Penn Engineering
University of Pennsylvania

FA26

Special Seminar on
Multi-Modal AI

Jiayuan Mao

Department of CIS, GRASP

https://cis7000-fall26.thelabone.org/

Penn Engineering
University of Pennsylvania

FA26

Lecture 20:
Vision-Language-Action Model

Special Seminar on Multi-Modal AI

Jiayuan Mao
Department of CIS, GRASP
https://cis7000-fall26.thelabone.org/



Quote CR: Ivo Andrić

Vision-Language-Action Models

Inputs, training objectives and action outputs

VLA Inputs and Outputs

$$A_t \sim \pi_\theta(\cdot \mid o_t, \ell), \qquad A_t = (a_t, \ldots, a_{t+H-1})$$

Observation: camera images, with robot state or history when supported

Instruction: the task the user wants the robot to perform

Output: actions in a specified robot control interface

Reference: Kim et al. (2024), OpenVLA · Black et al. (2024), π0

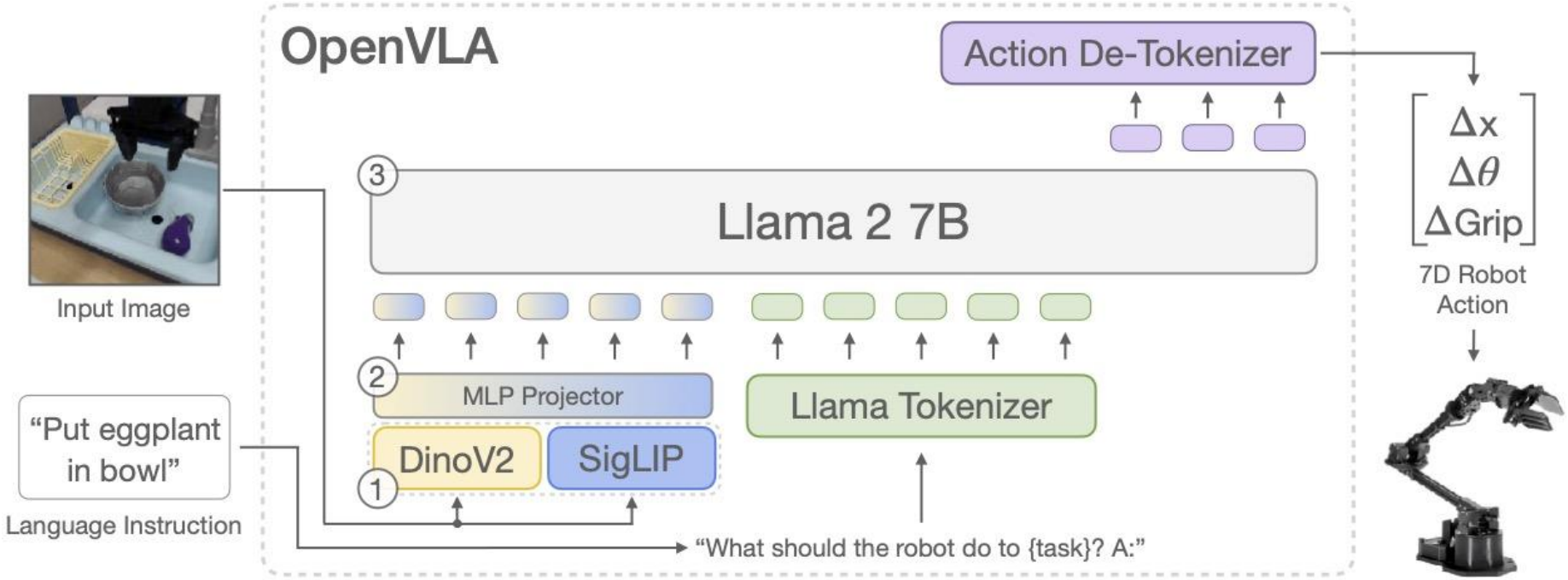
Training Stages

Stage	Data
Vision-language pretraining	Images and text
Robot policy training	Observations, instructions, actions
Target adaptation	Target robot demonstrations

“Transfer”: How VL capabilities transfer to action; how pretraining transfers to target adaptation

Reference: Kim et al. (2024), OpenVLA · Black et al. (2024), π0

OpenVLA: Model Architecture



Key idea: discretize 7D end-effector actions

Reference: Kim et al. (2024), OpenVLA

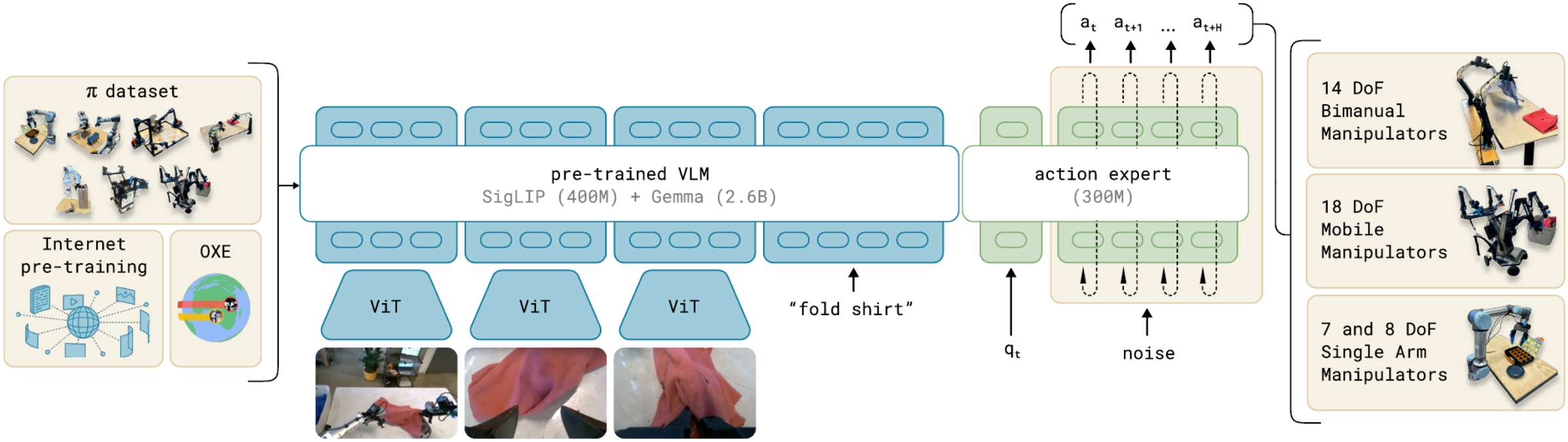
OpenVLA: Learning Objective

$$\mathcal{L}_{\text{act}} = -\mathbb{E}_{\mathcal{D}} \sum_{k=1}^{K} \log p_\theta(z_k \mid o, \ell, z_{<k})$$

Normalization: Normalize and discretize each action dimension

Reference: Kim et al. (2024), OpenVLA

π0: Continuous Action Prediction



Key idea: a pretrained VLM conditions a flow-matching action expert

Reference: Black et al. (2024), π0

π0: Flow Matching

$$A^\tau = \tau A + (1-\tau)\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I)$$

Train a vector field:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}[\| v_\theta(A^\tau, c, \tau) - (A - \epsilon) \|^2]$$

Reference: Black et al. (2024), π0

Summary: VLA Models

VLA policies condition robot actions on vision and language

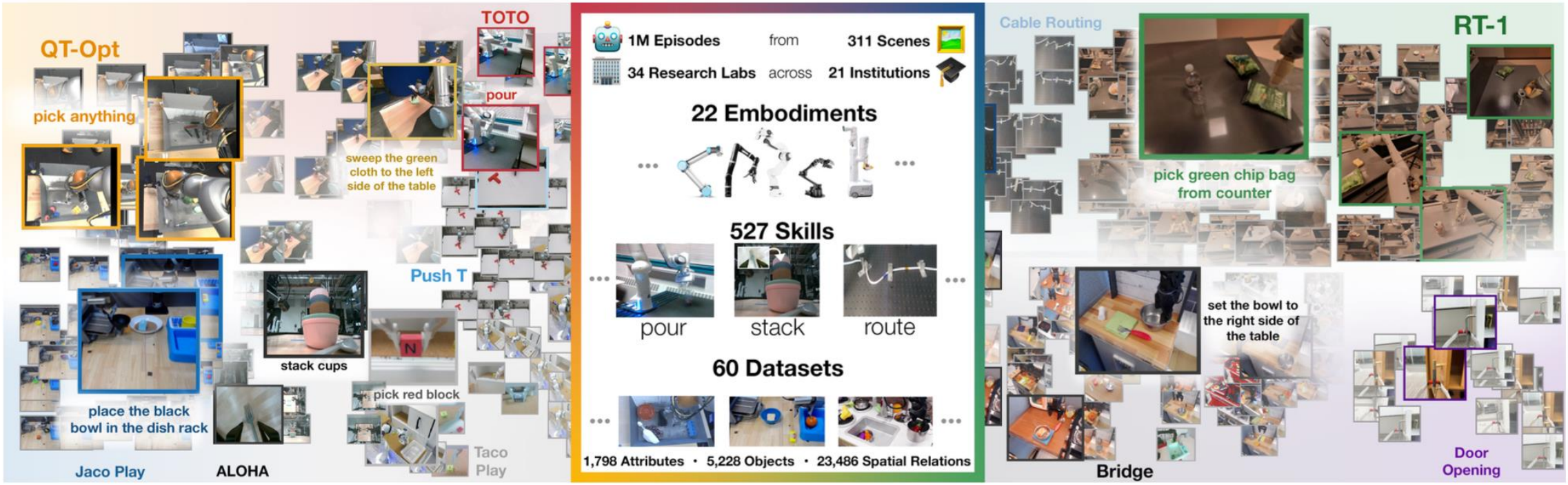
Pretrained vision-language features provide a starting point

Reference: Brohan et al. (2023), RT-2 · Kim et al. (2024), OpenVLA · Black et al. (2024), π0

Training Data

Coverage, action interfaces and language supervision

Open X-Embodiment: Shared Robot Data

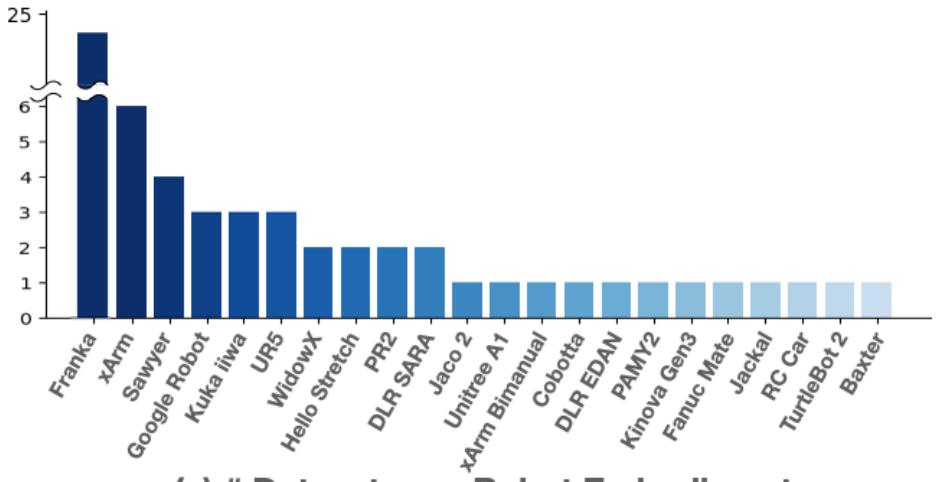


Different robots have different morphologies and actuators

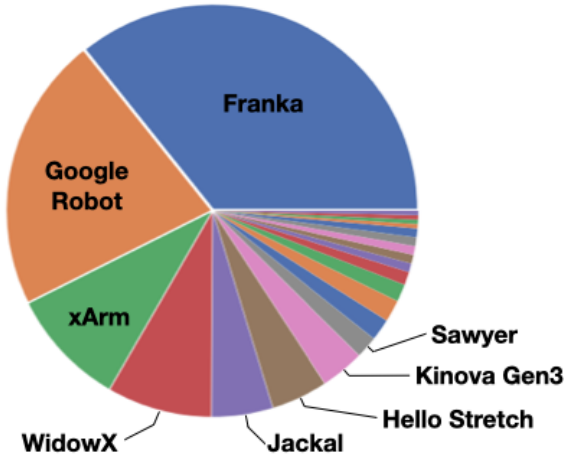
Standardize the data format before training a shared policy

Reference: Open X-Embodiment Collaboration (2024)

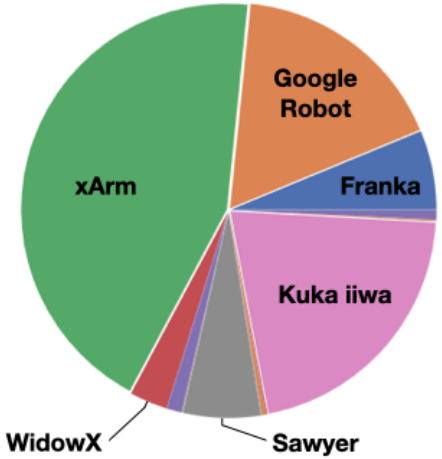
Dataset Size and Dataset Diversity



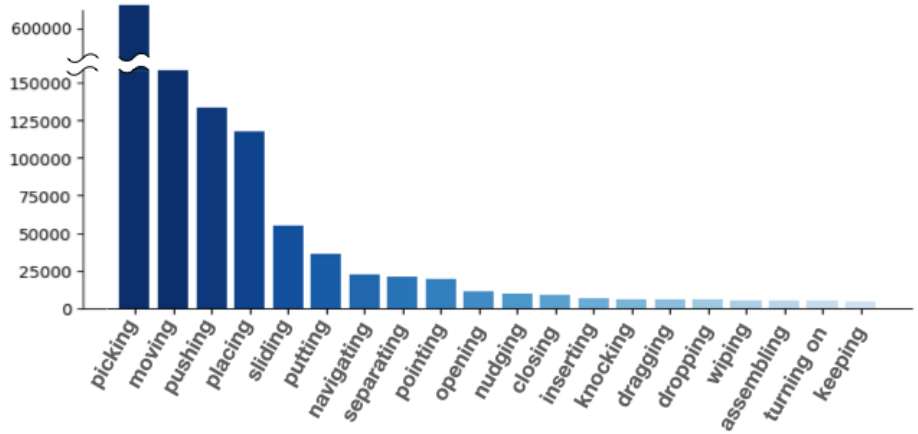
(a) # Datasets per Robot Embodiment



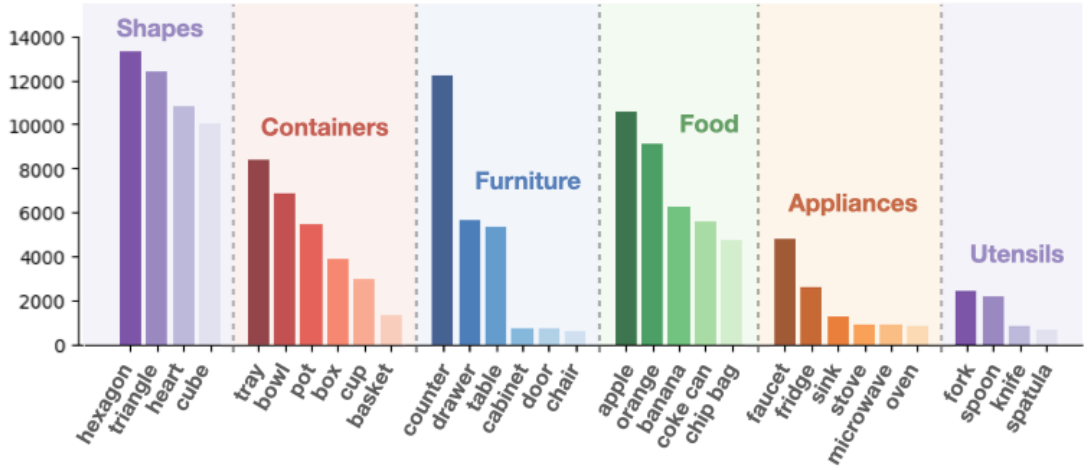
(b) # Scenes per Embodiment



(c) # Trajectories per Embodiment



(d) Common Dataset Skills



(e) Common Dataset Objects

Reference: Open X-Embodiment Collaboration (2024)

It is only a collection!!

Check
Absolute or delta?
World or body frame?
Gripper convention?
Trajectory frequency?

A common vector length is not a common physical meaning

Reference: Open X-Embodiment Collaboration (2024) · Kim et al. (2024), OpenVLA

GR00T N1: Training Data



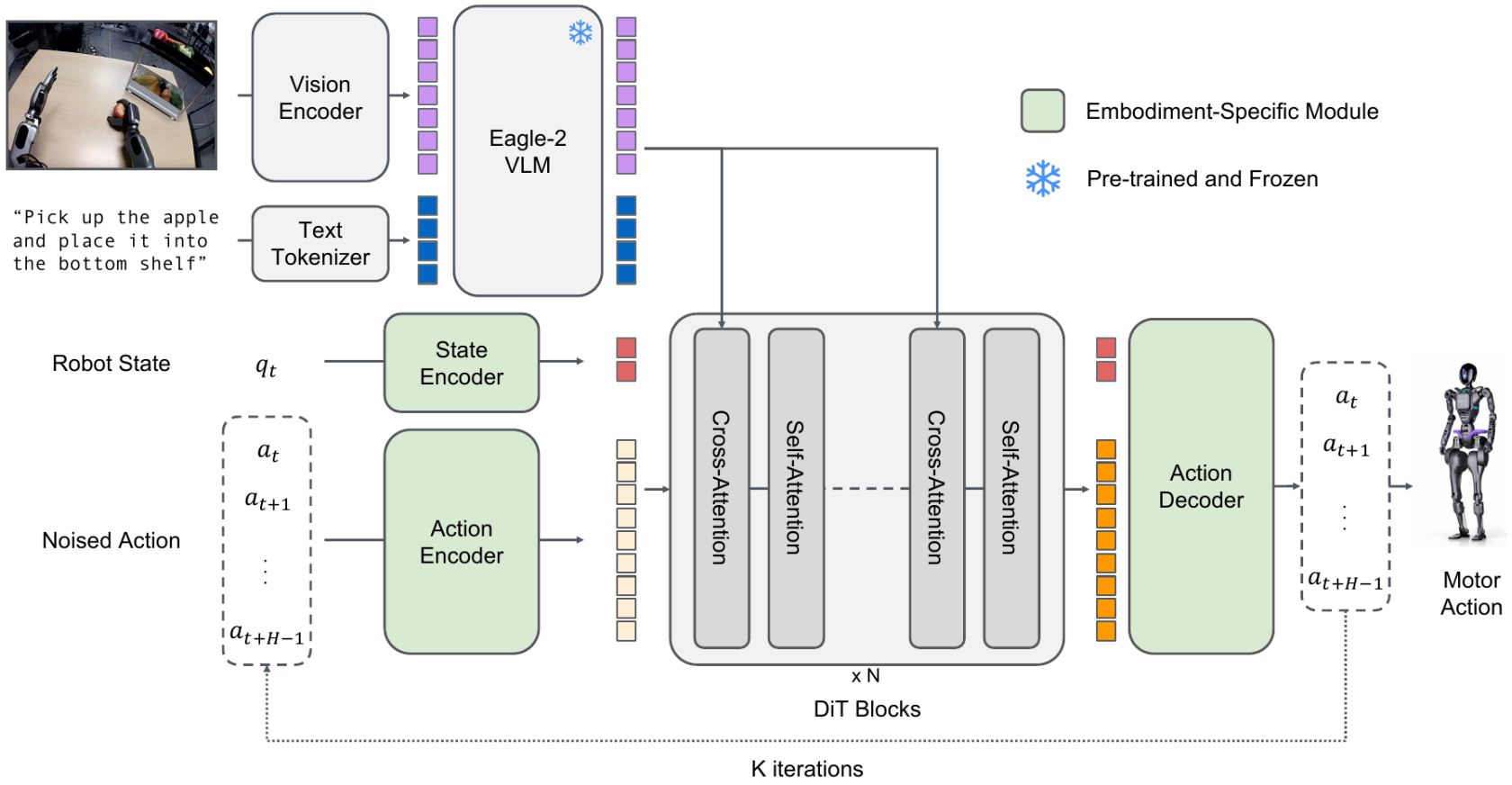
Robot demonstrations: real observations and commands

Synthetic trajectories: additional tasks and variations

Web and human videos: broad visual and motion experience

Reference: NVIDIA et al. (2025), GR00T N1 · Fig. 1

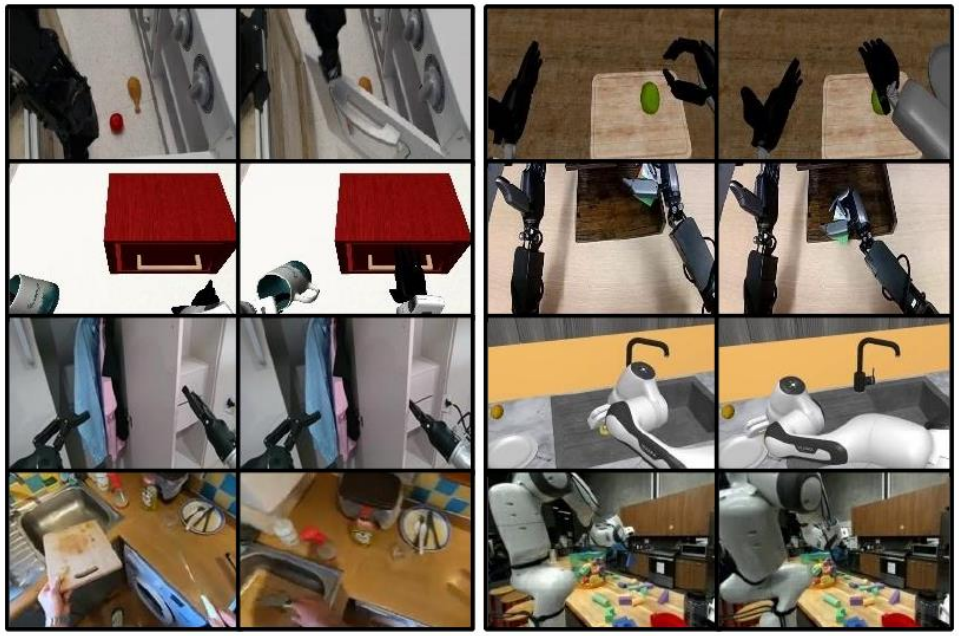
GR00T N1: Model Architecture



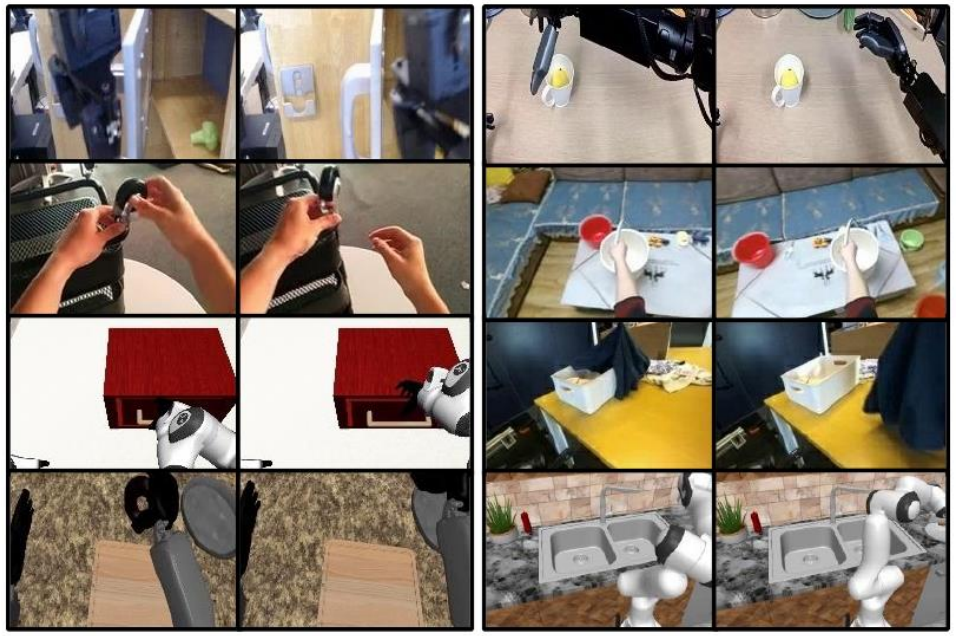
Shared VLM, with robot-specific state and action modules

Reference: NVIDIA et al. (2025), GR00T N1 · Fig. 3

Synthetic Training Data from “Latent Actions”



Latent action 1: Move the right arm to the left



Latent action 2: Move the right arm to the right

$$z = E(I_t, I_{t'}), \qquad \hat{I}_{t'} = D(I_t, z)$$

Training a VQ-VAE to represent the changes between two frames

Reference: NVIDIA et al. (2025), GR00T N1 · Fig. 4; Sec. 3

Synthetic Training Data with VideoGen

Prompt: use the right hand to pick up cucumber to basket



Prompt: use the left hand to pick up spray bottle to basket



Prompt: use the left hand to pick up spray bottle to beige bowl



Generate variations from a starting image and a task prompt

Filter videos, then infer pseudo-action labels

Reference: NVIDIA et al. (2025), GR00T N1 · Fig. 5, cropped; Sec. 3

Performances with Pretraining

Policy	RoboCasa	DexMimicGen	GR-1 Sim
BC Transformer	26.3%	53.9%	16.1%
Diffusion Policy	25.6%	56.1%	32.7%
GR00T N1	32.1%	66.5%	50.0%

Reference: NVIDIA et al. (2025), GR00T N1 · Table 2; Sec. 6

Summary: Training Data

Different robots still need additional “conditions” for the interface

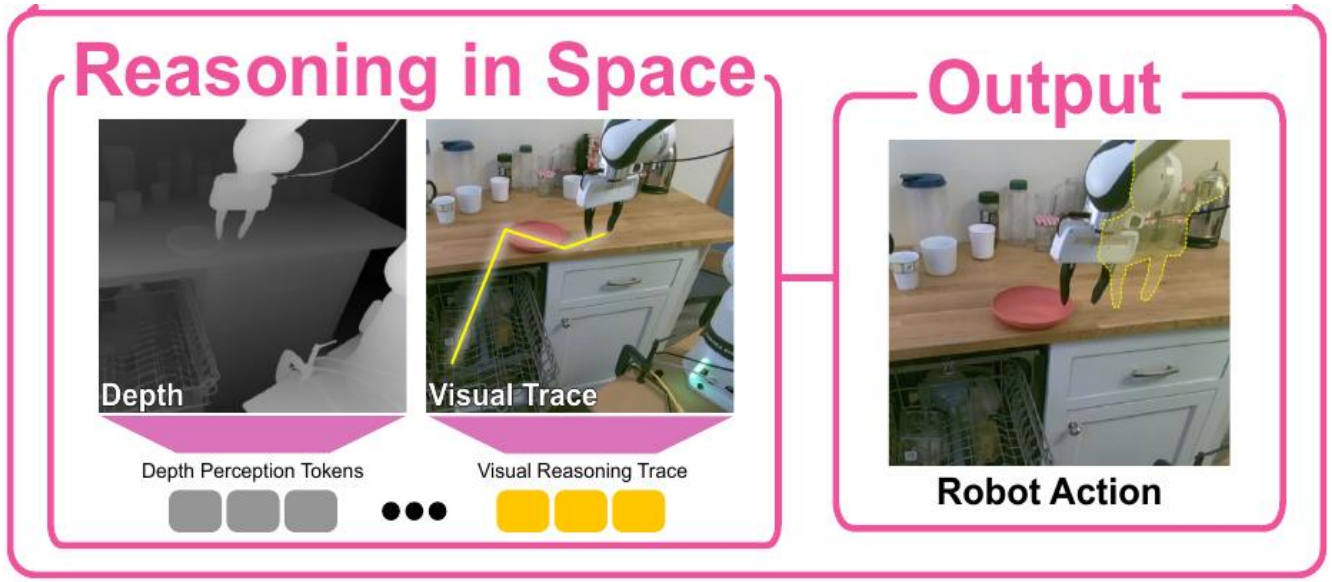
Additional data sources: video (latent action, motion tracking), video generation, simulation

Reference: NVIDIA et al. (2025), GR00T N1

VLM Transfer to VLA

Idea: Explicitly run VLMs before running the VLA

MolmoAct: Spatial Predictions Before Actions



$$p(d, \tau, a \mid I, \ell) = p(d \mid I, \ell)\, p(\tau \mid I, \ell, d)\, p(a \mid I, \ell, d, \tau)$$

Predict spatial information first before running the action branch

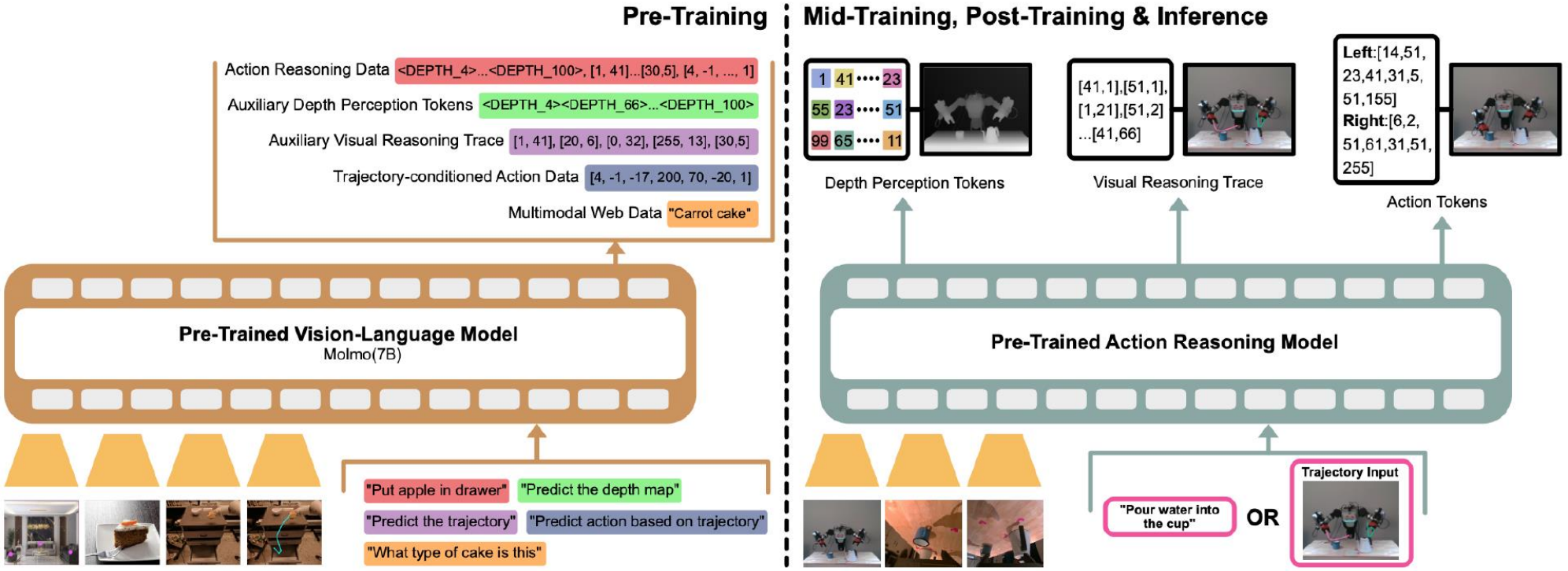
Reference: Lee et al. (2025), MolmoAct · Fig. 1, detail; Sec. 2.3

MolmoAct: Spatial Supervision

Prediction	**How the training label is constructed**
Depth tokens	RGB → depth specialist → learned quantizer
2D gripper trace	Point to the gripper in future frames (2D locations)
Action tokens	Discretized action tokens

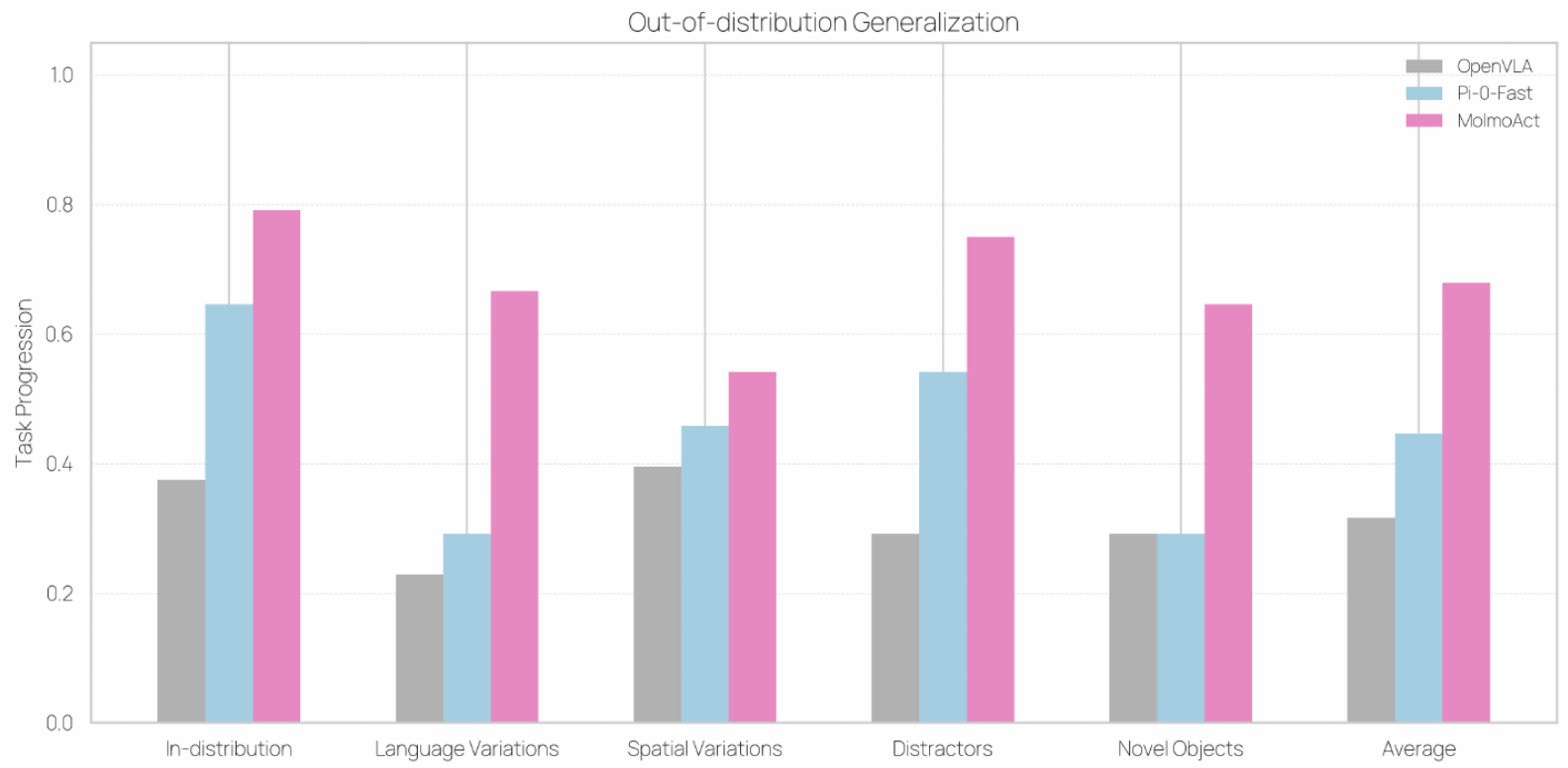
Reference: Lee et al. (2025), MolmoAct · Secs. 2.3, 3.1

MolmoAct: Training and Inference



Training includes depth, trace and trajectory-conditioned action examples

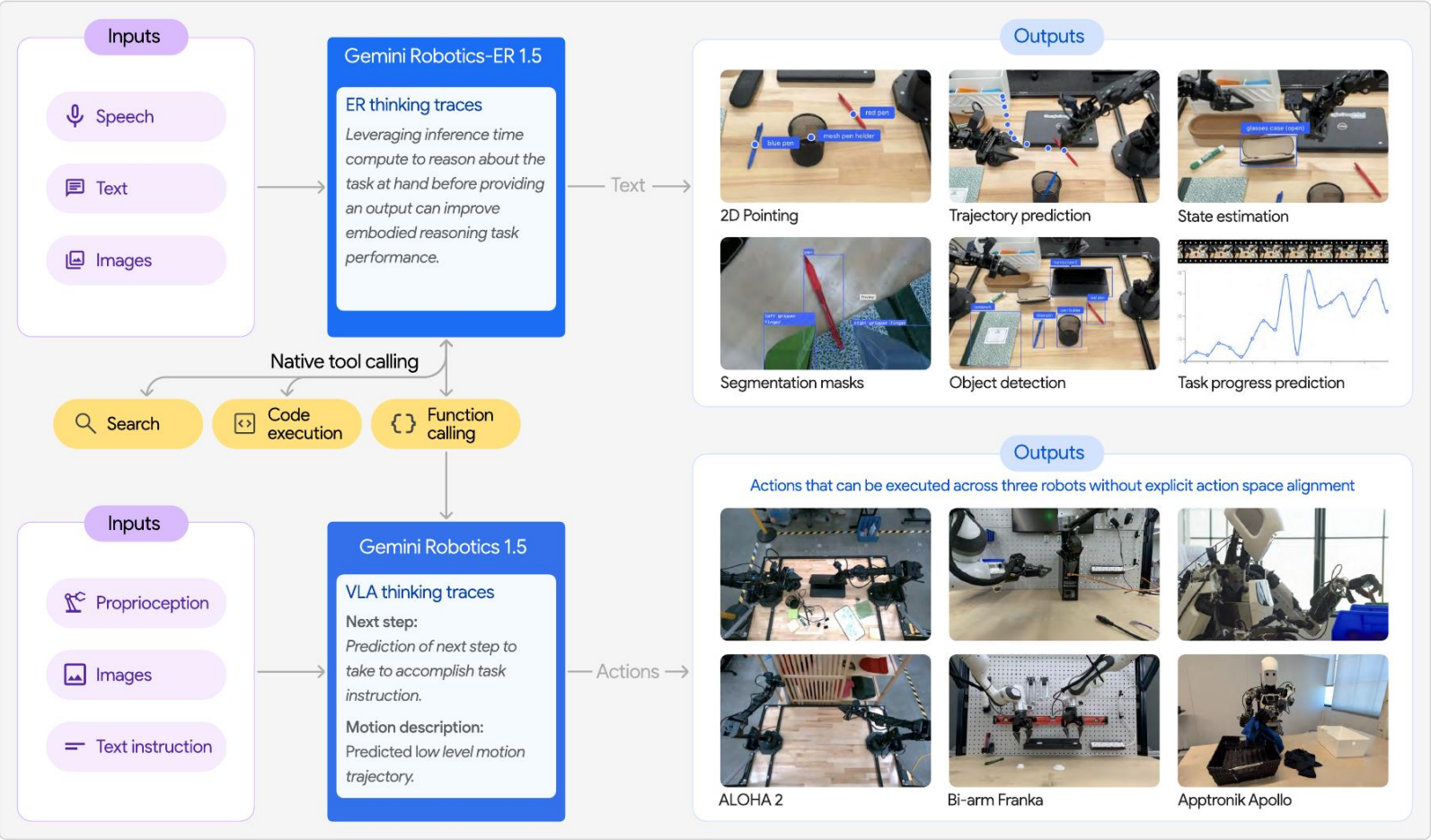
MolmoAct: Generalization Evidence



Metric: task progression after Franka task post-training

Reference: Lee et al. (2025), MolmoAct · Fig. 6(a); Sec. 5.3

Gemini Robotics 1.5: Two Models

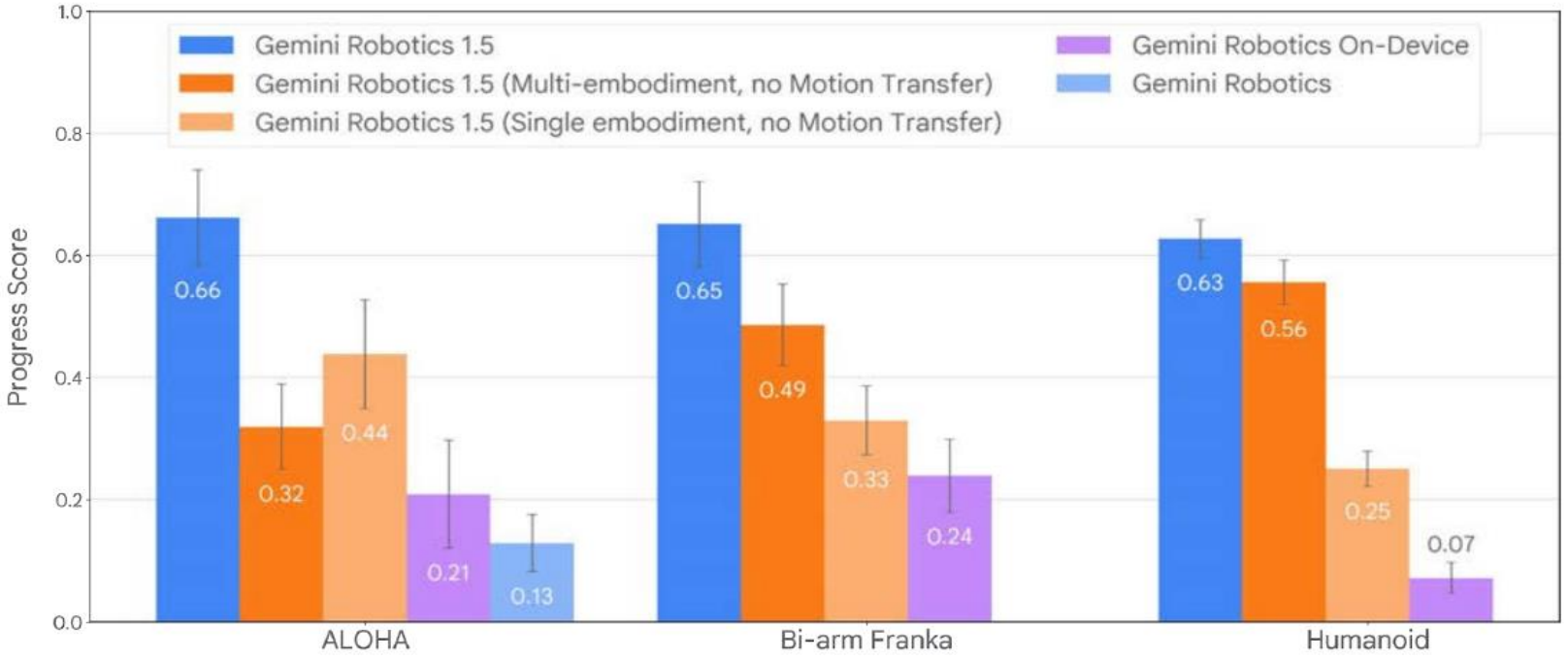


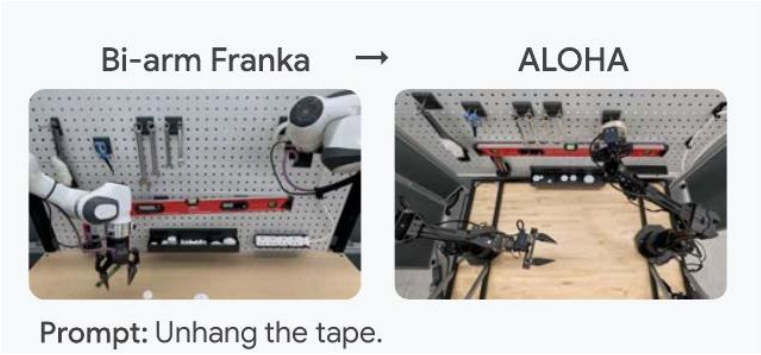
Robotics-ER 1.5: reason, plan and call tools

Robotics 1.5: generate robot actions

Reference: Gemini Robotics Team (2025), Gemini Robotics 1.5 · Fig. 1; Secs. 2–3

Gemini Robotics 1.5: Cross-Robot Transfer

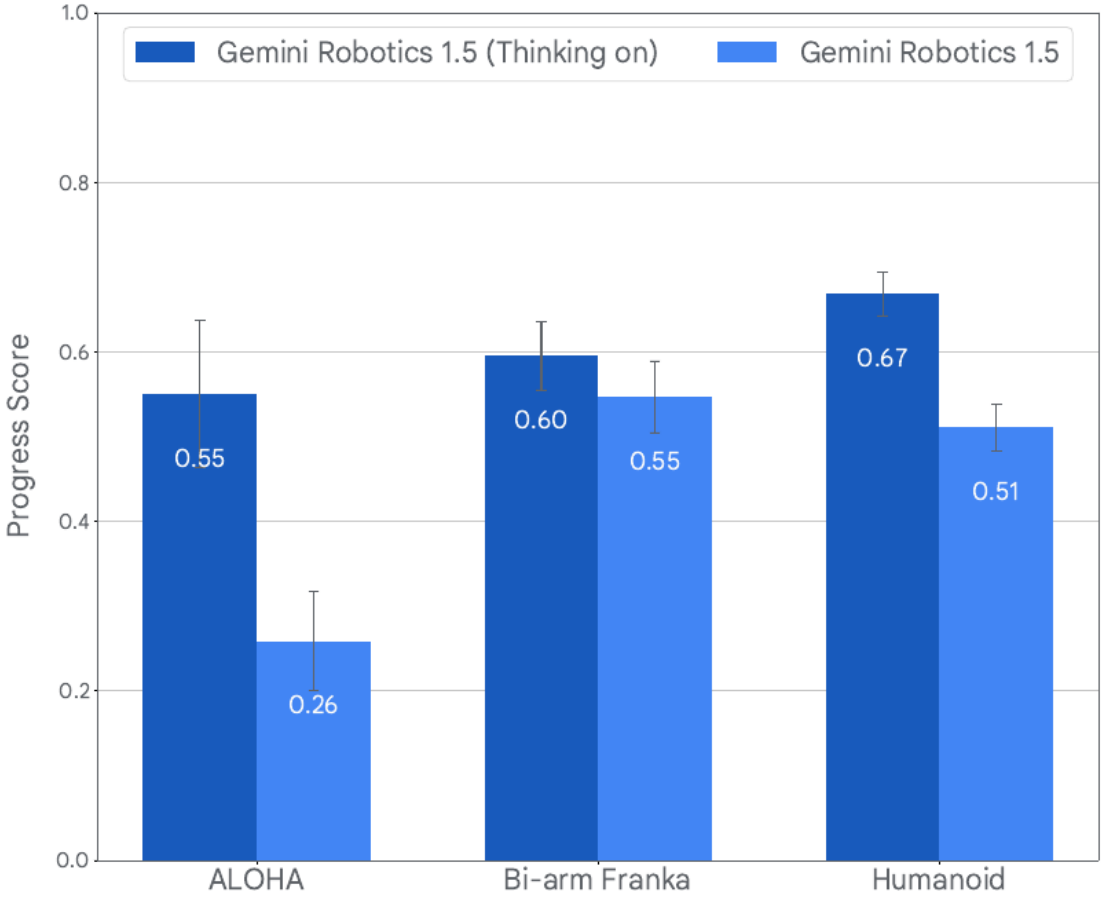




A task is trained on one robot and tested on another known robot

Reference: Gemini Robotics Team (2025), Gemini Robotics 1.5 · Fig. 5, details; Sec. 3.2

Gemini Robotics 1.5: Thinking Before Acting

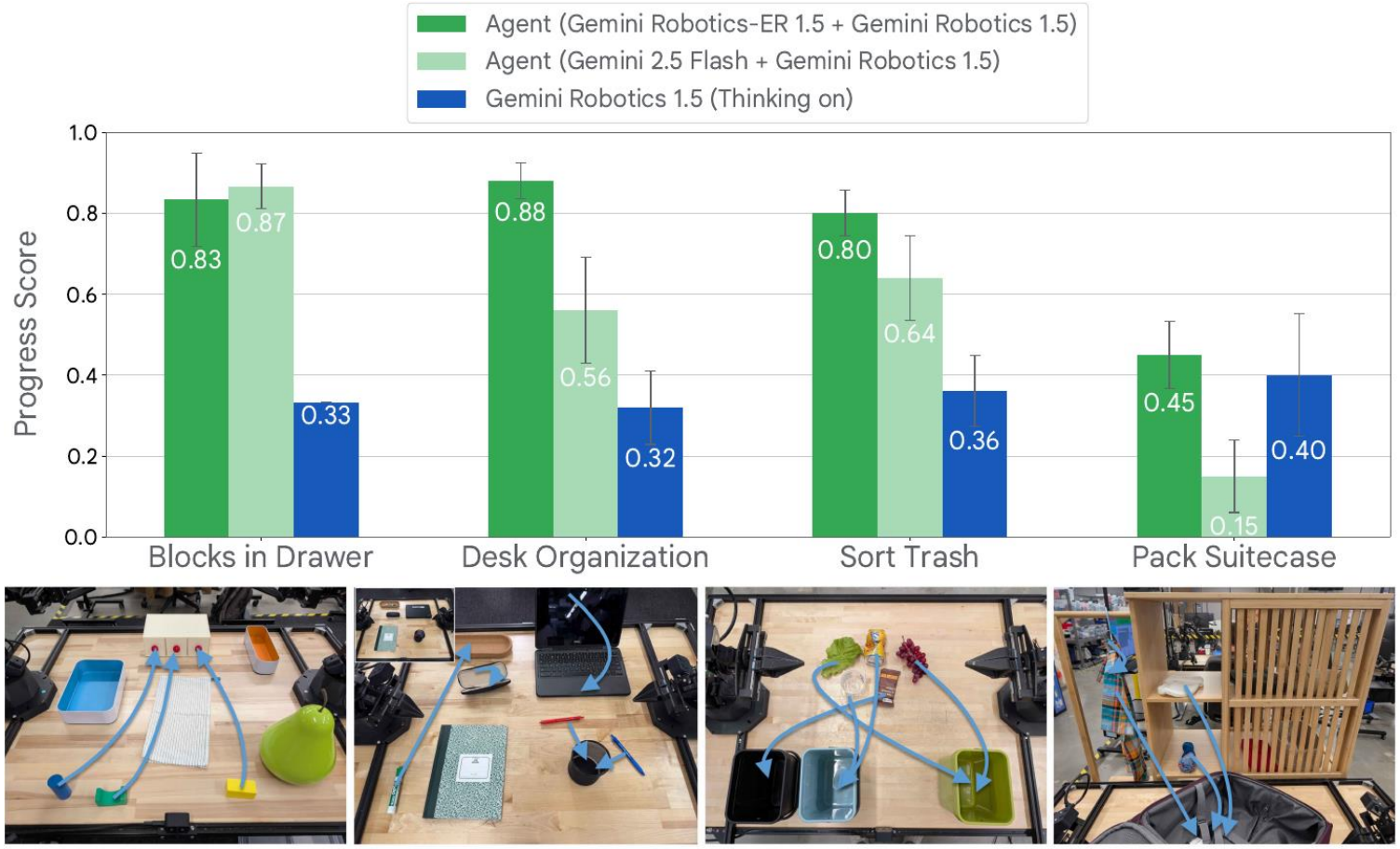


Predict a subtask or motion description before producing actions

Thinking improves progress on the tested multi-step tasks

Reference: Gemini Robotics Team (2025), Gemini Robotics 1.5 · Fig. 6; Sec. 3.3

Gemini Robotics 1.5: Long-Task Evidence



Compare a planner + VLA with the VLA alone

Reference: Gemini Robotics Team (2025), Gemini Robotics 1.5 · Fig. 17, ALOHA subset; Sec. 5.2

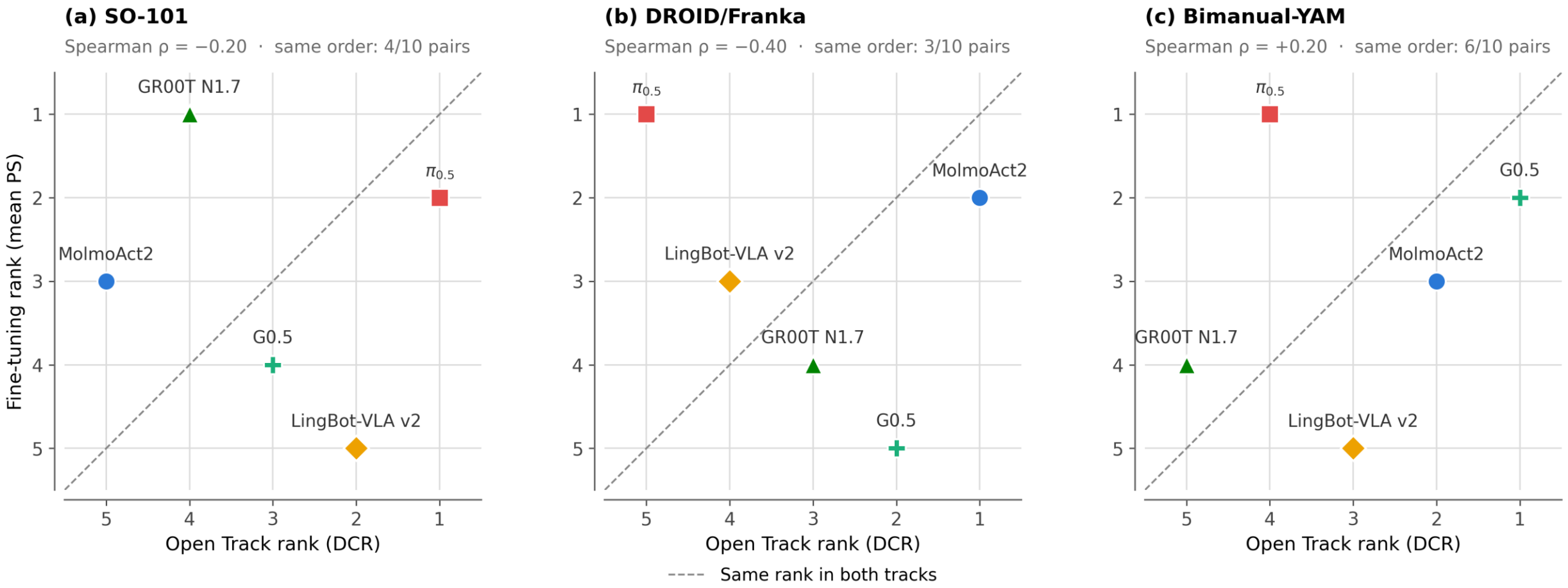
Summary: Explicit VLM Reasoning

Explicit VLM reasoning improves:

- Spatial reasoning (depth, localization)
- Long-horizon reasoning (breaking tasks into pieces)
- Better cross-embodiment transfer
- Better handling of environment feedbacks

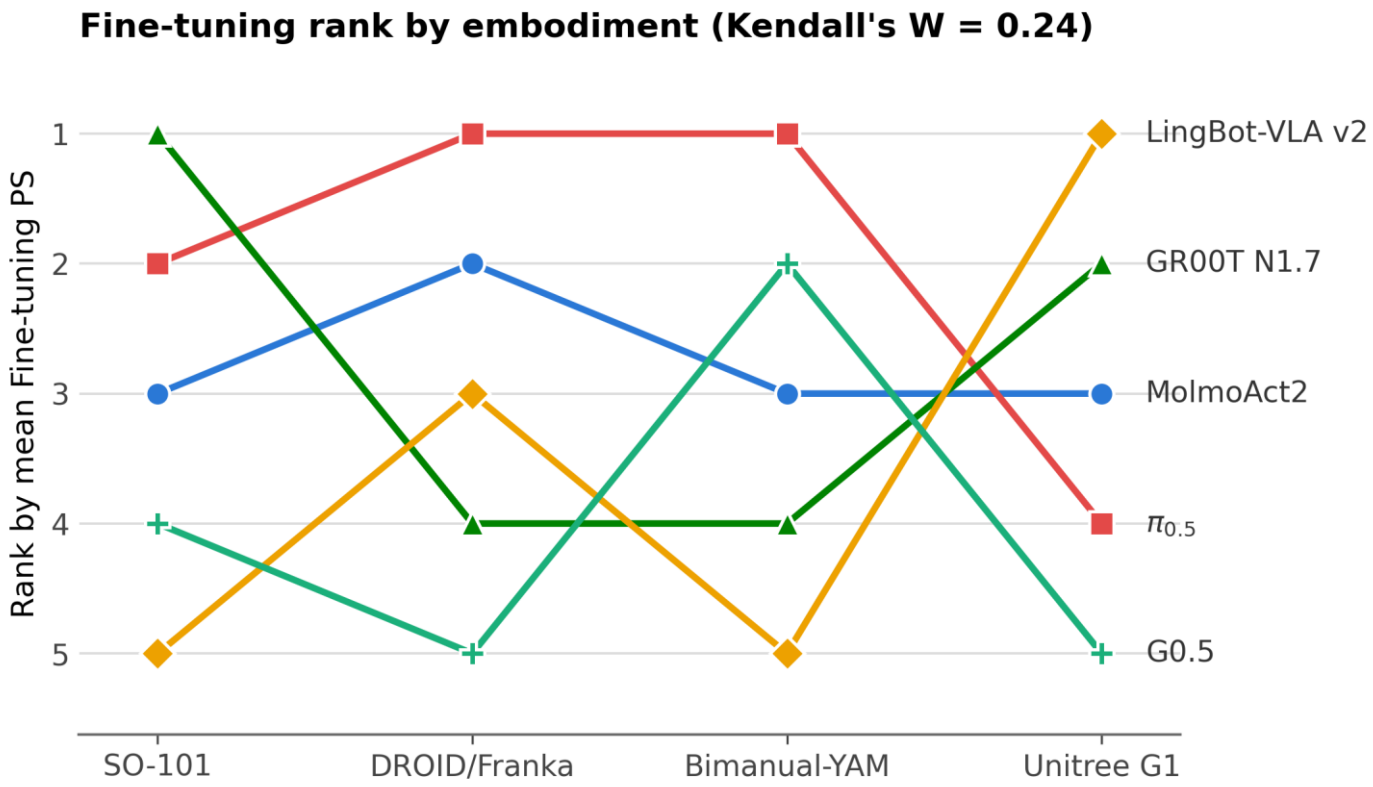
VLA Transfer to VLA

Strong open-world performance does not consistently predict post-fine-tuning performance.



Reference: RoboColosseum (https://arxiv.org/abs/2611.02497)

Relative policy strengths vary across robot embodiments and task settings.



Reference: RoboColosseum (https://arxiv.org/abs/2611.02497)

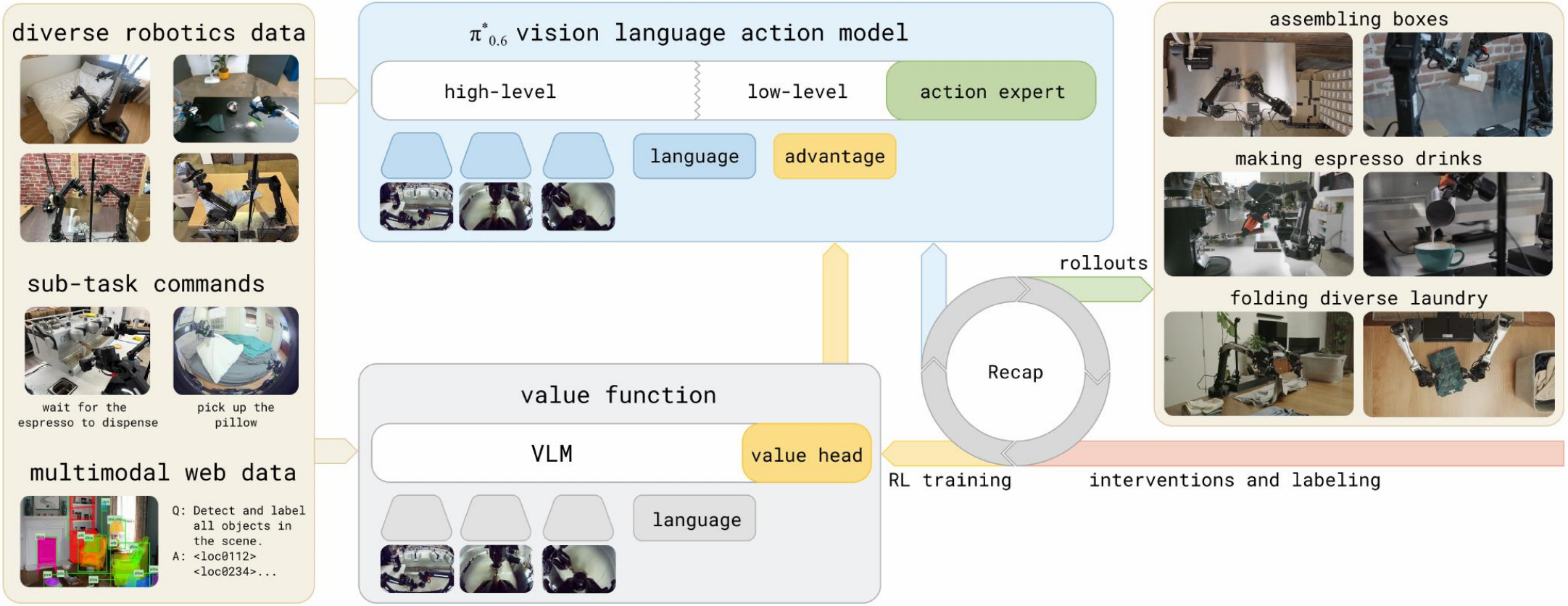
Summary: VLA “Finetunability”

Strong base performance doesn’t mean “it is easy to finetune for a new robot or a new task”!

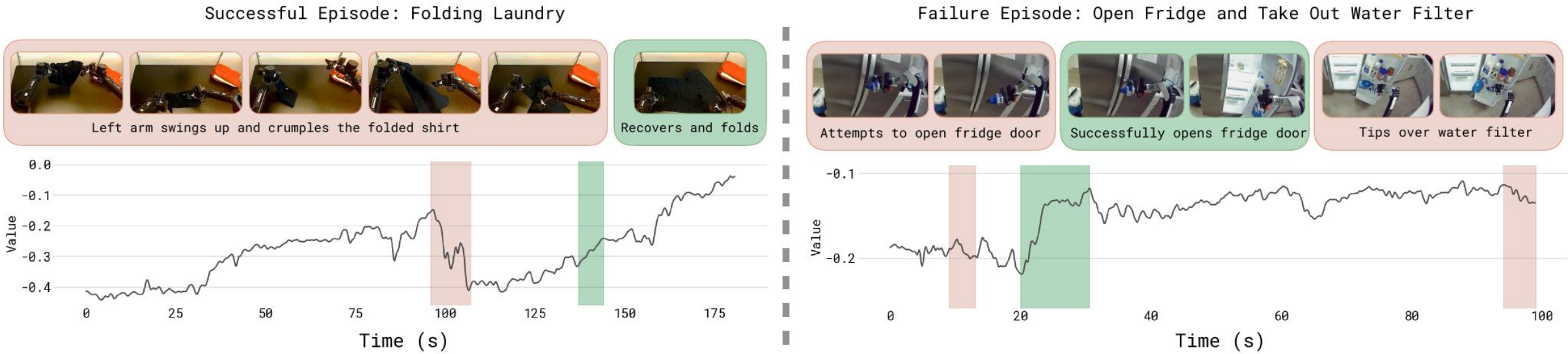
Reference: Lee et al. (2025), MolmoAct · Gemini Robotics Team (2025), Gemini Robotics 1.5

Reinforcement Learning with VLAs

π*0.6: Learning from Experience



RECAP: Learning a Value Function



Episode event	Reward
Nonterminal step	−1
Successful / failed termination	0 / large negative penalty

RL 101: Advantage

$$\hat{A}_t = \sum_{k=0}^{N-1} r_{t+k} + V_\phi(o_{t+N}, \ell) - V_\phi(o_t, \ell)$$

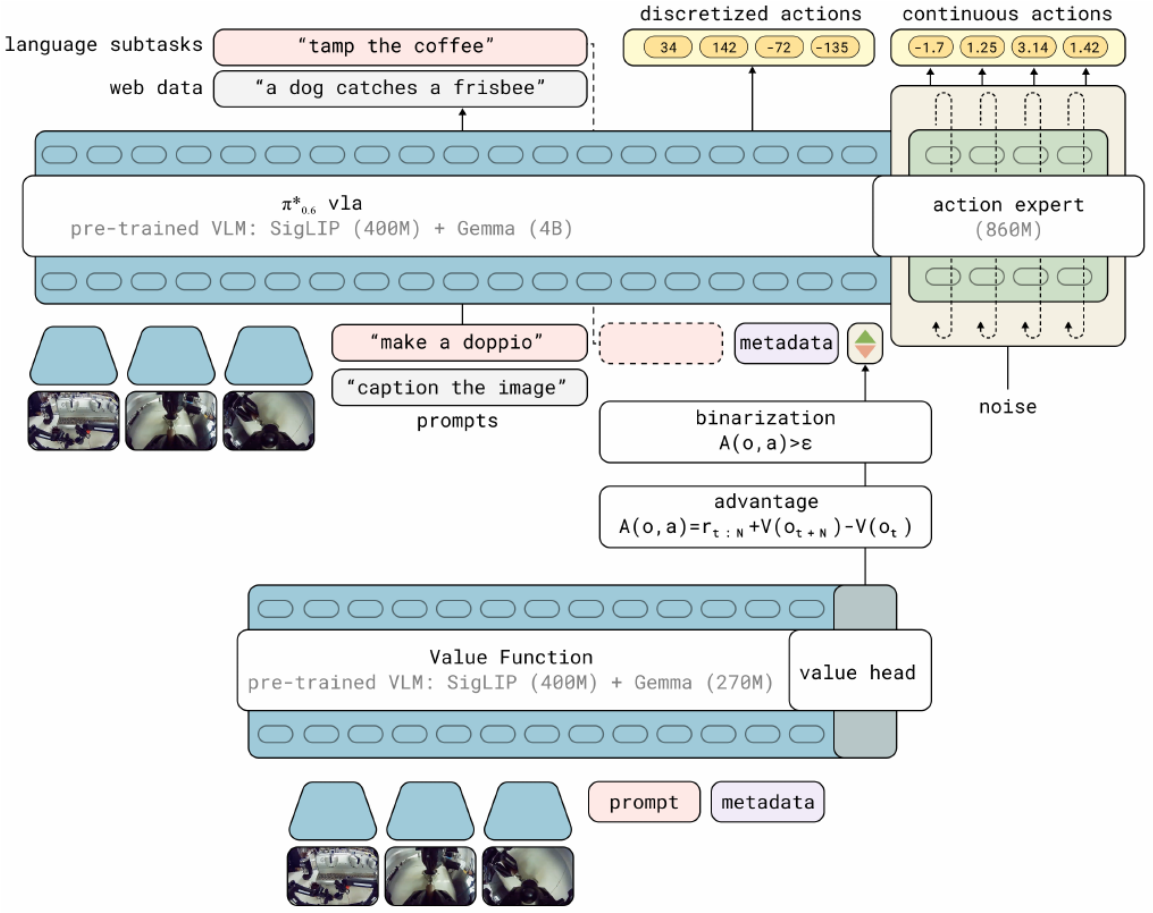
Observed reward + estimated future return − estimated current return

$$I_t = \mathbf{1}\left[\hat{A}_t > \epsilon_\ell\right]$$

Pi0.6 Hack: Use a task-dependent threshold

Reference: Physical Intelligence (2025), π*0.6 / RECAP · Appendix F; Sec. IV-B

Advantage-Conditioned Policy Training



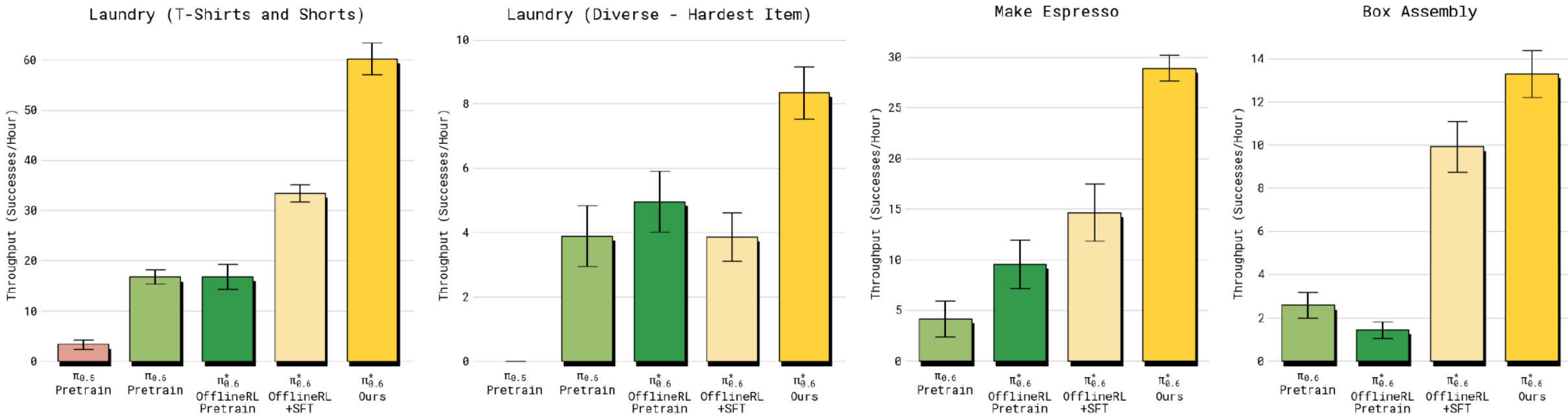
$$a \sim \pi_\theta(\cdot \mid o, \ell, I = 1)$$

Train with improving and non-improving examples

A trick in RL to stabilizes training

Reference: Physical Intelligence (2025), π*0.6 / RECAP · Fig. 3; Secs. IV-B, V

Results



Reference: Physical Intelligence (2025), π*0.6 / RECAP · Fig. 7; Sec. VI-C

Summary: Reinforcement Learning in VLAs

Train value functions (VLMs), advantages and policy functions (VLAs)
Apply standard RL tricks to stabilize training

Reference: Physical Intelligence (2025), π*0.6 / RECAP