

Reinforcement Learning for Real-World Robotics

Kaifeng Zhang

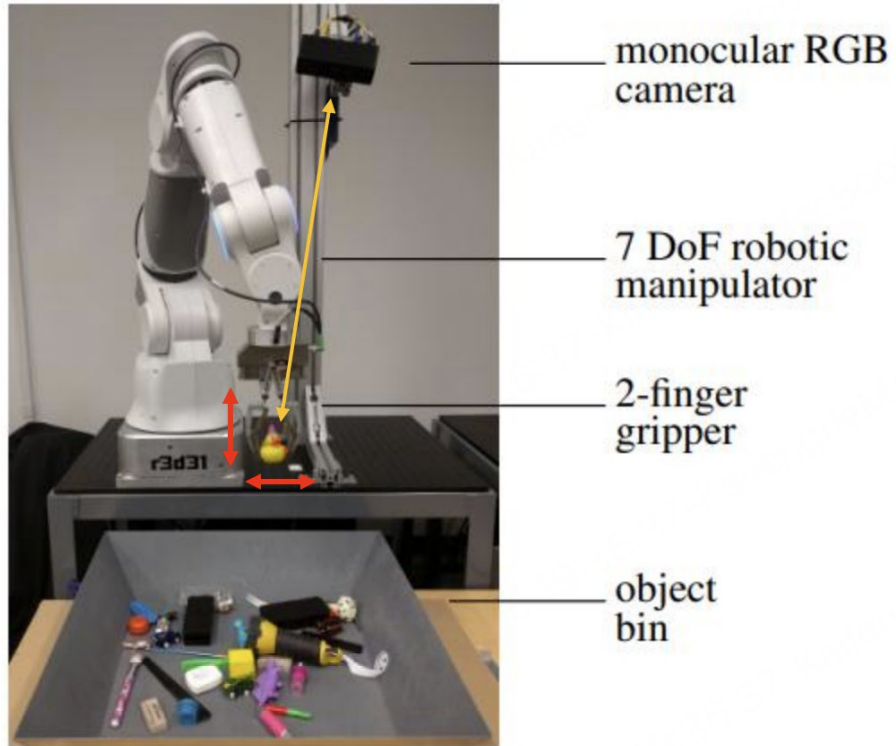


RL Formulation and Algorithms

Practices in Robotics

Frontiers: RL for Dexterous Manipulation

How to Solve a Robotic Task



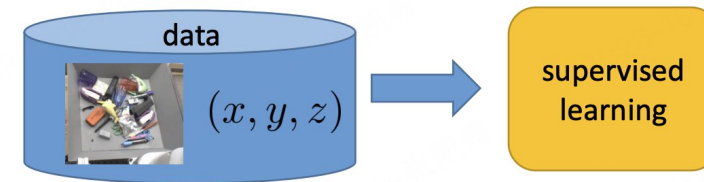
Option 1:

Understand the problem, design a solution

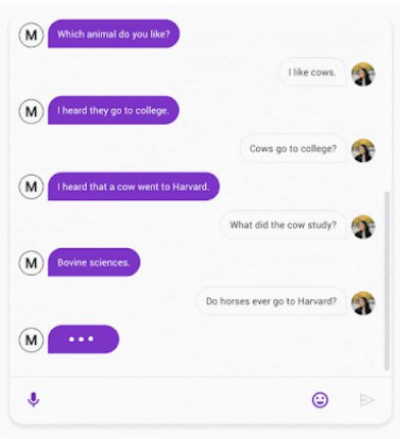


Option 2:

Set it up as a machine learning problem



The Main Idea Behind Robot Learning



Explaining a joke

Prompt

Explain this joke:

Joke: Did you see that Google just hired an eloquent whale for their TPU team? It showed them how to communicate between two different pods!

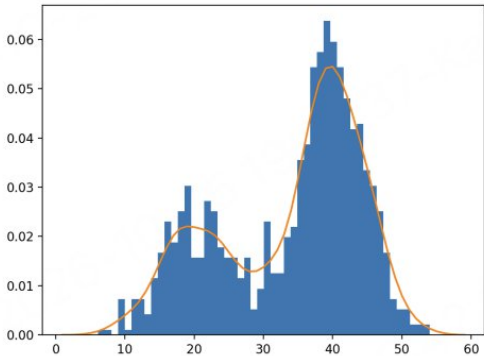
Model Response

Prediction: TPUs are a type of computer chip that Google uses for deep learning. A "pod" is a group of TPUs. A "pod" is also a group of whales. The joke is that the whale is able to communicate between two groups of whales, but the speaker is pretending that the whale is able to communicate between two groups of TPUs.

$$p_{\theta}(\mathbf{x})$$

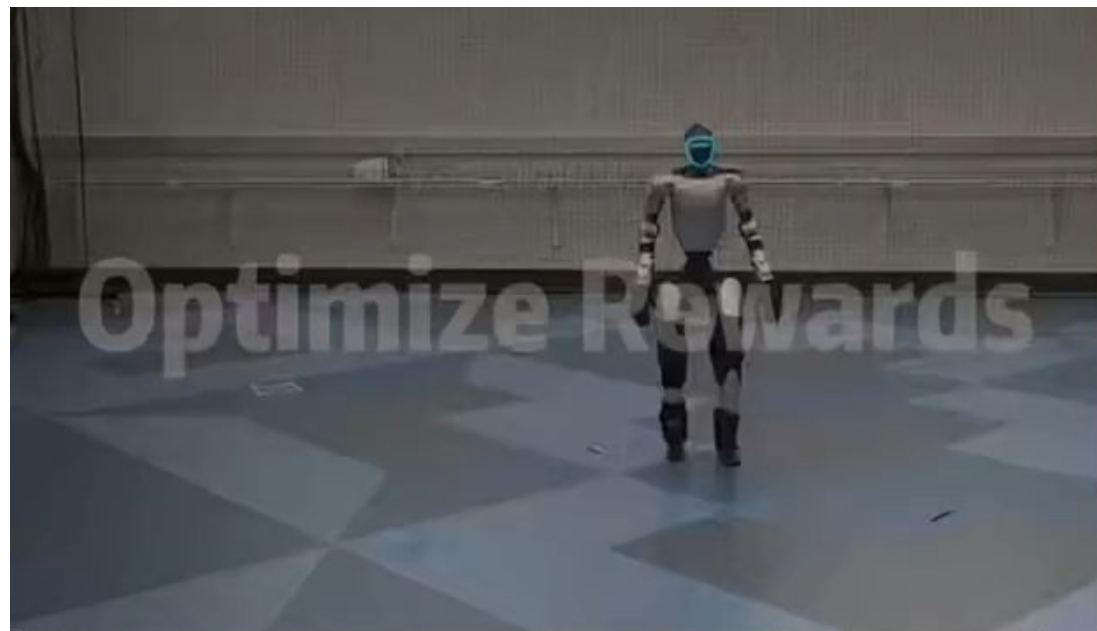
$$p_{\theta}(\mathbf{y}|\mathbf{x})$$

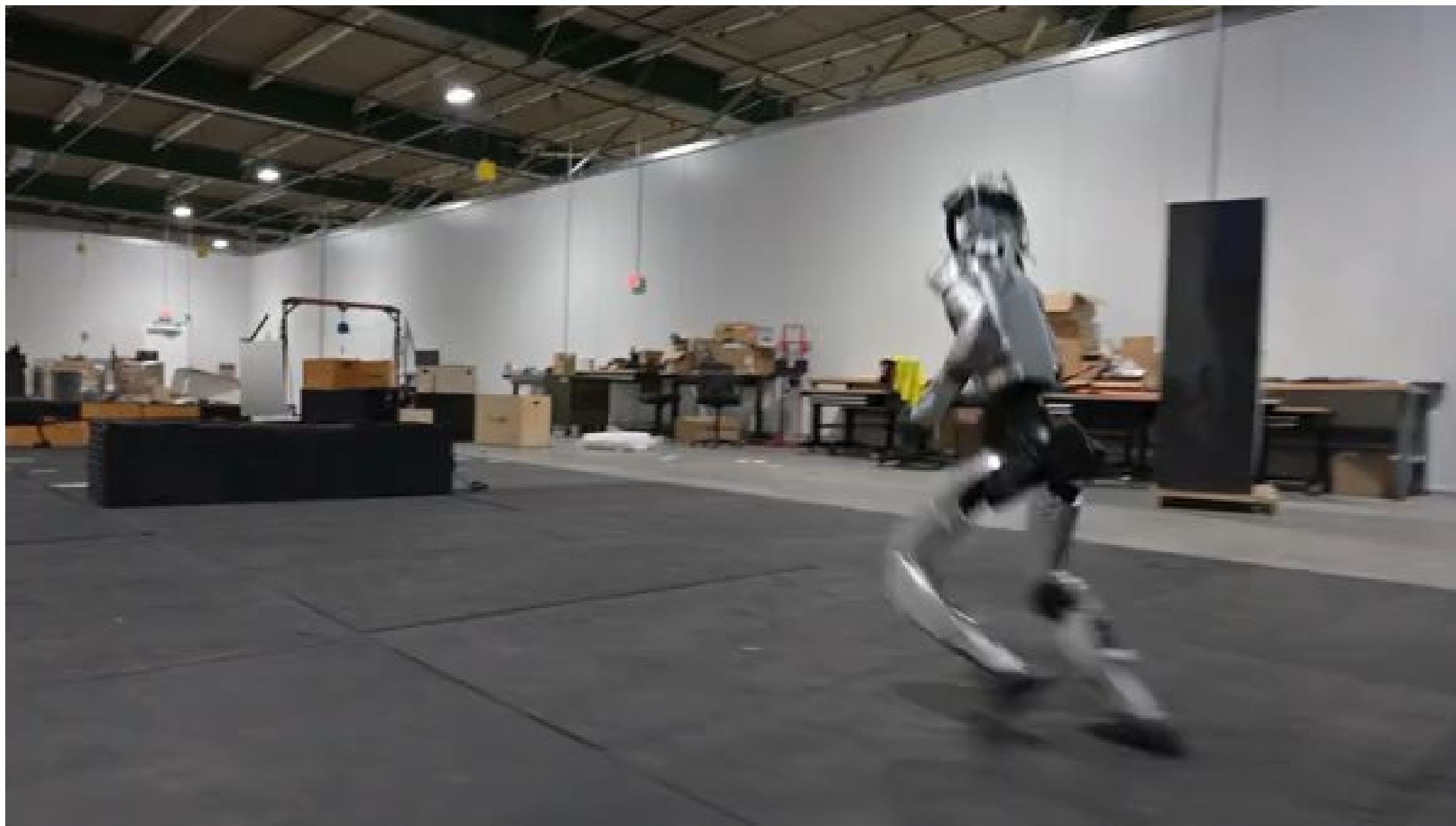
$$\pi_{\theta}(a|o)$$



Why does reinforcement learning do differently?

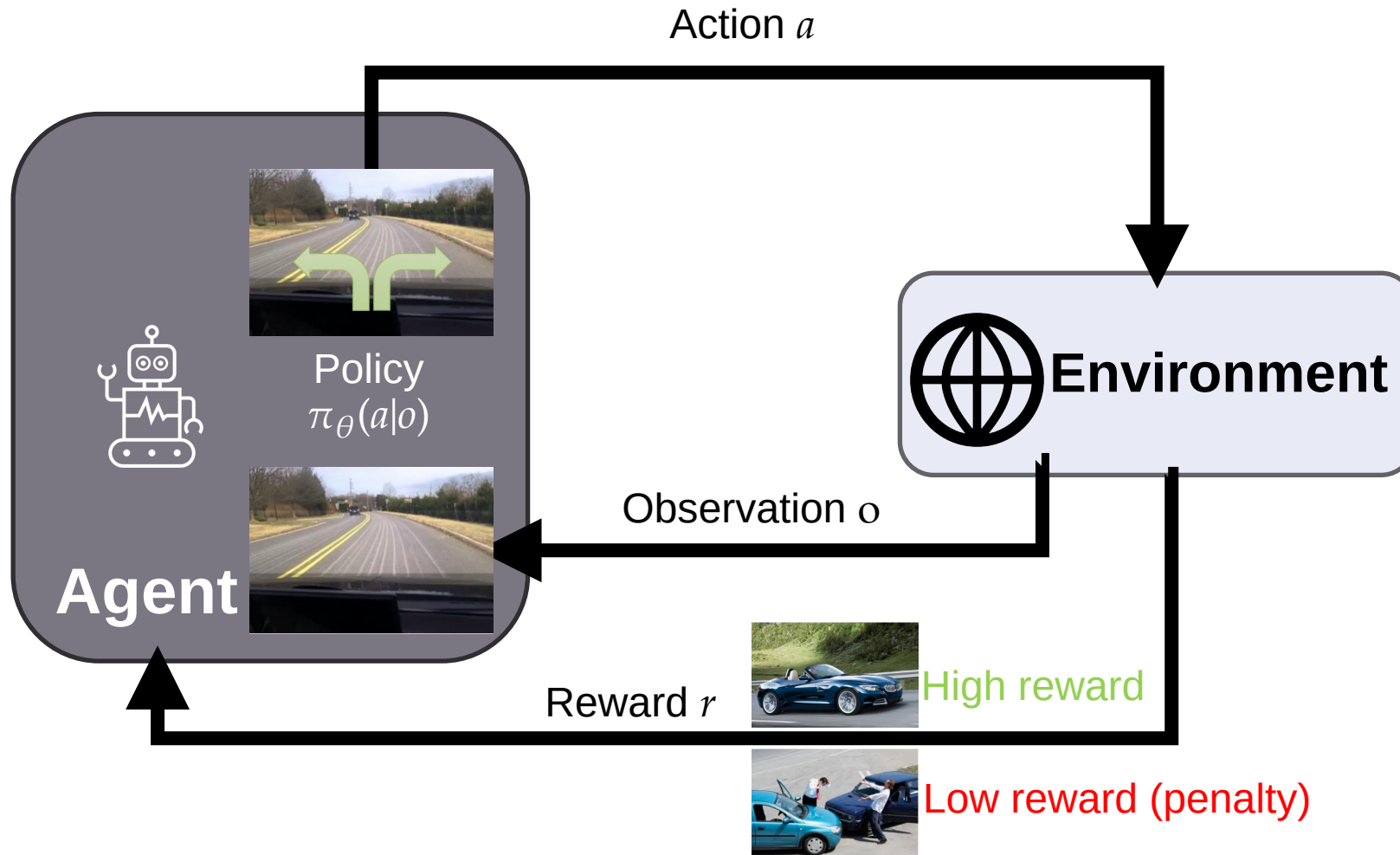
Reactive? Robust? Generalizable?





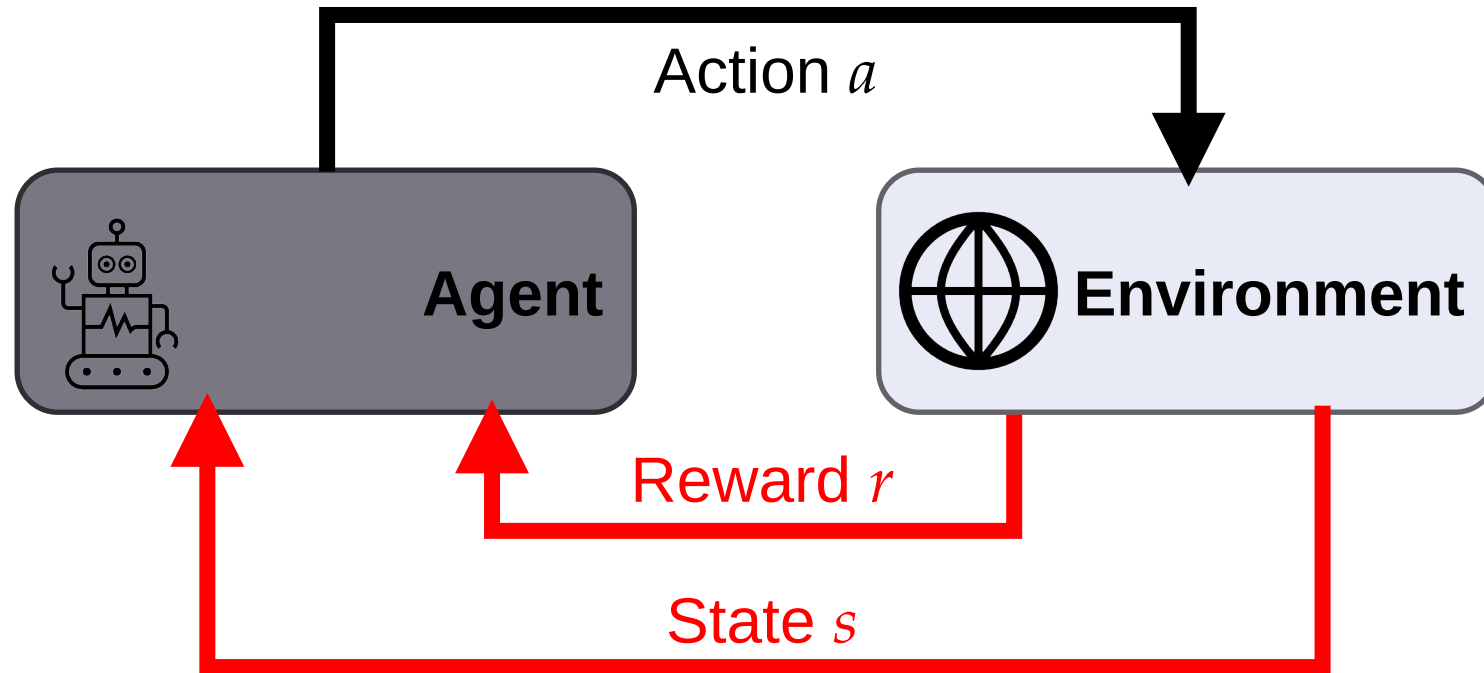
RL Formulation and Algorithms

RL Formulation



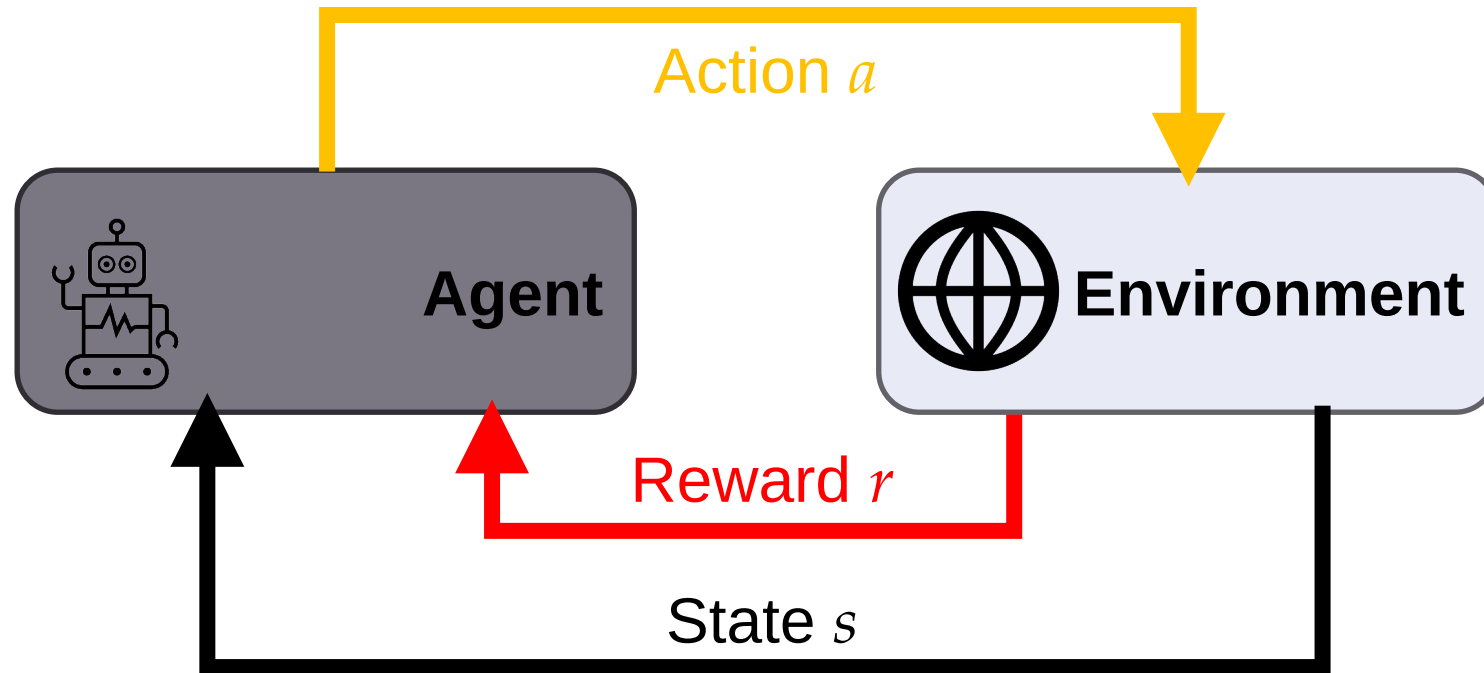
What's special about RL problem

Stochasticity: Rewards and state transitions may be random and unknown



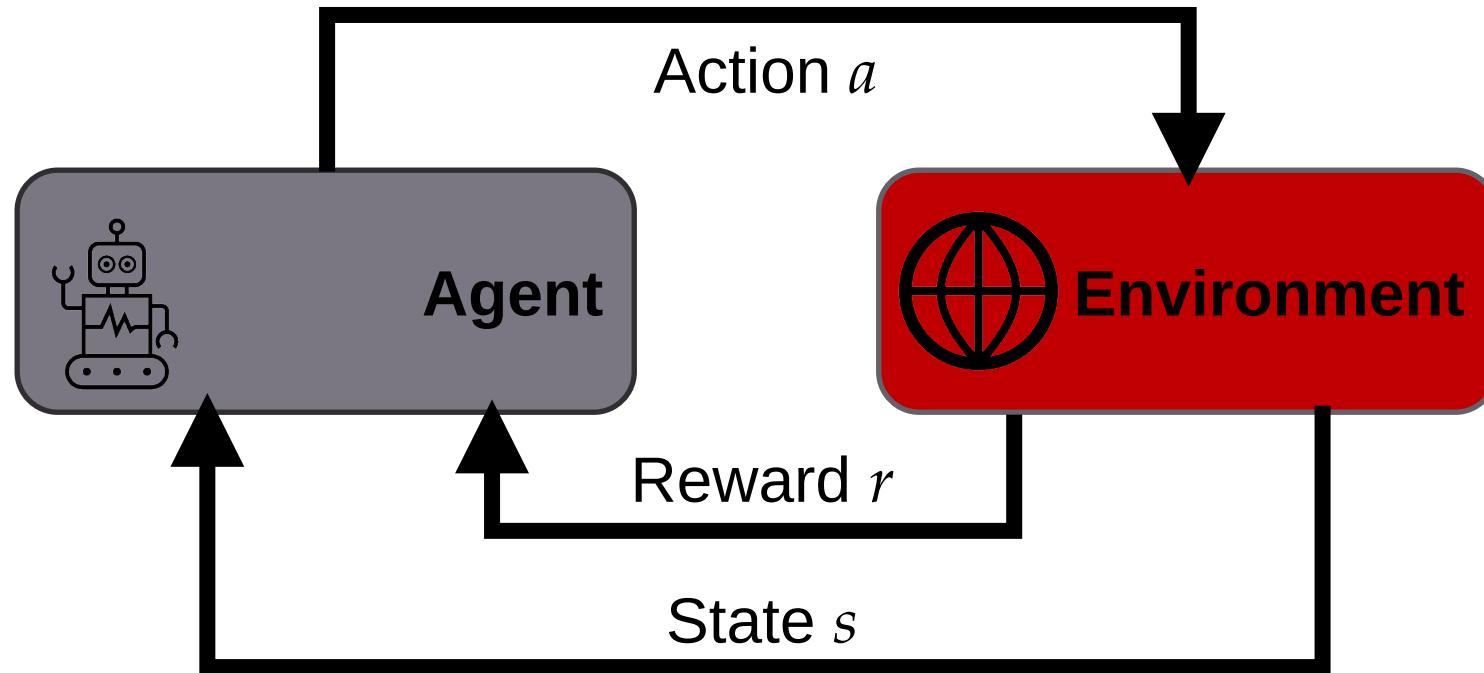
What's special about RL problem

Credit assignment: Reward r_t may not directly depend on action a_t



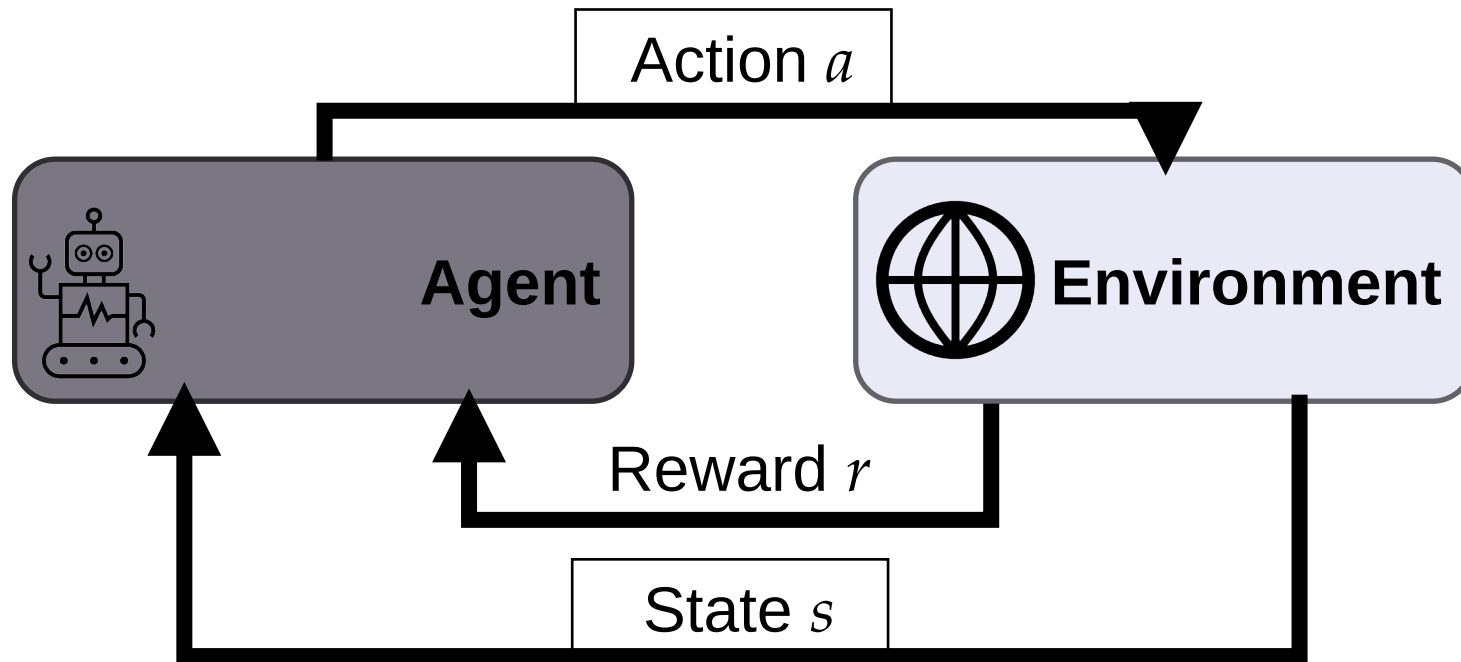
What's special about RL problem

Nondifferentiable: Can't backprop through world; can't compute $\frac{\partial r}{\partial a}$ or $\frac{\partial x}{\partial a}$



What's special about RL problem

Nonstationary: What the agent experiences depends on how it acts



Markov Decision Process

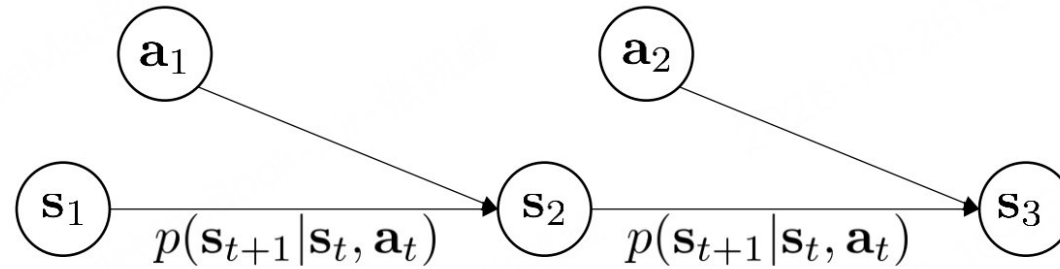
$$\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, r\}$$

states $\mathbf{s} \in \mathcal{S}$

actions $\mathbf{a} \in \mathcal{A}$

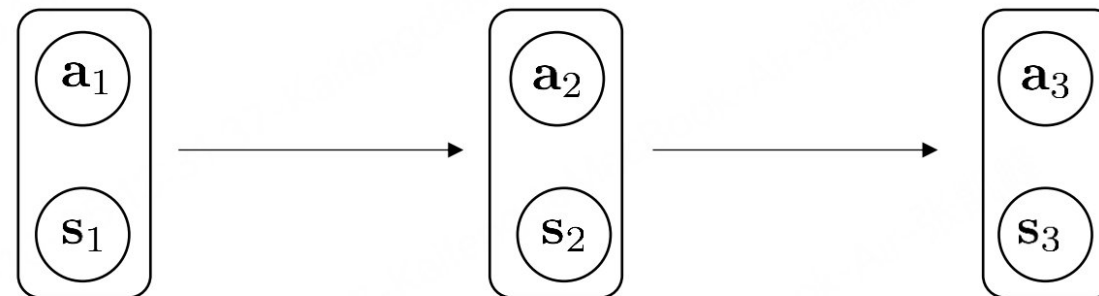
transition operator $\mathcal{T}$

reward function $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$



Richard Bellman

Can we turn a Markov decision process into a Markov chain?



$$p((\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) | (\mathbf{s}_t, \mathbf{a}_t)) = p(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t) \pi_{\theta}(\mathbf{a}_{t+1} | \mathbf{s}_{t+1})$$

The Objective of RL

$r(\mathbf{s}_t, \mathbf{a}_t)$: reward function

tells us which states and actions are better



high reward



low reward

by the time you're about to hit someone, it's too late!

need to optimize for **long-term** reward

$$\theta^* = \arg \max_{\theta} \mathbb{E}_{\tau \sim p_{\theta}(\tau)} \left[\sum_t r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

$$\underbrace{p_{\theta}(\mathbf{s}_1, \mathbf{a}_1, \dots, \mathbf{s}_H, \mathbf{a}_H)}_{p_{\theta}(\tau)} = p(\mathbf{s}_1) \prod_{t=1}^H \pi_{\theta}(\mathbf{a}_t | \mathbf{s}_t) p(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t)$$

Basic Concepts for RL

RL Objective

$$\theta^* = \arg \max_{\theta} E_{\tau \sim p_{\theta}(\tau)} \left[\sum_t r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

Q Function

$$Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t) = \sum_{t'=t}^T E_{\pi_{\theta}} [r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) | \mathbf{s}_t, \mathbf{a}_t]: \text{ total reward from taking } \mathbf{a}_t \text{ in } \mathbf{s}_t$$

Value Function

$$V^{\pi}(\mathbf{s}_t) = \sum_{t'=t}^T E_{\pi_{\theta}} [r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) | \mathbf{s}_t]: \text{ total reward from } \mathbf{s}_t$$

$$V^{\pi}(\mathbf{s}_t) = E_{\mathbf{a}_t \sim \pi(\mathbf{a}_t | \mathbf{s}_t)} [Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t)]$$

$$E_{\mathbf{s}_1 \sim p(\mathbf{s}_1)} [V^{\pi}(\mathbf{s}_1)] \text{ is the RL objective!}$$

Types of RL Algorithms

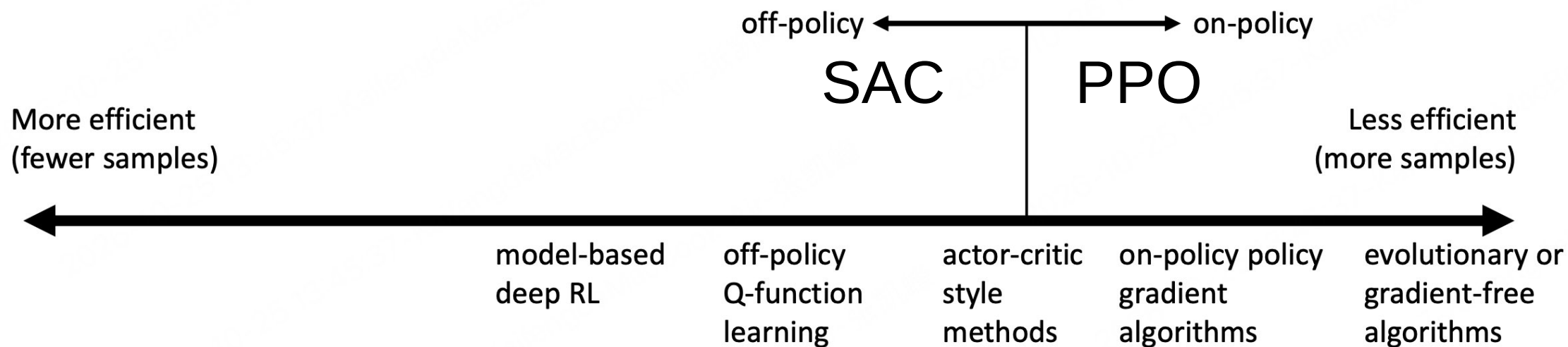
$$\theta^* = \arg \max_{\theta} E_{\tau \sim p_{\theta}(\tau)} \left[\sum_t r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

- Policy gradients: directly differentiate the above objective
- Value-based: estimate value function or Q-function of the optimal policy (no explicit policy)
- Actor-critic: estimate value function or Q-function of the current policy, use it to improve policy
- Model-based RL: estimate the transition model, and then...
 - Use it for planning (no explicit policy)
 - Use it to improve a policy
 - Something else

Examples of Specific RL Algorithms

- Value function methods
 - Q-learning, DQN
 - Temporal difference learning
 - Fitted value iteration
- Policy gradient methods
 - REINFORCE
 - Natural policy gradient
 - PPO
- Actor-critic algorithms
 - Asynchronous advantage actor-critic (A3C)
 - Soft actor-critic (SAC)
- Model-based RL algorithms
 - Dyna (+ “Dyna-like” methods such as MBPO)
 - muZero

Sample Efficiency?



Actor-Critic Style Methods

$$\max J(\theta) = E_{\tau \sim \pi_{\theta}} \left[\sum_t \gamma^t r_t \right]$$

This is the objective function (cumulative return) for RL

$$p_{\theta}(\tau) = p(s_0) \pi_{\theta}(a_0|s_0) \cdot p(s_1|s_0, a_0) \cdots$$

The transition model has nothing to do with θ

$$\nabla_{\theta} \log p_{\theta}(\tau) = \sum_t \nabla_{\theta} \log \pi_{\theta}(a_t|s_t)$$

$$\nabla_{\theta} J = E_{\tau} \left[\left(\sum_t \gamma^t r_t \right) \cdot \sum_t \nabla_{\theta} \log \pi_{\theta}(a_t|s_t) \right]$$

$$\nabla_{\theta} p_{\theta}(\tau) = p_{\theta}(\tau) \nabla_{\theta} \log p_{\theta}(\tau)$$

$$\nabla_{\theta} J = E_{s, a \sim \pi_{\theta}} [\nabla_{\theta} \log \pi_{\theta}(a|s) V(s)]$$

Practices in Robotics

Sim or Real

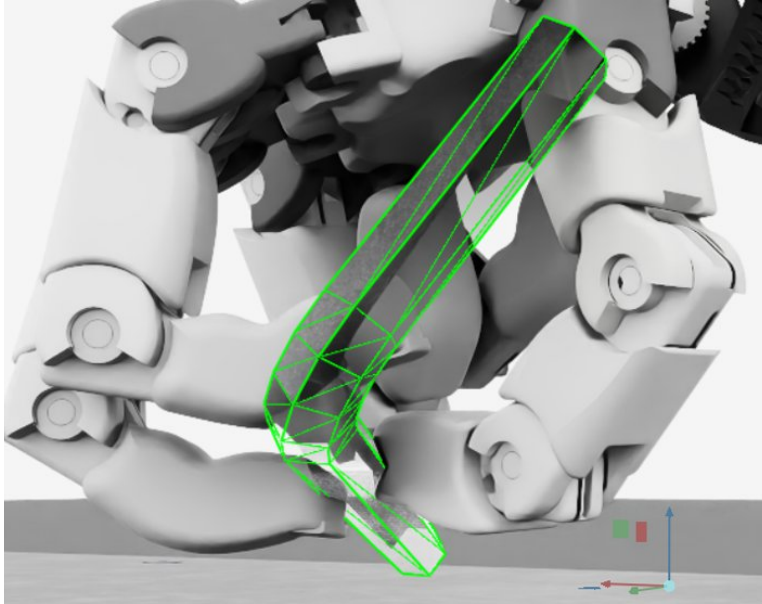


Simulated data

- Cheap
- Fast
- Scalable
- Safe
- Labeled
- Not beholden to real-world probability distributions

Difficulty of using simulated data

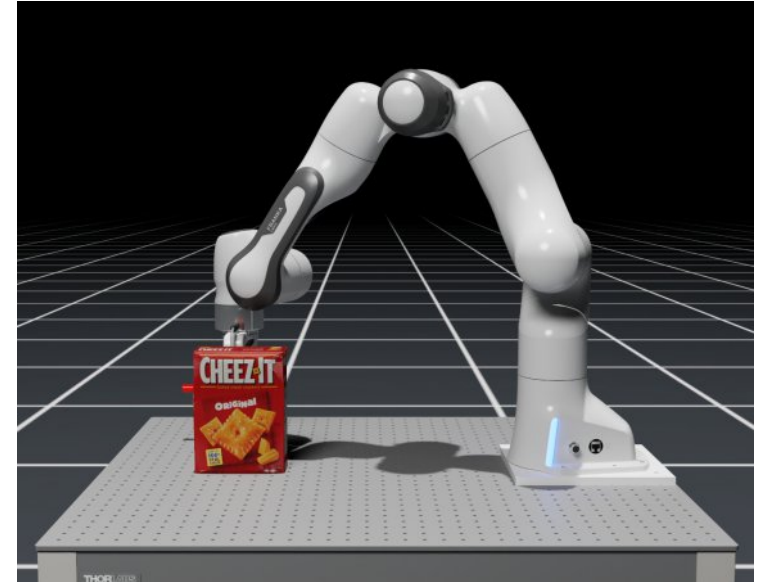
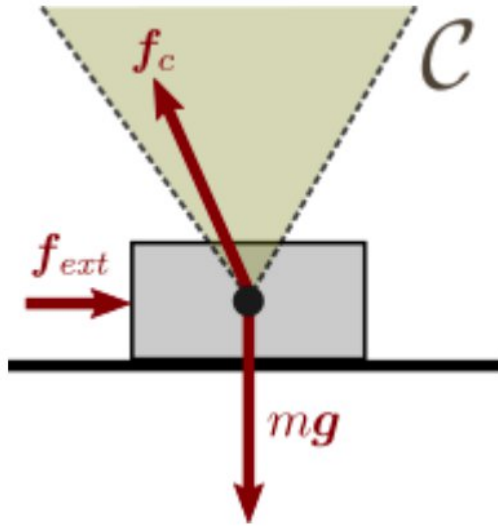
Physics simulators make big assumptions to run faster



FPS: 101.57, Frame time: 9.85 ms
NVIDIA GeForce RTX 4090 D: 1.3 GiB used, 19.1 GiB available
Process Memory: 7.4 GiB used, 115.0 GiB available
1280x720

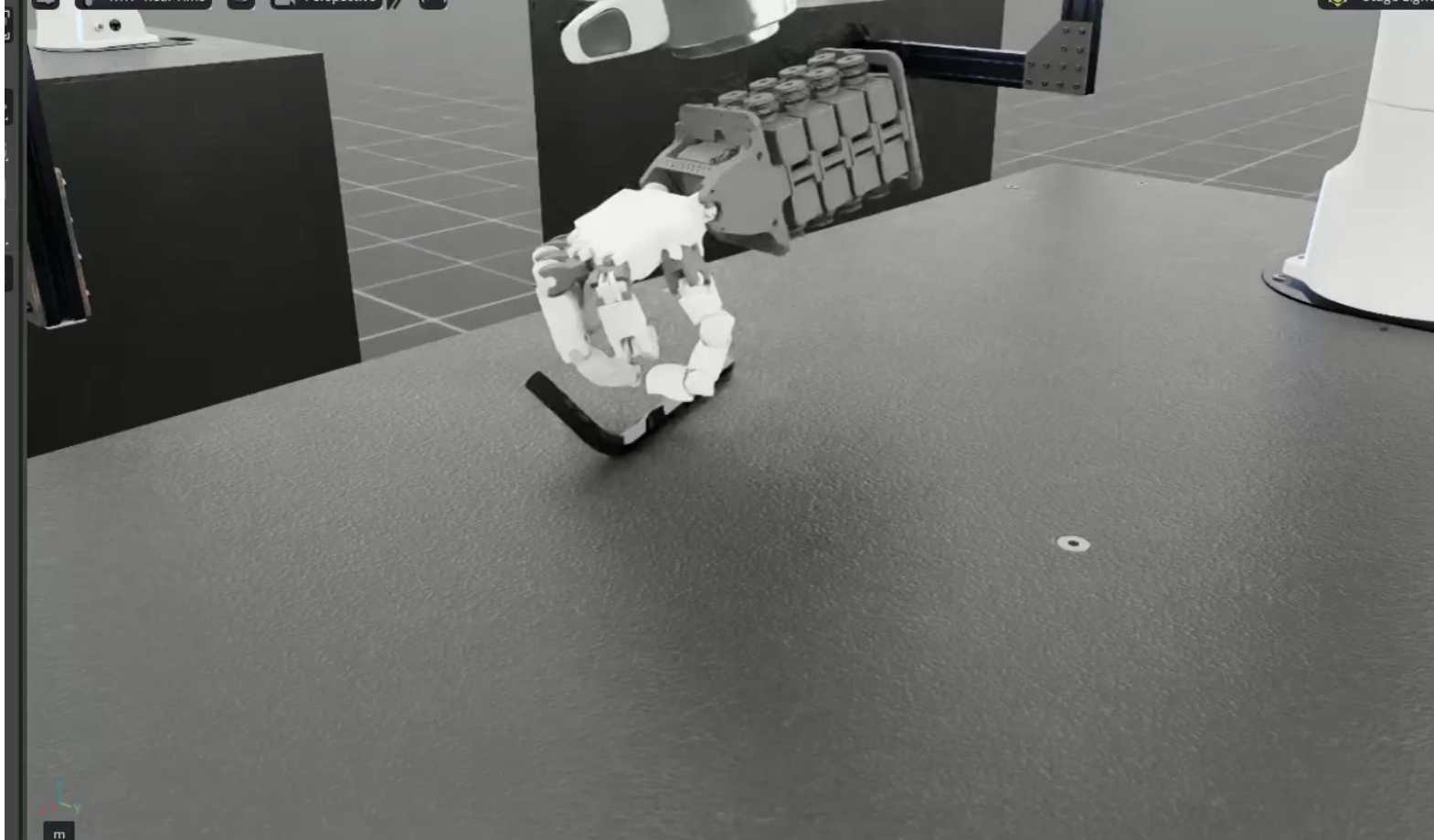
Difficulty of using simulated data

Physics simulators make big assumptions to run faster
Accurate physics model means more parameters to identify



Difficulty of using simulated data

- Neural nets will exploit/overfit to differences in data distributions



SFV: Reinforcement Learning of Physical Skills from Videos

(with audio)



Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik,
Pieter Abbeel, Sergey Levine

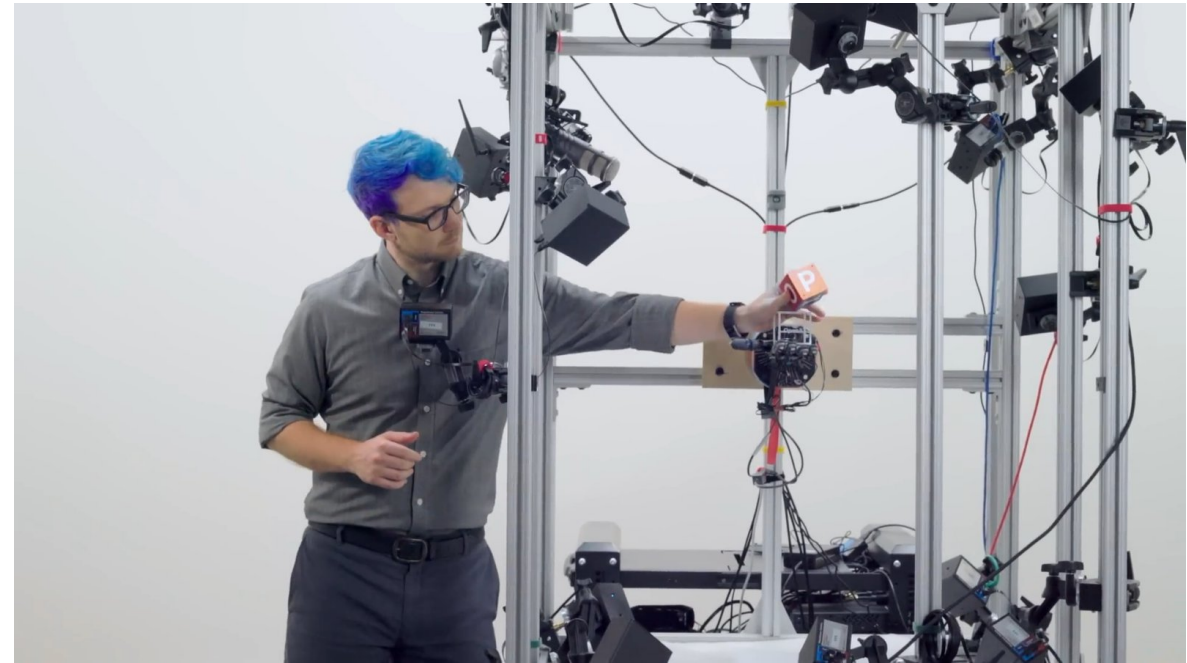
UC Berkeley



Domain Randomization

Idea: if one policy works in millions of simulation environments, this policy may also work in real-world.

Method: Increase the diversity in simulation domains so that the real world **may** look like another simulator.

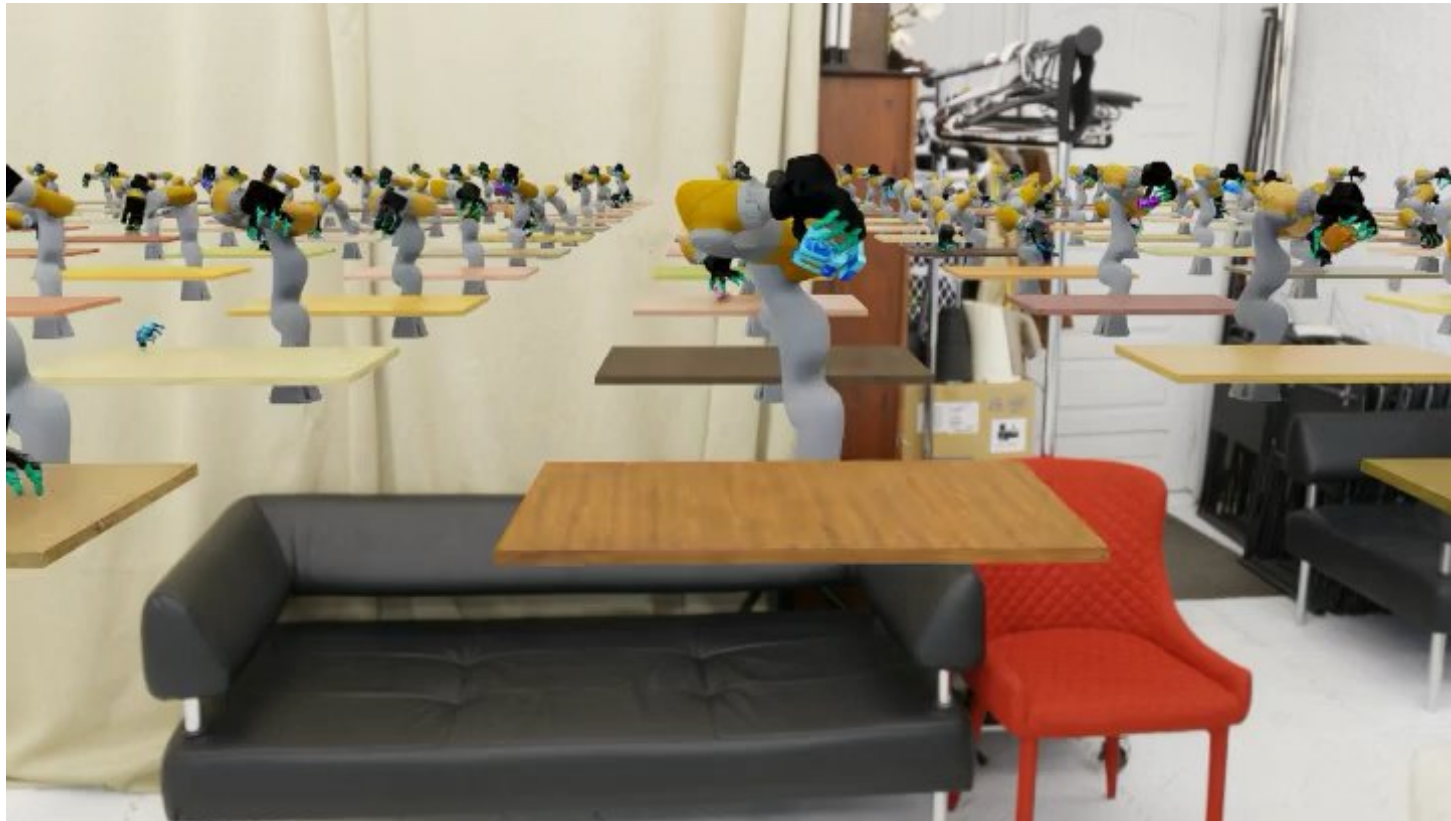


resource: Learning dexterous in-hand manipulation

Teacher Student Distillation

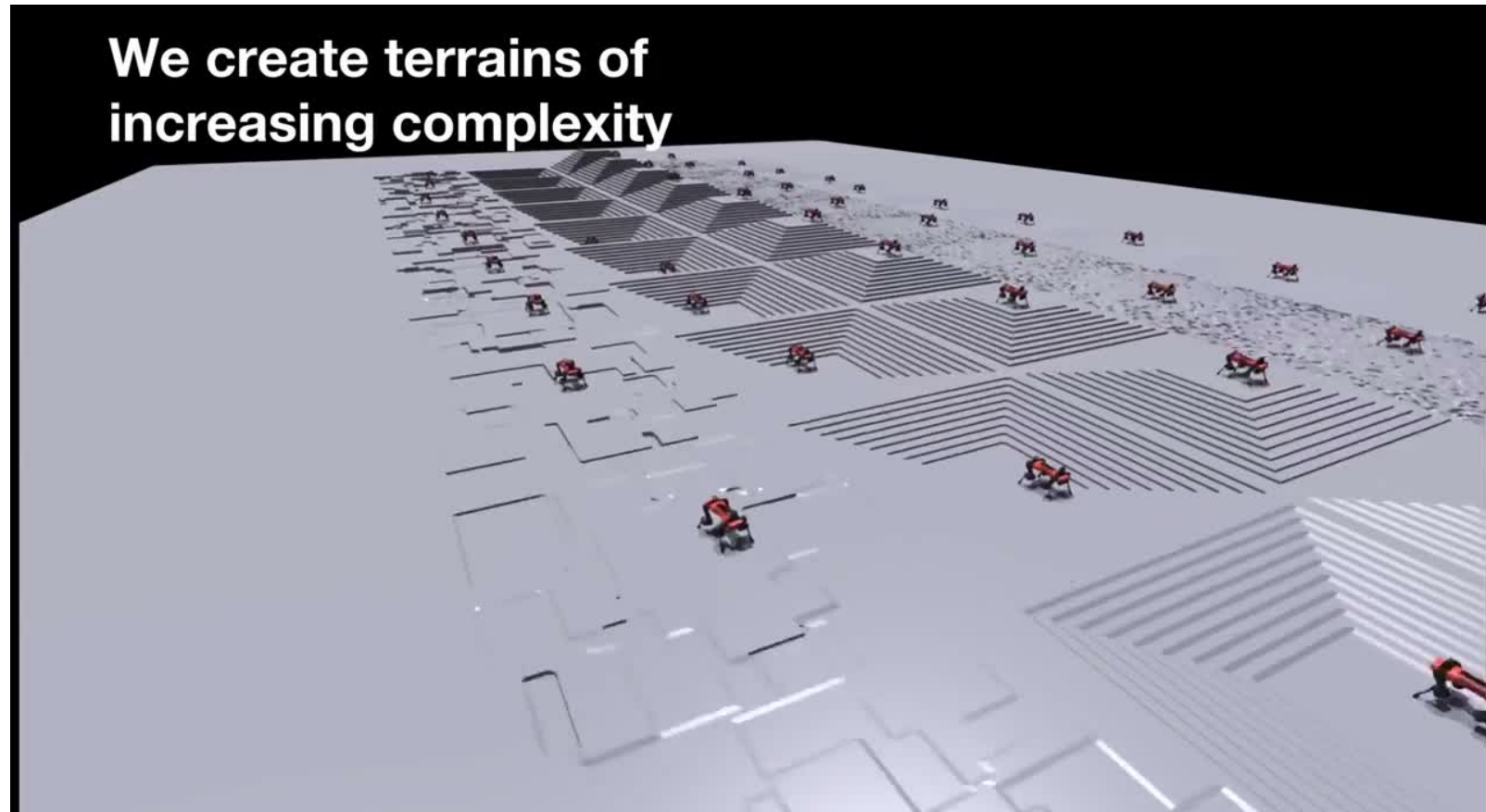
Idea: the privileged information cannot be got in the real-world.

Method: Training policies in simulation with privileged information and distill the policy to real-world inputs.



Curriculum Learning

Idea & Method: training RL policy from simple tasks to complex tasks motivates a stable training process.

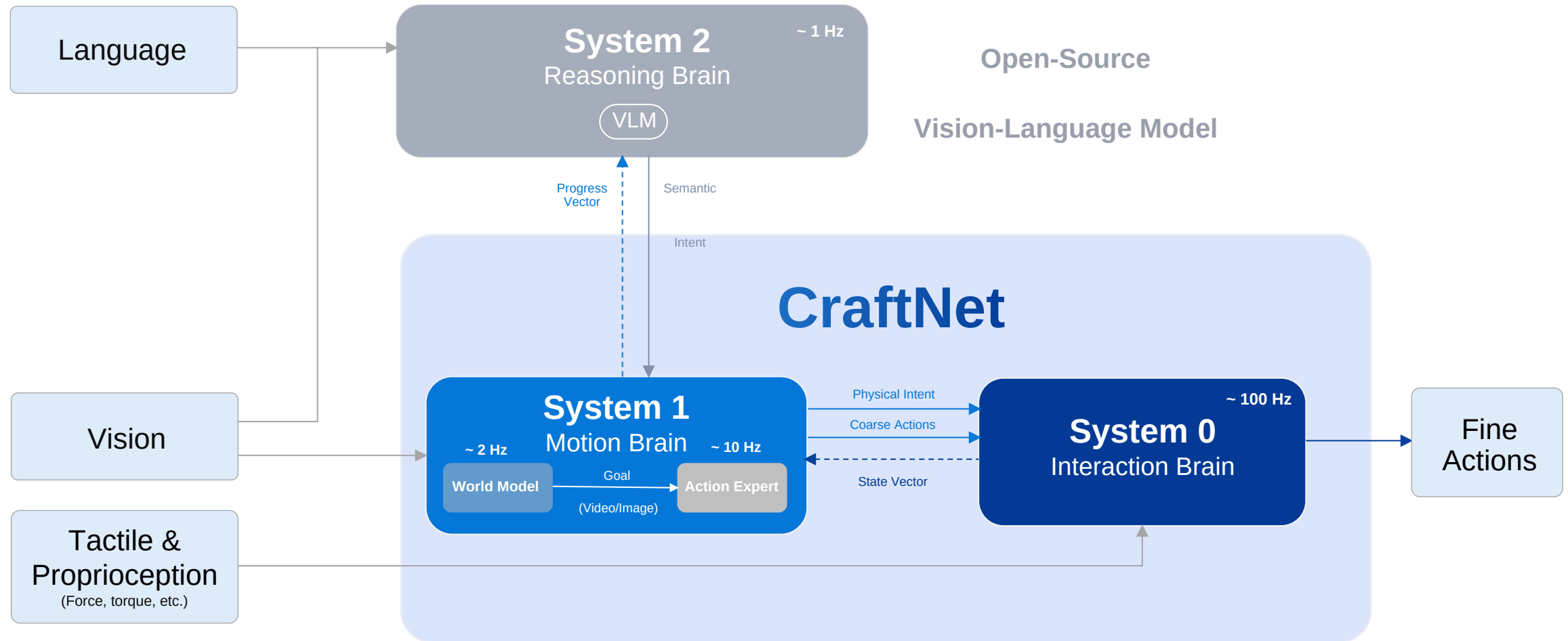


Frontiers: RL for Dexterous Manipulation

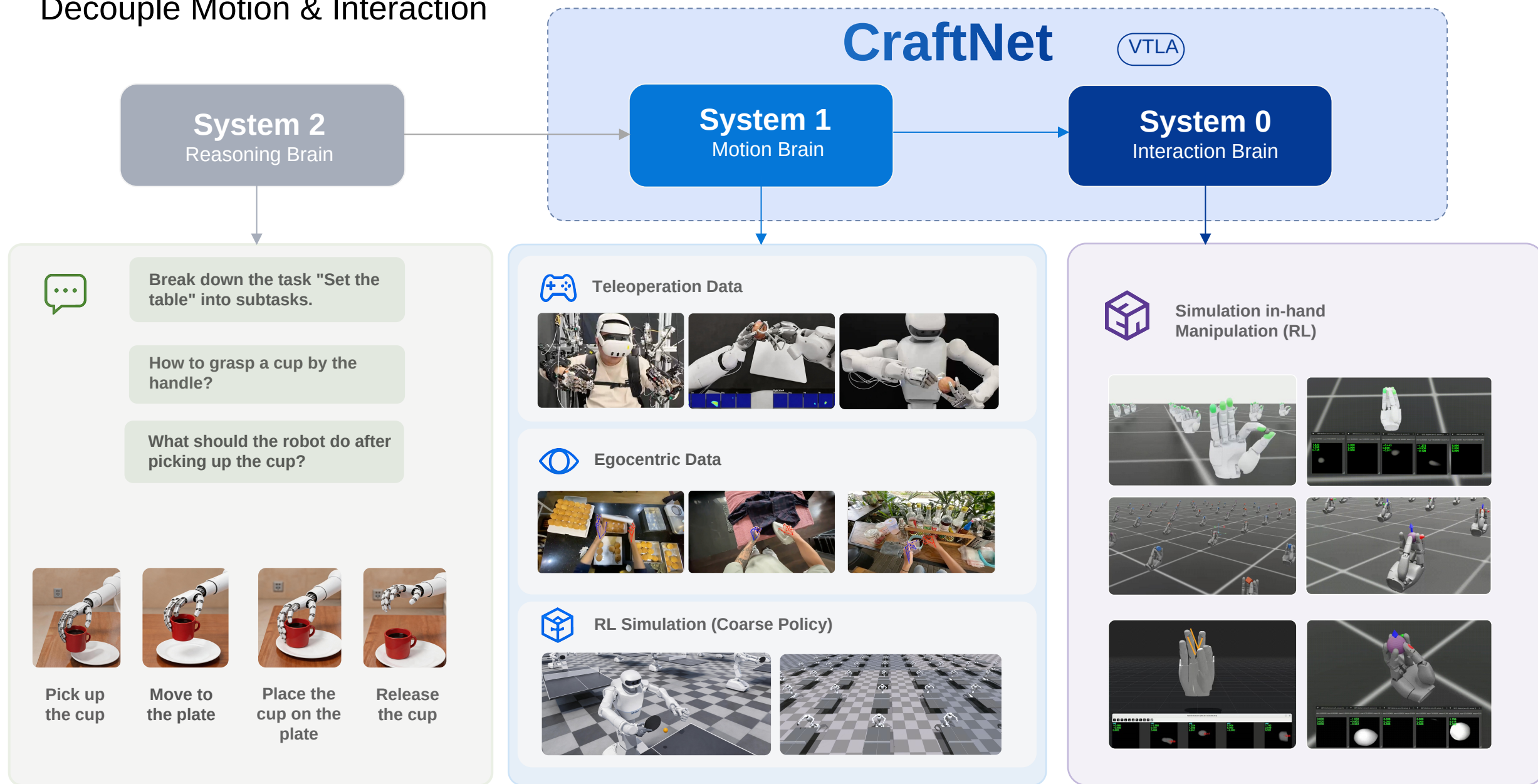
Teleoperation is Difficult during Interaction!

**Interaction needs a RL
Policy!**

Decouple Motion & Interaction



Decouple Motion & Interaction



Key problem:

What is the Interface for Motion & Interaction?

Turning Everyday Human Videos into Robot Manipulation Through Wave's Smooth Real-time Control

Paper Leads



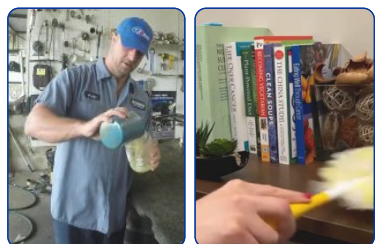
Jitendra Malik
Professor at UC Berkley



Pieter Abbeel
Professor at UC Berkley

and more
collaborators

Human Videos



In-the-wild RGB Videos

3D Reconstruction



Human Video Hand-Object Reconstruction

Recovering 3D hand-object interactions from monocular RGB videos for robot manipulation.

Retargeting



Visualizations of our retargeted hand-object interactions, physically simulated in MuJoCo.

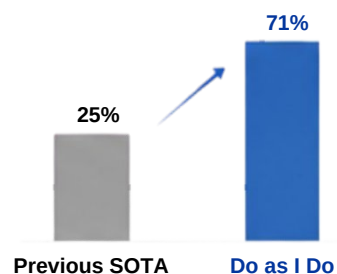
Real-world Validation: 10 Tasks on 22-DoF Wave



Introducing a pipeline that enables scalable robot data generation from **human demonstrations** and validating the approach on a **22-DoF Sharpa Wave platform**

Results

Retargeting Success Rate
(655 Internet Trajectories)



Enabling Complex Tool Use and In-hand Rotation by Minimizing the Sim2Real Gap

Paper Leads



C. Karen Liu[†]
Professor at
Stanford University



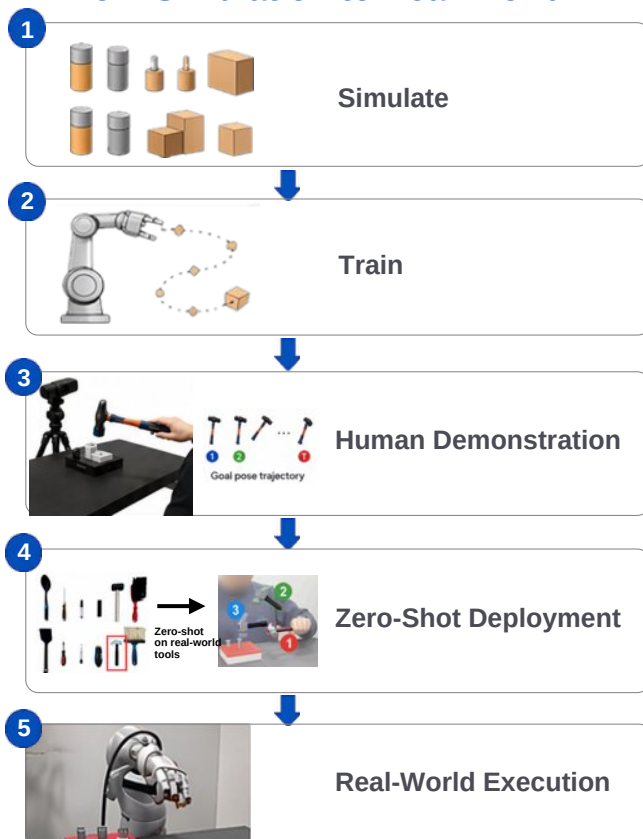
Jeannette Bohg[†]
Assistant Professor at
Stanford University

and more
collaborators

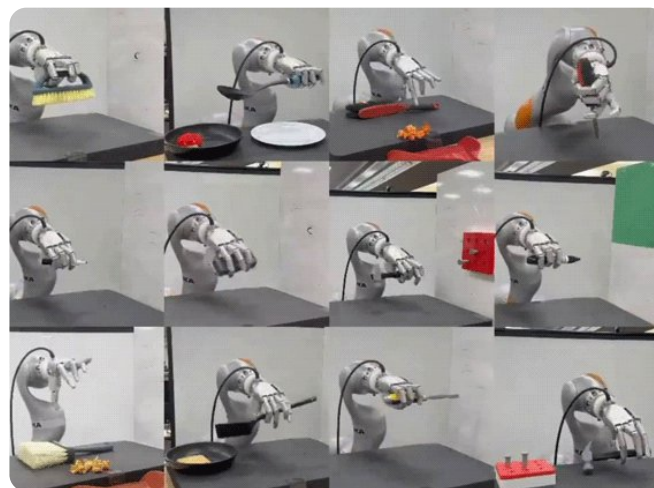
[†] Equal Advising



From Simulation to Real-World



Zero-Shot Success on Real Sharpa Hand



One policy.No fine-tuning.

Enabling **complex tool use, in-hand rotation and minimizing the sim-to-real gap** for direct real world deployment with high DoF and accurate URDF simulation model .

Proven Real-World Performance



37%

Outperforms prior
methods by **37%**



120+

Zero-shot rollouts
in the real world



6 Tool Categories

24 Diverse Tasks

Enabling Zero-Shot Sim2Real Precise Assembly Through Wave's High-DoF Dexterity

Paper
Leads

C. Karen Liu[†]
Professor at
Stanford University

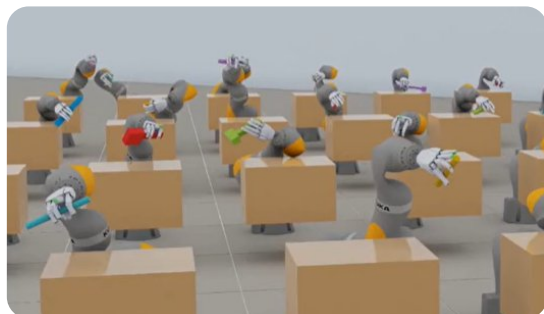


Jeannette Bohg[†]
Assistant Professor at
Stanford University

**and more
collaborators**

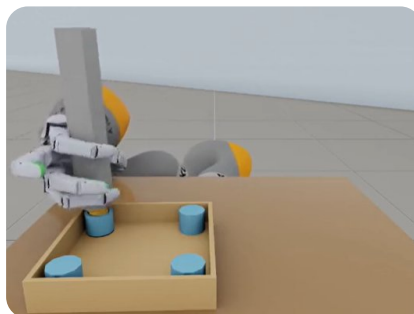
[†] Equal Advising

Play Pretraining



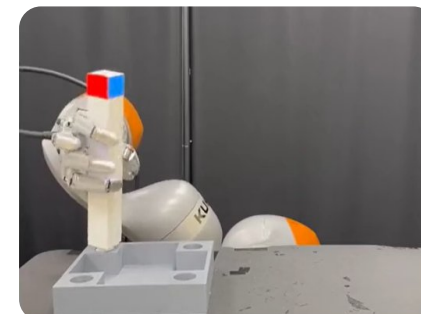
Learn Reusable Manipulation Priors

Assembly Finetuning



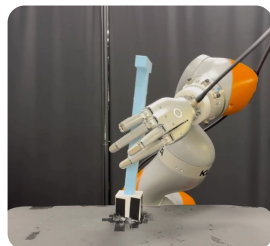
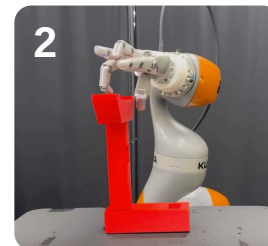
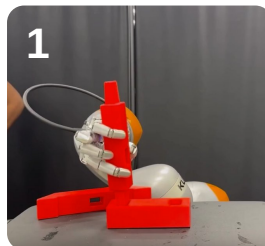
Adapt Play Prior to Assembly Tasks

Sim-to-Real Transfer



Precise Screwing Task

**More
Rollouts**

**Tight Insertion****Multi-Part Assembly**

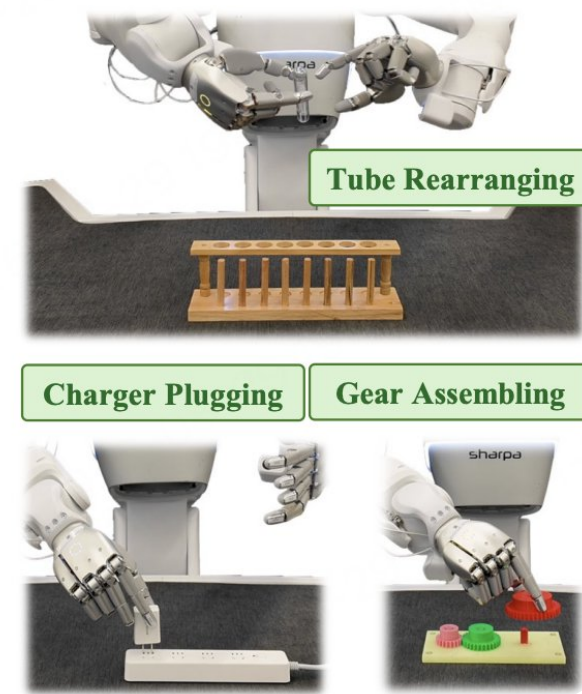
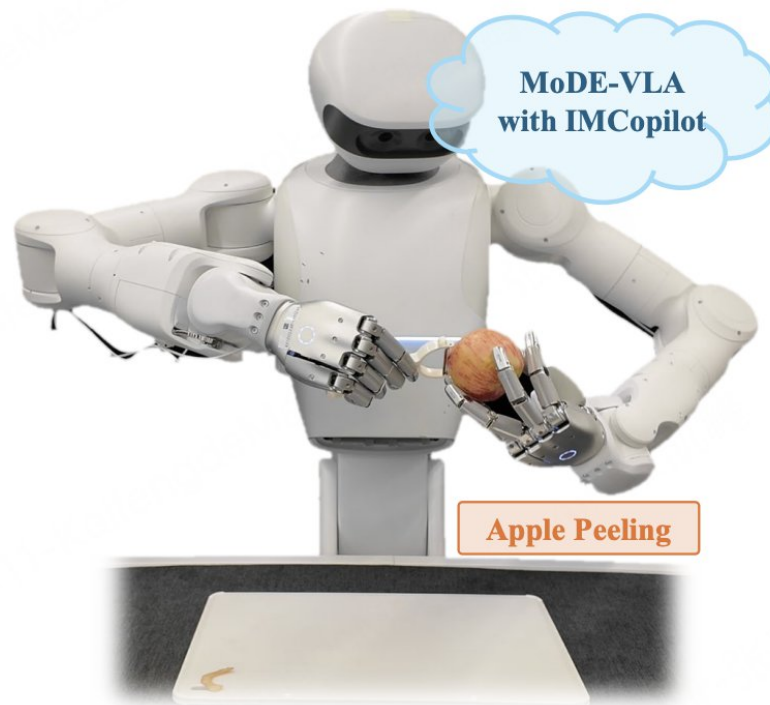
Play2Perfect enables fast RL for precise contact-rich assembly, leveraging Wave's **high-DoF dexterity and human-like manipulation capability**.

Decouple Motion & Interaction

Towards Human-Like Manipulation through RL-augmented Teleoperation
and Mixture-of-Dexterous-Experts VLA

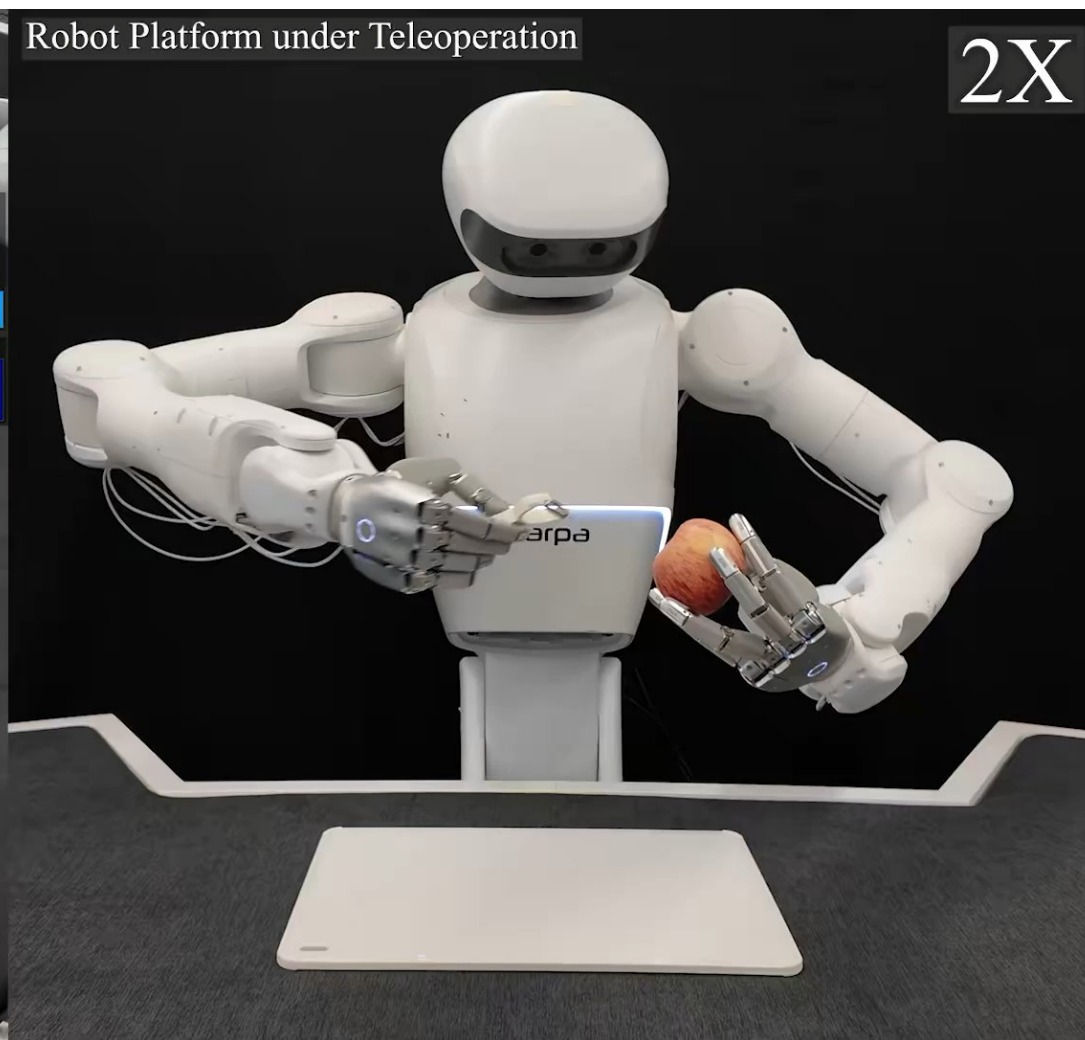
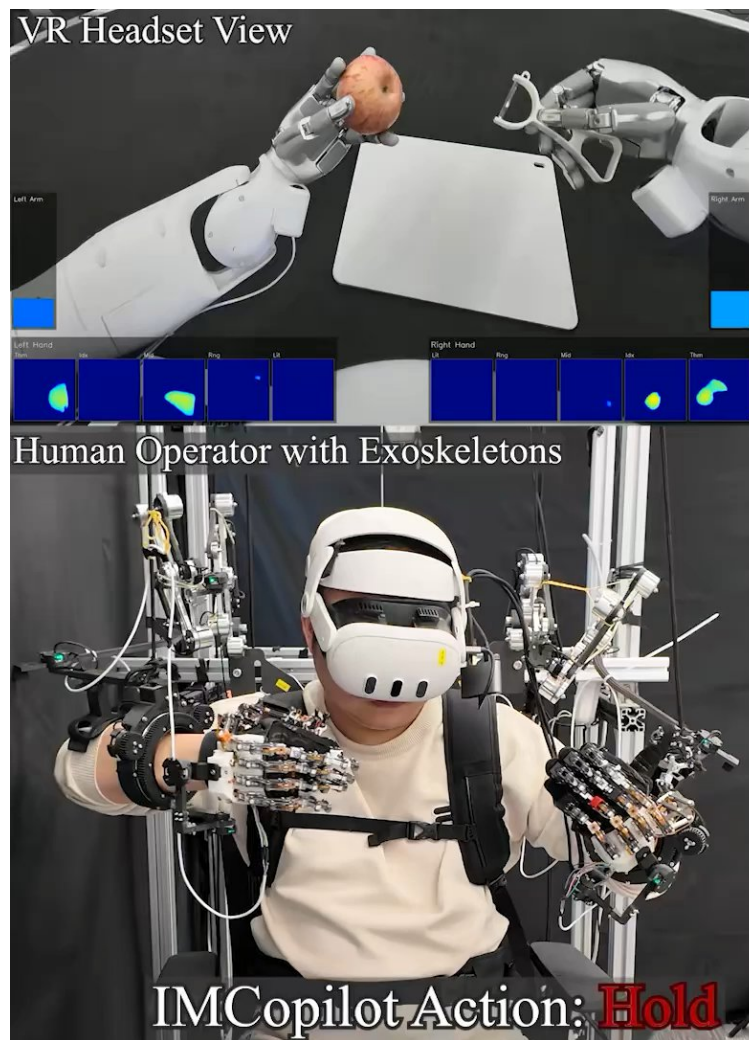
Interface: Pedal Trigger

Decouple Motion & Interaction

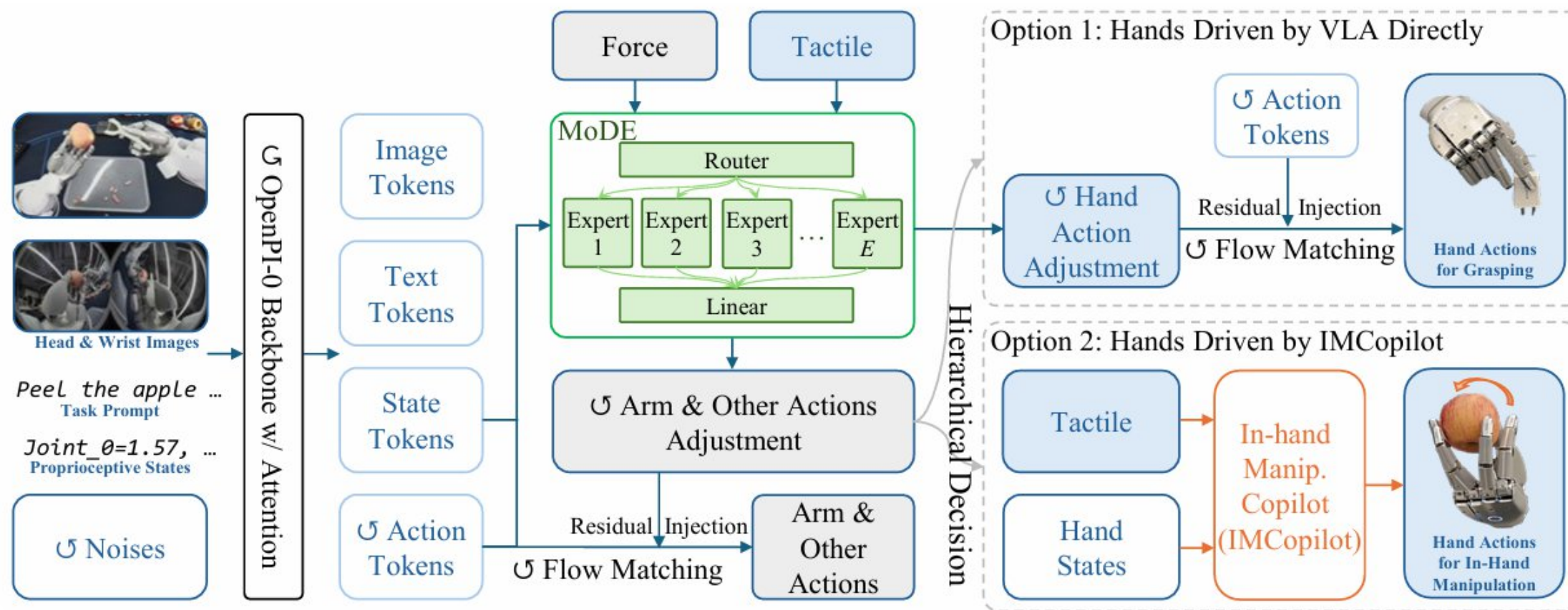


IMCoPilot takes the pedal trigger as the signal for in-hand manipulation primitive.

Decouple Motion & Interaction



Decouple Motion & Interaction



MoDE-VLA: Force/tactile experts add residual corrections to π_0 , while a hierarchical controller switches between VLA control and **IMCopilot**

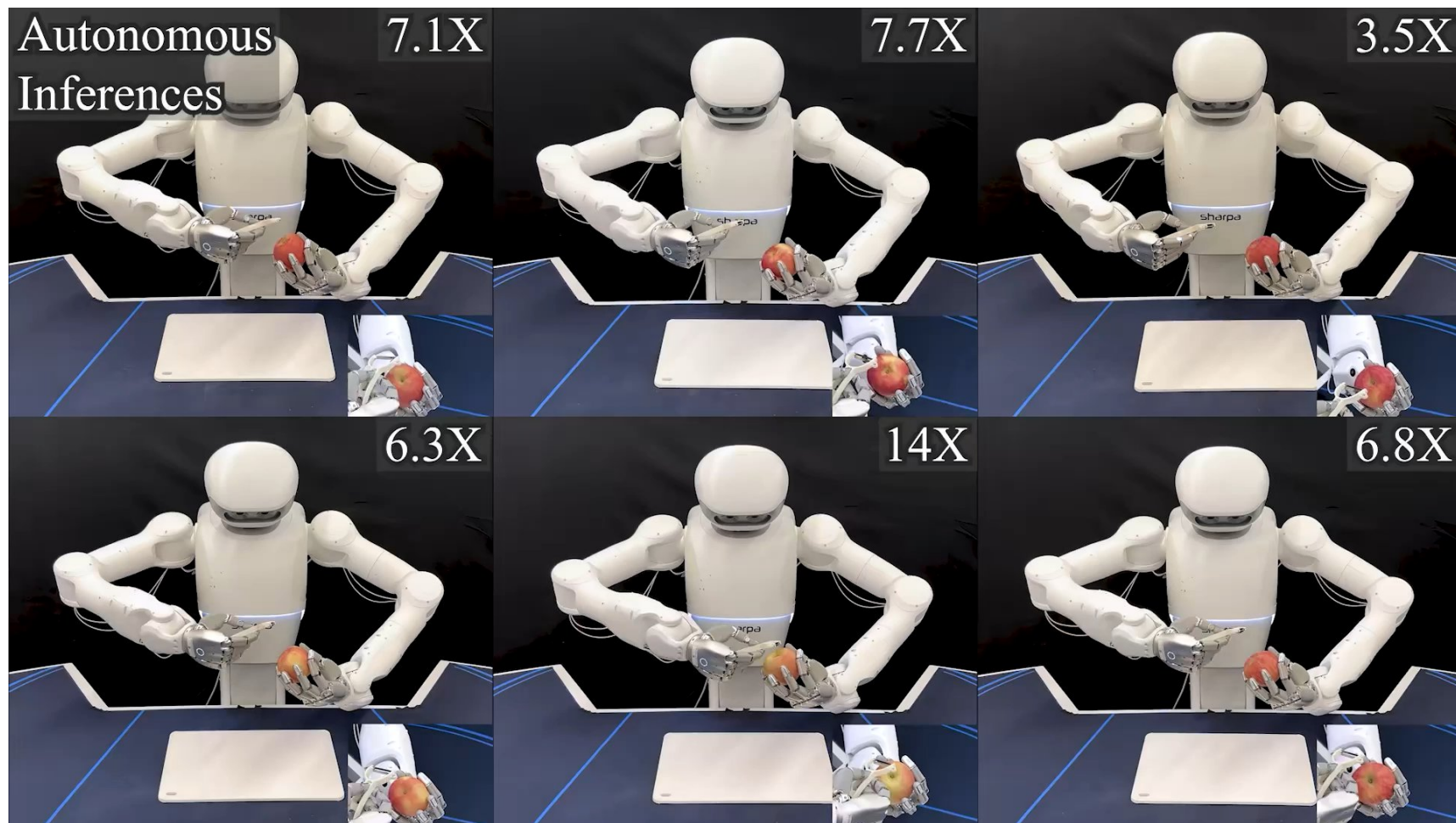
Decouple Motion & Interaction

Method	Apple Peeling		Tube Rearr.	Gear Assem.	Charger Plug.	Average
	SR $\uparrow$	PCR $\uparrow$	SR $\uparrow$	SR $\uparrow$	SR $\uparrow$	SR $\uparrow$
π_0 [3]	/	8%	15%	40%	5%	15%
Ours	30%	73%	30%	60%	15%	34%
<i>Ablations</i>						
w/o Force	15%	48%	25%	45%	5%	23%
w/o Tactile	30%	70%	15%	45%	15%	26%
w/o IMCopilot [†]	/	25%	–	–	–	–

/: Task failure (0% SR). –: Not applicable. [†]IMCopilot is only used in Apple Peeling.

Our **MoDE-VLA** makes use of force and tactile signals compared to ablation across all experimental tasks. **IMCopilot** motivates the complex apple peeling task data collection.

Decouple Motion & Interaction

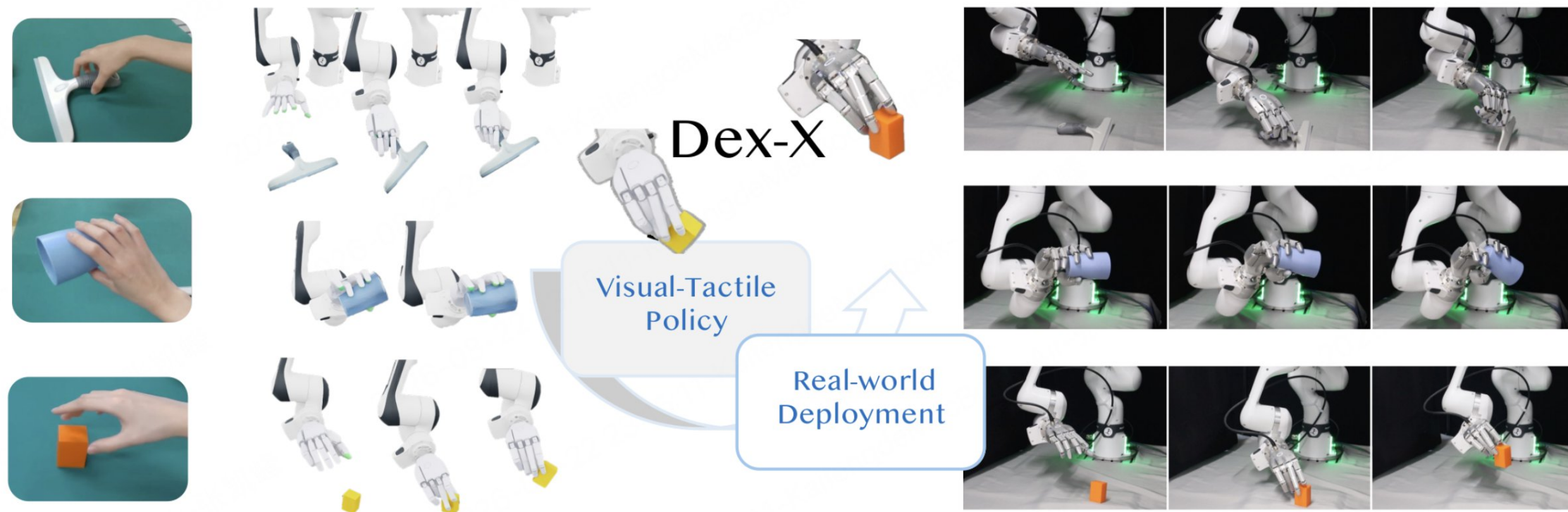


Decouple Motion & Interaction

Dex-x: Learning Visual-Tactile Dexterous Manipulation from Human Videos with Simulated Interaction

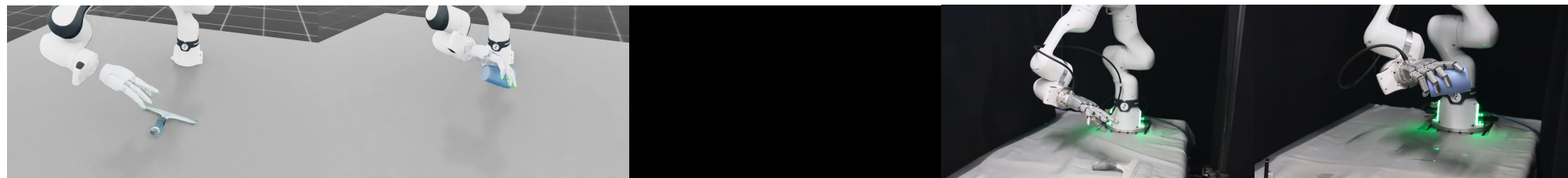
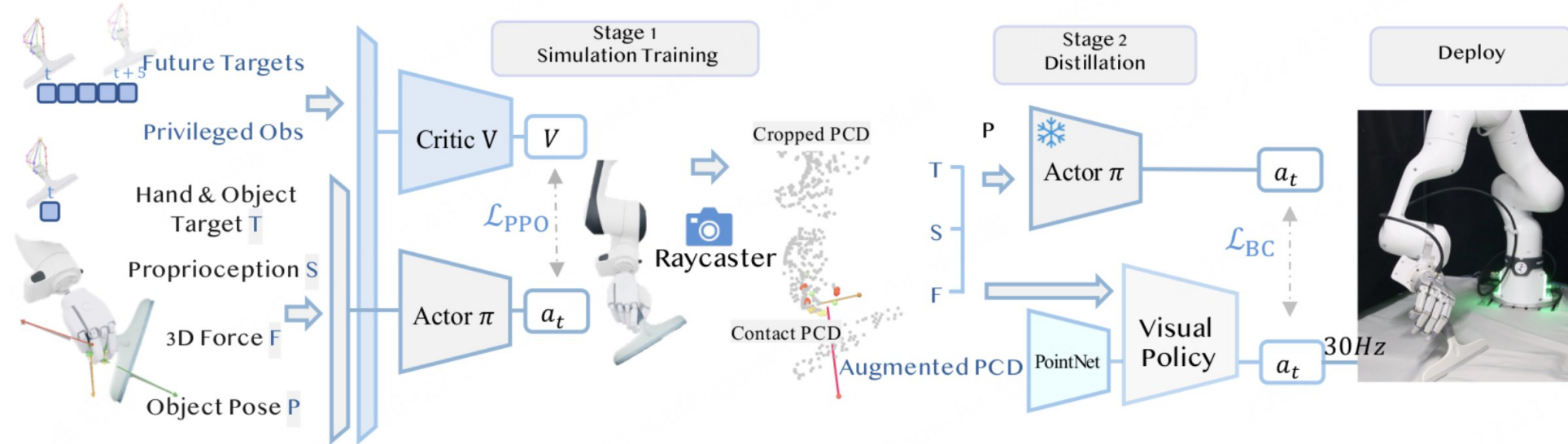
Interface: Point Cloud Depth

Decouple Motion & Interaction



Dex-X is a Top-Down Real2Sim2Real Pipeline.

Decouple Motion & Interaction



In the Real2Sim stage, RL policy would track the object and hand reference motion.

In the Sim2Real stage, the RL Policy would be distilled into a point cloud depth based student policy and realize zero-shot deployment.

Decouple Motion & Interaction

Category	DEX-X (ours)			DEX-X	ManipTrans	Kin. Retarget
	Reach	Grasp	Manip	Overall	Overall	Overall
Pick Up	99.2	97.4	89.9	89.6	1.5	1.8
Peg Insertion*	56.8	43.7	43.3	43.1	—	0.0
Tool Use	86.9	82.3	74.6	74.3	35.0	24.1
In-hand Rotation	—	—	36.8	36.8	56.9	0.0
In-hand Translation	—	—	64.8	64.8	11.5	0.0
Bimanual Handover	99.0	88.8	87.9	86.6	4.7	—
Average	85.5	78.0	66.2	65.9	21.9	5.2

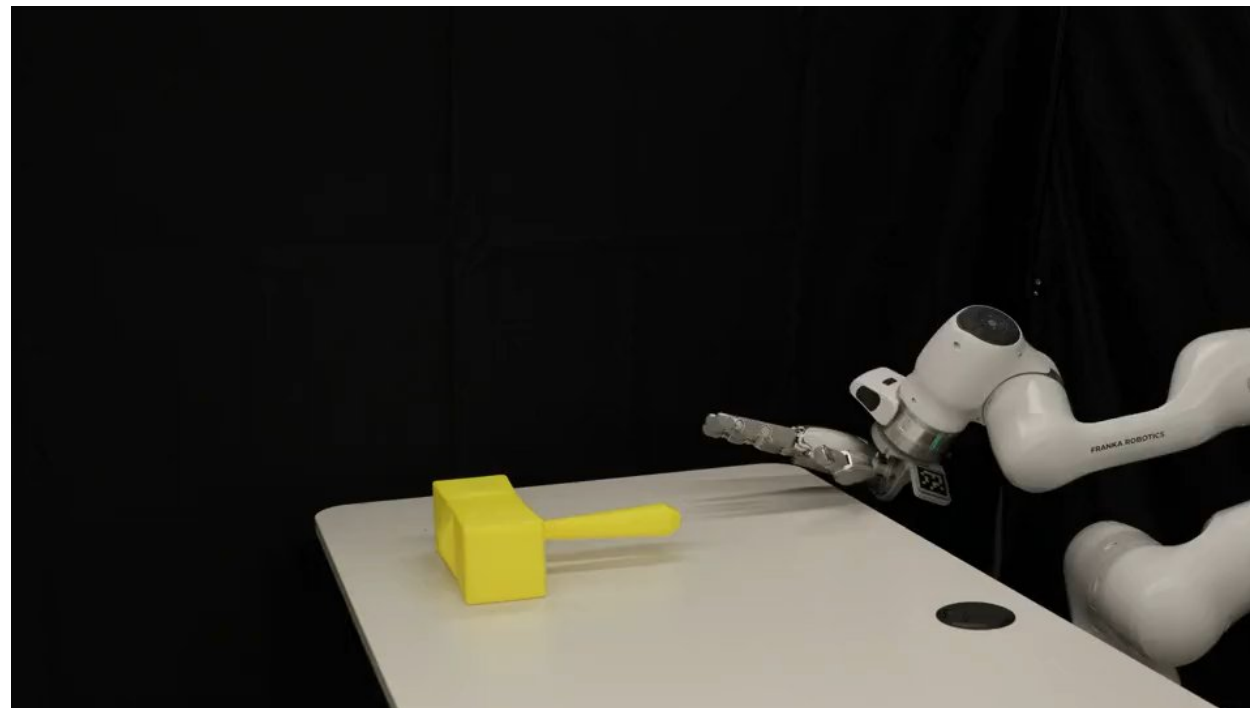
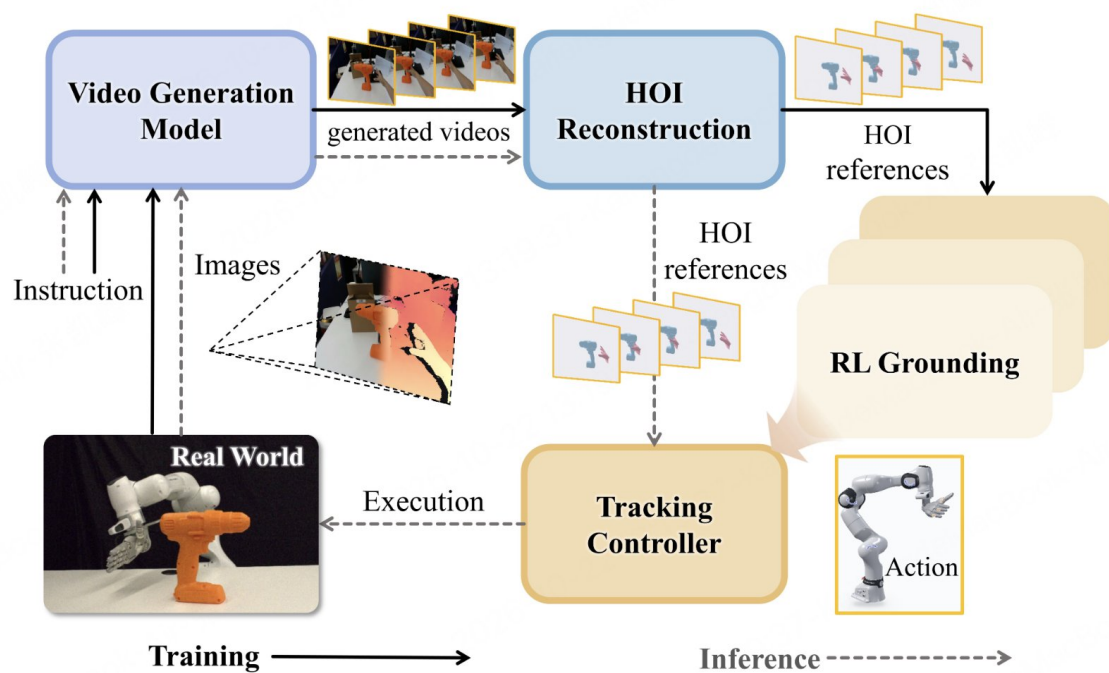
Per-category success rate (%) for Dex-X and other baselines
Dex-X outperforms Kinematic Retargeting and ManipTrans with high success rates.

Decouple Motion & Interaction

Grounding Generated Video Plans in Simulation Towards Versatile Dexterous Controllers

Interface: HOI Reference

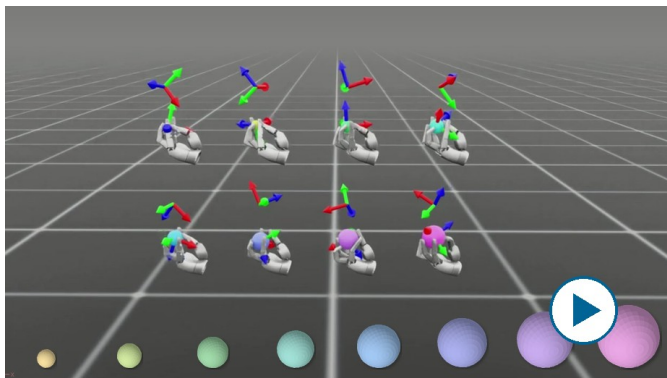
Decouple Motion & Interaction



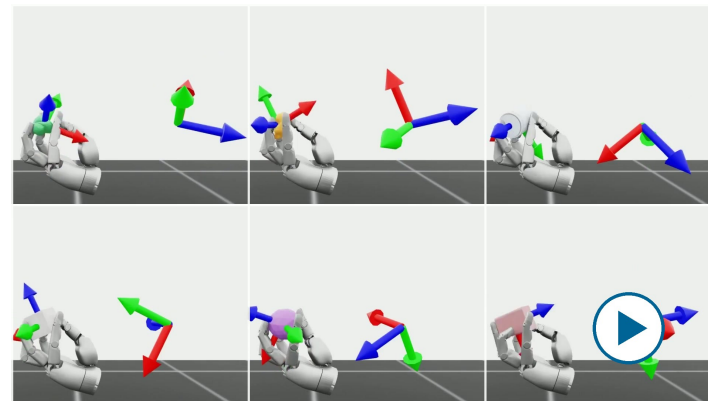
Decouple Motion & Interaction

TacBPM: A Tactile-Conditioned Behavior Prior Model for Dexterous Reorientation

Decouple Motion & Interaction



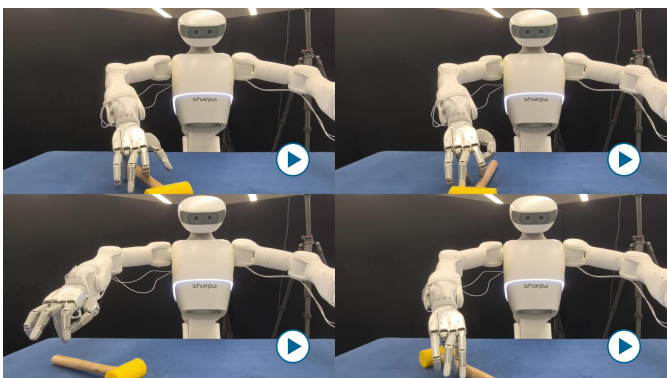
8 specialist across
different sphere scales



specialist-to-generalist distillation

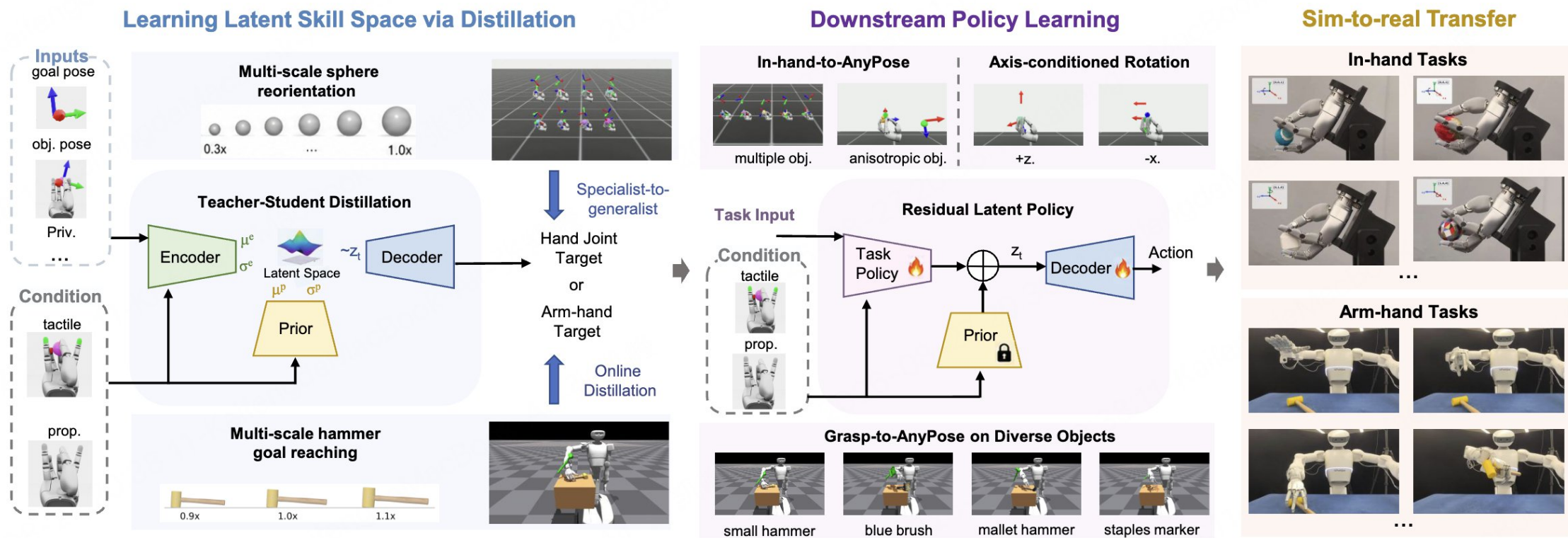


quickly adapt to
new object geometry



quickly adapt to
new embodiment

Decouple Motion & Interaction



Stage 1: TacBPM distills eight reorientation policy across eight different scales in a VAE-like neural network.
Stage 2: TacBPM quickly adapts to new downstream object geometries.
Stage 3: TacBPM quickly adapts to new tasks and embodiments with lower computation resources.

Decouple Motion & Interaction

Table I. Sphere-scale and unseen-object evaluation of the generalist prior ablations. Rows include seen sphere scales, unseen sphere scales, and object shapes excluded from teacher training.

Case	From scratch		No tactile		Ours	
	Succ. $\uparrow$	Steps † $\downarrow$	Succ. $\uparrow$	Steps † $\downarrow$	Succ. $\uparrow$	Steps † $\downarrow$
sphere (0.3)	24.50 $\pm$ 2.67%	79.40 $\pm$ 2.27	92.90 $\pm$ 1.59%	35.03 $\pm$ 1.59	97.30$\pm$1.01%	31.07$\pm$1.26
sphere (0.4)	25.20 $\pm$ 2.69%	79.81 $\pm$ 2.21	96.10 $\pm$ 1.20%	29.76 $\pm$ 1.16	98.40$\pm$0.78%	27.58$\pm$0.94
sphere (0.5)	99.80 $\pm$ 0.28%	23.28$\pm$0.30	100.00$\pm$0.00%	23.42 $\pm$ 0.18	99.90 $\pm$ 0.20%	23.45 $\pm$ 0.27
sphere (0.6)	20.40 $\pm$ 2.50%	84.18 $\pm$ 2.00	96.50 $\pm$ 1.14%	33.33 $\pm$ 1.26	98.90$\pm$0.65%	29.72$\pm$0.98
sphere (0.7)	14.60 $\pm$ 2.19%	89.27 $\pm$ 1.67	94.20 $\pm$ 1.45%	38.31 $\pm$ 1.43	97.00$\pm$1.06%	35.29$\pm$1.11
sphere (0.8)	10.20 $\pm$ 1.88%	92.95 $\pm$ 1.47	92.90 $\pm$ 1.59%	41.73 $\pm$ 1.36	96.80$\pm$1.09%	39.34$\pm$1.17
sphere (0.9)	8.40 $\pm$ 1.72%	93.95 $\pm$ 1.30	91.40 $\pm$ 1.74%	50.63 $\pm$ 1.50	92.00$\pm$1.68%	49.81$\pm$1.50
sphere (1.0)	5.10 $\pm$ 1.36%	96.11 $\pm$ 1.05	85.40$\pm$2.19%	61.13 $\pm$ 1.66	84.70 $\pm$ 2.23%	60.88$\pm$1.66
Seen Avg.	26.03 $\pm$ 21.29%	77.37 $\pm$ 21.21	93.67 $\pm$ 2.97%	36.67 $\pm$ 11.68	95.62$\pm$3.47%	34.64$\pm$11.72
sphere (1.1)	4.90 $\pm$ 1.34%	96.20 $\pm$ 1.05	55.50 $\pm$ 3.08%	85.10 $\pm$ 1.39	62.00$\pm$3.01%	83.10$\pm$1.49
sphere (1.2)	3.30 $\pm$ 1.11%	97.52$\pm$0.84	1.70 $\pm$ 0.80%	100.04 $\pm$ 0.44	13.30$\pm$2.11%	99.80 $\pm$ 0.98
block_01	18.30 $\pm$ 2.40%	85.26 $\pm$ 1.95	58.20 $\pm$ 3.06%	72.56 $\pm$ 2.07	79.50$\pm$2.50%	61.10$\pm$1.92
trash_can_01	11.70 $\pm$ 1.99%	90.62 $\pm$ 1.61	14.60 $\pm$ 2.19%	94.56 $\pm$ 1.25	79.52$\pm$1.05%	61.07$\pm$0.90
switch_01	13.80 $\pm$ 2.14%	89.30 $\pm$ 1.68	28.70 $\pm$ 2.81%	85.27 $\pm$ 1.72	47.10$\pm$3.10%	80.49$\pm$1.90
box_01	13.50 $\pm$ 2.12%	89.69 $\pm$ 1.65	29.00 $\pm$ 2.81%	90.24 $\pm$ 1.68	41.80$\pm$3.06%	88.24$\pm$1.99
bottle_03	11.80 $\pm$ 2.00%	90.55 $\pm$ 1.62	57.70 $\pm$ 3.06%	69.21 $\pm$ 1.91	77.80$\pm$2.58%	59.99$\pm$1.79
strawberry	14.50 $\pm$ 2.18%	88.40 $\pm$ 1.76	75.00 $\pm$ 2.69%	52.26 $\pm$ 1.87	82.00$\pm$2.38%	47.83$\pm$1.69
Unseen Avg.	11.47 $\pm$ 3.47%	90.94 $\pm$ 2.80	40.05 $\pm$ 17.51%	81.16 $\pm$ 10.81	60.38$\pm$17.06%	72.71$\pm$12.27

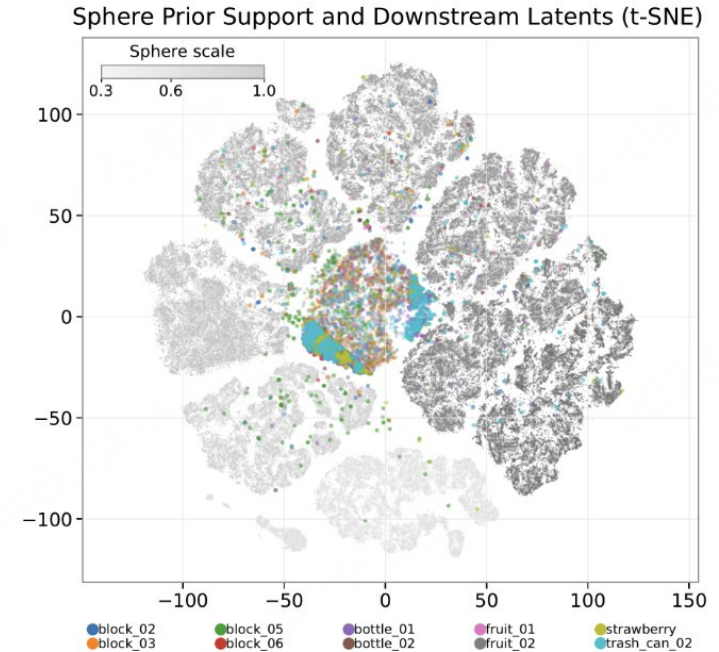


Fig. 3. Qualitative latent visualization. The t-SNE plot shows sphere-prior latent samples and executed downstream task latents.

TacBPM quickly adapts to new object geometries and t-SNE visualization shows that the 8 specialist latent (gray points) prior empowers new objects geometries (colored points).

Decouple Motion & Interaction

Table IV. **Real-robot commanded-axis rotation.** Entries report mean signed rotation and success for each commanded direction.

Object	Method	$+x$ Rot., Succ.	$-x$ Rot., Succ.	$+y$ Rot., Succ.	$-y$ Rot., Succ.	$+z$ Rot., Succ.	$-z$ Rot., Succ.
corner block	From scratch	428°, 6/10	162°, 3/10	27°, 0/10	81°, 1/10	293°, 4/10	113°, 2/10
	No tactile	59°, 1/10	22°, 0/10	31°, 0/10	13°, 0/10	257°, 4/10	30°, 0/10
	Ours	644°, 9/10	524°, 8/10	753°, 10/10	441°, 7/10	806°, 10/10	405°, 7/10
small tennis	From scratch	243°, 4/10	171°, 3/10	54°, 1/10	149°, 2/10	441°, 7/10	342°, 5/10
	No tactile	337°, 5/10	528°, 8/10	19°, 0/10	14°, 0/10	576°, 8/10	473°, 7/10
	Ours	765°, 10/10	519°, 8/10	383°, 6/10	338°, 5/10	945°, 10/10	394°, 6/10
standard tennis	From scratch	70°, 1/10	40°, 0/10	288°, 4/10	38°, 0/10	216°, 3/10	342°, 5/10
	No tactile	59°, 1/10	41°, 0/10	202°, 3/10	132°, 2/10	218°, 3/10	239°, 4/10
	Ours	378°, 6/10	192°, 3/10	495°, 7/10	137°, 2/10	456°, 7/10	633°, 9/10
multiface object (unseen)	From scratch	158°, 2/10	22°, 0/10	315°, 5/10	279°, 4/10	167°, 3/10	16°, 0/10
	No tactile	258°, 4/10	297°, 4/10	23°, 0/10	48°, 0/10	162°, 3/10	239°, 4/10
	Ours	675°, 9/10	459°, 7/10	797°, 10/10	324°, 5/10	754°, 10/10	799°, 10/10

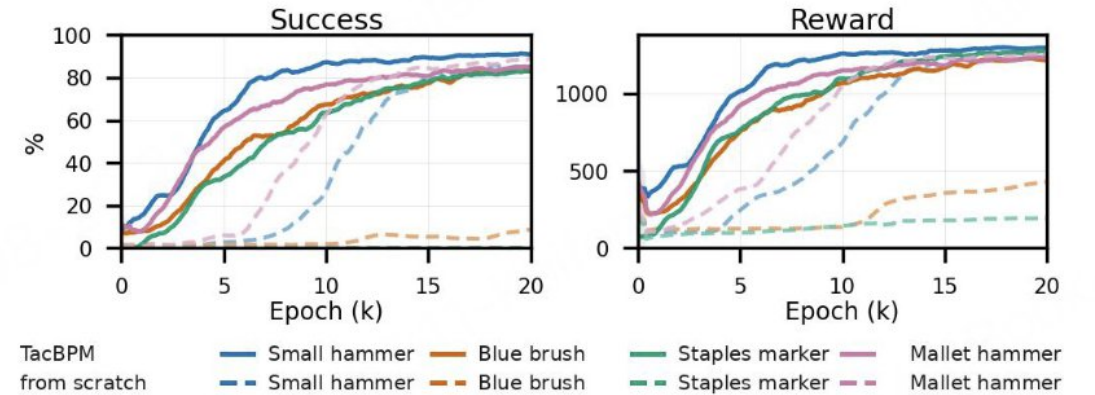


Fig. 8. **Arm-hand Grasp-to-AnyPose training curves.** The panels show success rate and episode reward across four novel tool geometries.

TacBPM quickly adapts to new object (zero-shot) and quickly converges on new tasks and embodiments.

sharp
We Manufacture Time **by Making Robots Useful**

