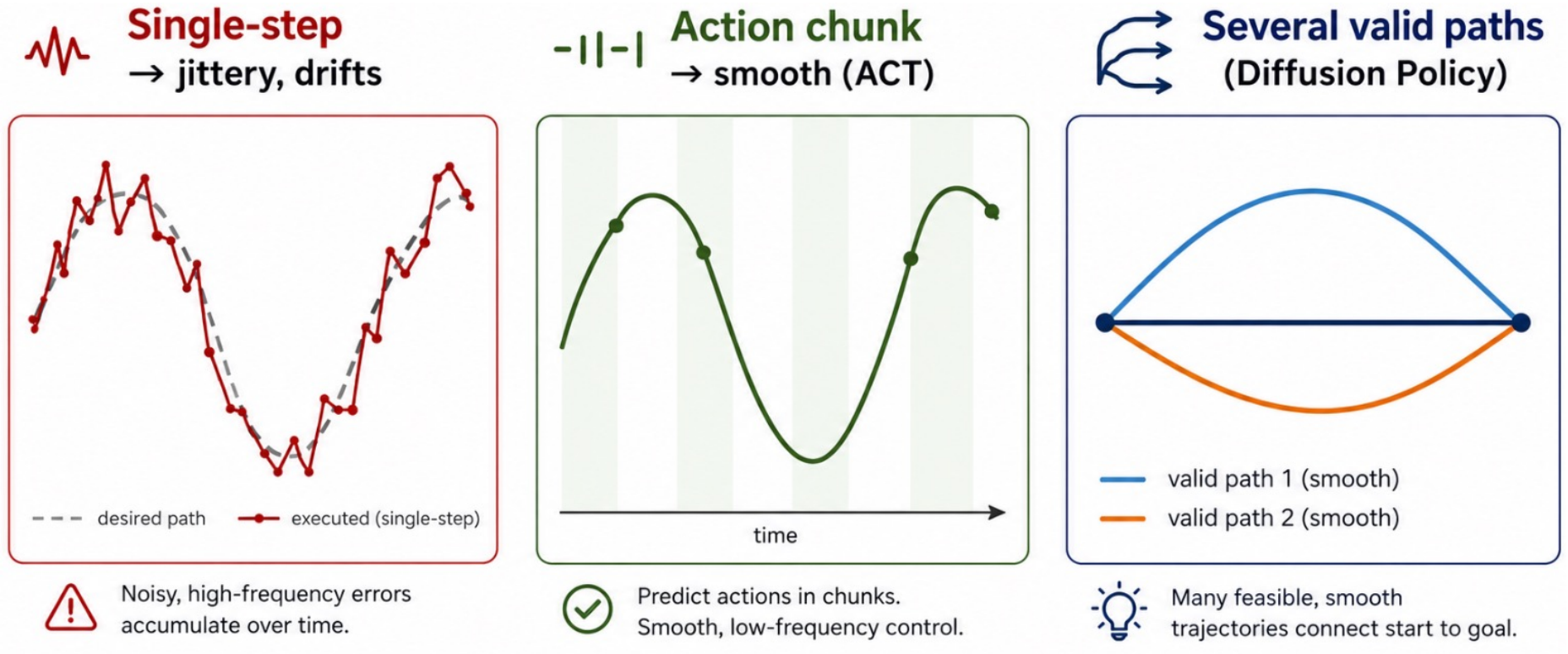
Teleoperation → Behavior Cloning

1. A human puppets the robot through a task (teleoperation), recording observation–action pairs.
2. Train the policy to reproduce the human's action for each observation — behavior cloning.
3. Deploy the loop on the robot.



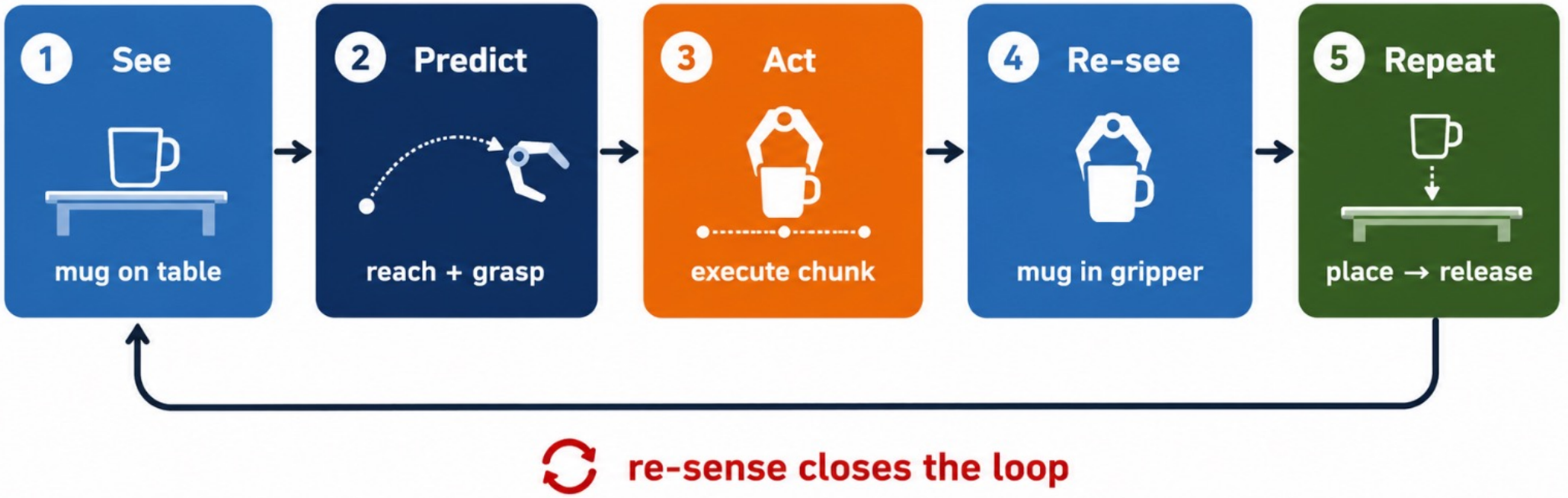
Predict a Chunk, Not a Step

- **Action chunking (ACT):** predict the next short burst of actions, not one twitch at a time → smoother, less drift.
- **Diffusion Policy:** generate the action the way image models generate pictures — handles "several good ways to do it."



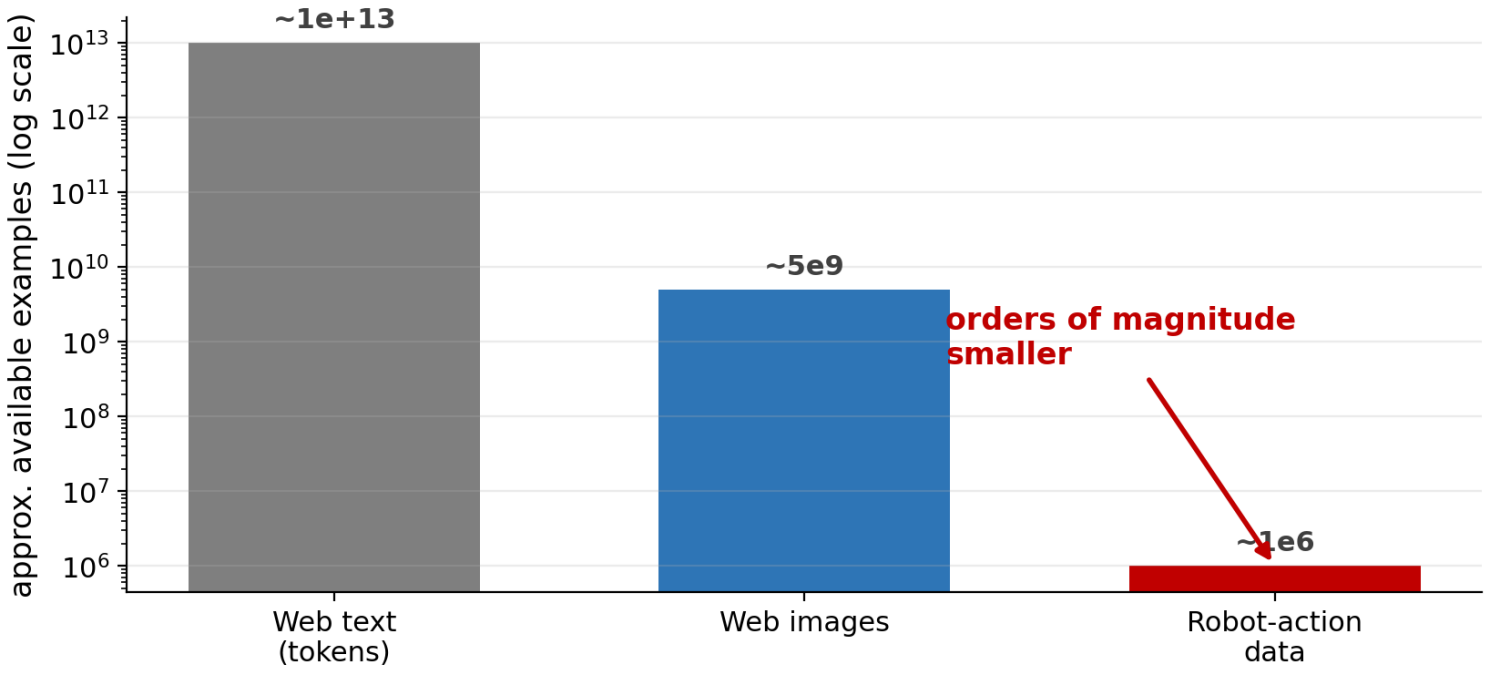
Worked Example: Pick-and-Place

- Walk one full cycle — the policy never plans symbolically; it maps each image to the next motions.



There is No Internet of Robot Data

- Text / image models train on ~the whole web. Robot data must be physically collected, one teleoperated hour at a time.
- **"Data islands":** every lab, robot, and gripper produces incompatible data.



Think: The Data Bottleneck

Robot data is the bottleneck.
Name three ways to get more — and a risk of each.

- more teleoperation
- video of humans doing tasks
- simulation
- sharing data across robots

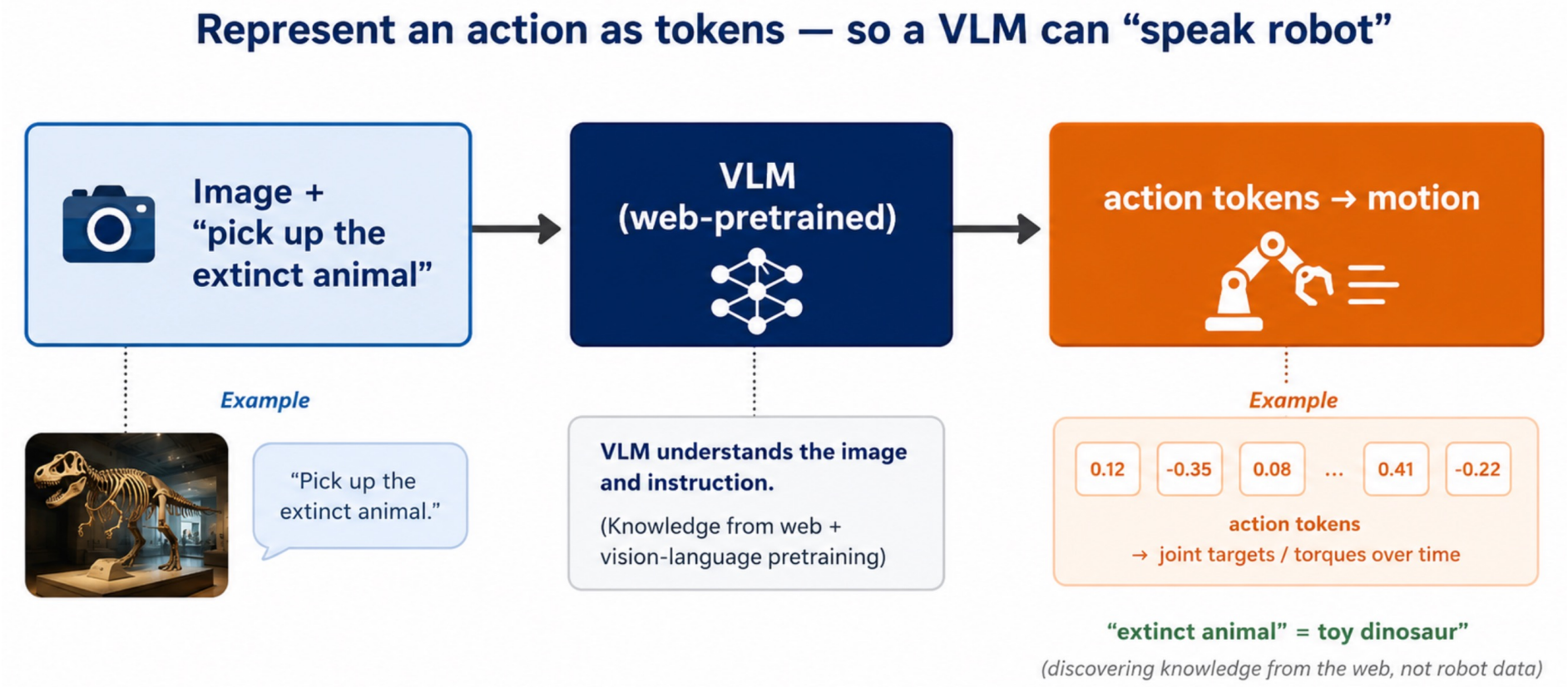
4. Vision-Language-Action Models

Stand on a VLM's Shoulders

- **Situation:** Imitation policies are skilled but narrow — they only know what they were shown.
- **Complication:** They lack common sense. A from-scratch policy doesn't know "a stapler is graspable."
- **Question:** *Where do we get world knowledge without more robot data?*
- **Answer:** Start from a pretrained VLM (callback to L28) — it already learned objects, language, and relationships from the web — and teach it to also output actions. That's a VLA model.

RT-2: Teach a Model to Speak Robot

- The key trick (no math): represent a robot action as a sequence of tokens — just like words.
  Now a VLM can emit actions the same way it emits text.



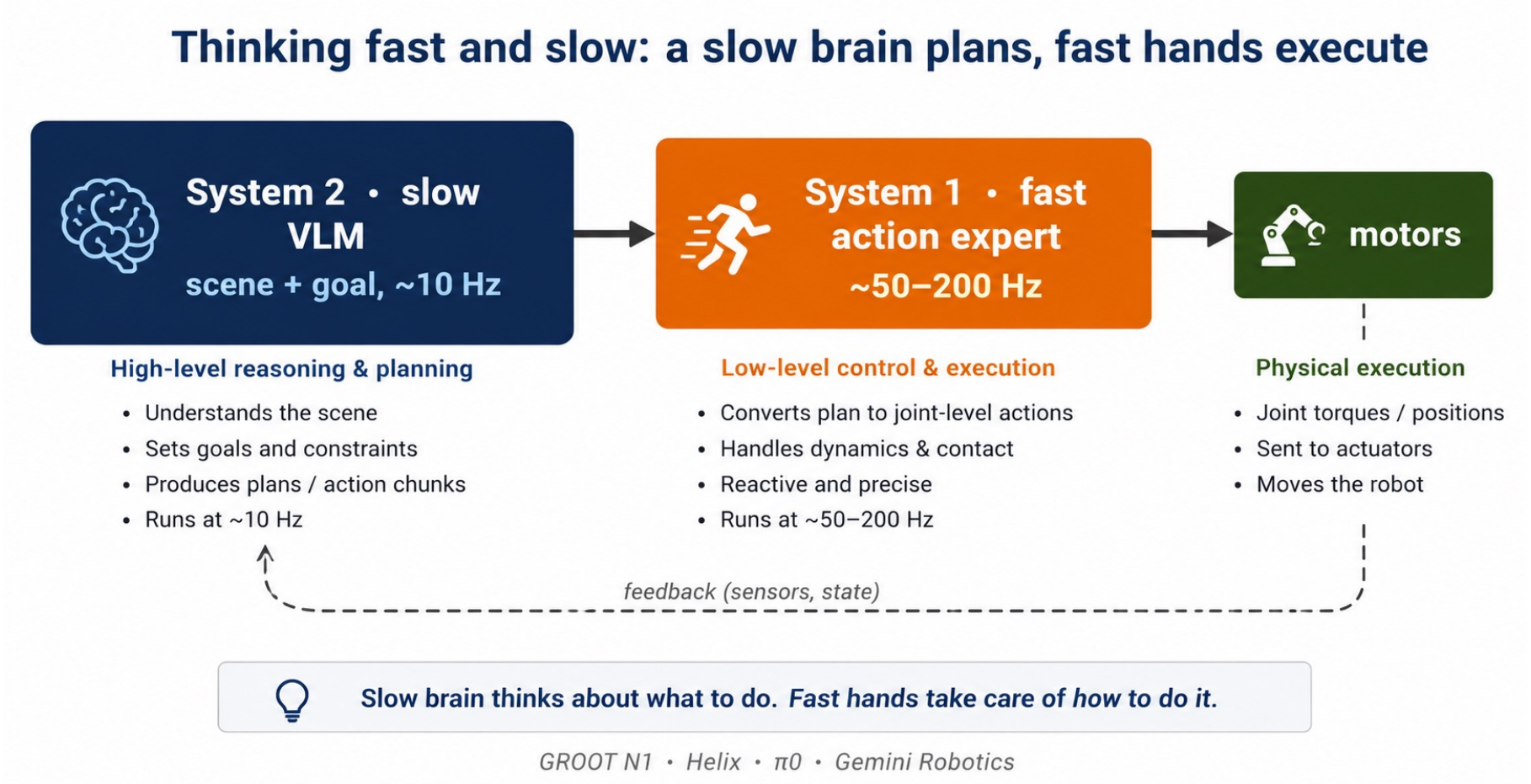
Why This is The Unlock

- The robot can follow an instruction that needs outside knowledge
  — e.g. “move the banana to the number that is 2+1”
  — because the language brain already knows it.

VLA = a VLM that learned to move.
Web knowledge is the cheat code around the data wall.

Slow Brain + Fast Hands

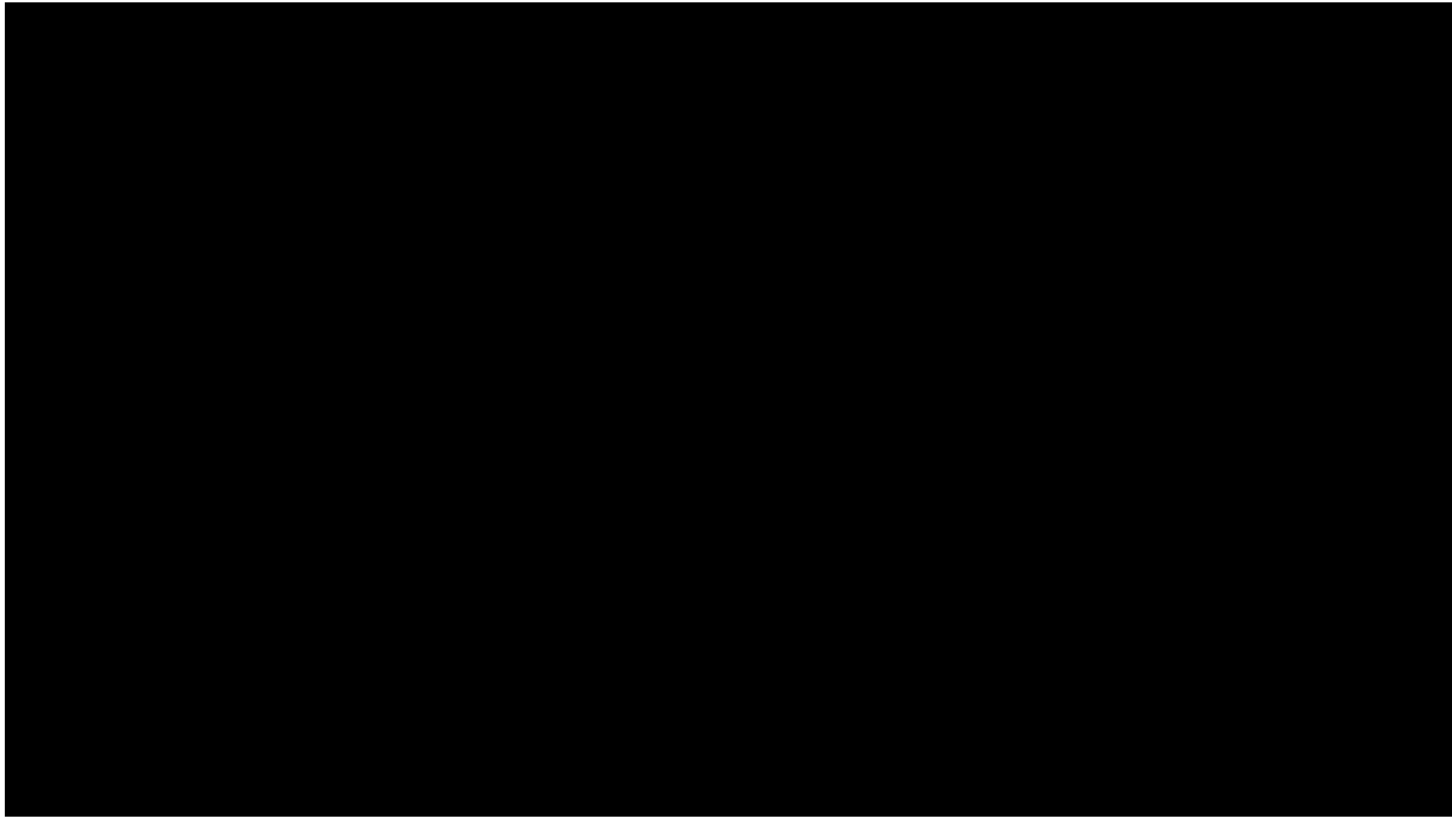
- The architecture nearly everyone converged on by 2025–26 — framed as Kahneman's “thinking fast and slow.”



See It: π0 Folds Laundry

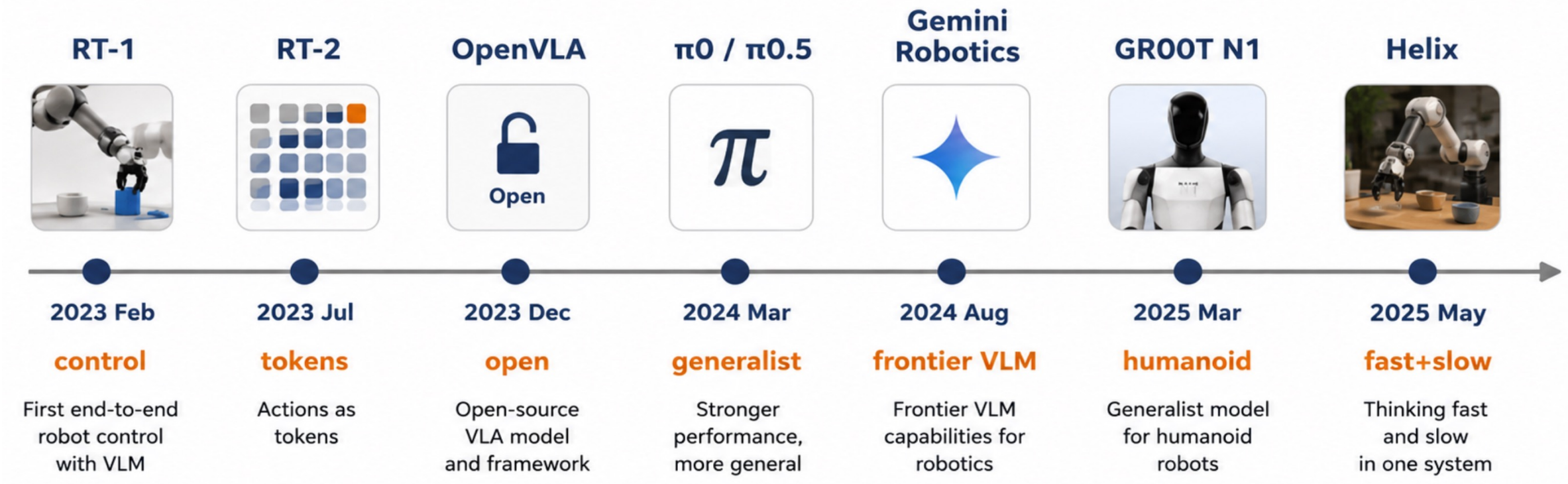
Action expert trained with flow matching — same family as the Diffusion Policy idea from slide 16.

See It: Humanoid Foundation Models



Cross-embodiment: the same VLA family driving arms, mobile bases, and full humanoids.

The VLA Family at a Glance



The Demo-to-Reliability Gap

- Videos are cherry-picked and sometimes teleoperated
  — check the fine print.
- Generalization is still narrow; success drops hard off-distribution.
- Latency forces small models
  — you can't just use the biggest VLM.
- Evaluation is unsolved
  — no “MMLU for robots”; lab numbers aren't comparable.

Summary

1. Physical action is continuous, fast, and irreversible — the hard part.
2. End-to-end learned policies replaced hand-engineered pipelines.
3. Imitation learning (demos → behavior cloning) is today's workhorse for manipulation.
4. Data scarcity is the binding constraint of the whole field.
5. VLA models borrow web knowledge from a VLM and add a fast action expert — slow brain, fast hands.

Frontier: What's Moving Now

- Cross-embodiment generalists (one model, many robot bodies).
- On-device VLAs — squeezing the slow brain to run locally.
- Learning from human video, not just robot teleoperation.
- Autonomous self-improvement and safety / physical-constraint compliance.

The Problem This All Rests On

- The data wall is the thing standing between dazzling demos and dependable robots.
- The cheapest escape hatch: simulation — generate millions of robot-hours for free.

But a policy that's perfect in a simulator still fails in the messy real world. Why?

Bridge to L39: Sim-to-Real

L38 gave the robot a policy — but it's starving for data. The cheapest food is simulation.

Two questions for next lecture:

- Why does a policy that's perfect in sim fall apart in reality? (the “reality gap”)
- What tricks — domain randomization, better physics, real-world fine-tuning — close it?

L39: Sim-to-Real Challenges.