INTRODUCTION TO COMPUTER VISION

Lecture 19 – Vision-Language-Action (VLA) Models

Gyeongsik Moon

Visual Computing and AI Lab

Korea University

Slide credit: Yunzhu Li

Robot Training with RL

sim_output = []

1. In the physics-based simulator run below loop: for t in range(T)
    1. Forward state (robot's current position/rotation and target position) to the policy network
    2. The policy network outputs an action from the input state
    3. sim_output.append({'state': input_state, 'action': action})
    4. Apply the action using a PD controller
    5. Obtain updated robot's current state
2. Compute reward (high much close to the target) for all elements in sim_output
3. For some elements in sim_output, increase probability of the output action with high reward

Robot Training with RL

Roll out

sim_output = []

1. In the physics-based simulator run below loop: for t in range(T)
    1. Forward state (robot's current position/rotation and target position) to the policy network
    2. The policy network outputs an action from the input state
    3. sim_output.append({'state': input_state, 'action': action})
    4. Apply the action using a PD controller
    5. Obtain updated robot's current state
2. Compute reward (high much close to the target) for all elements in sim_output
3. For some elements in sim_output, increase probability of the output action with high reward

Robot Training with RL

sim_output = []

1. In the physics-based simulator run below loop: for t in range(T)
    1. Forward state (robot's current position/rotation and target position) to the policy network
    2. The policy network outputs an action from the input state
    3. sim_output.append({'state': input_state, 'action': action})
    4. Apply the action using a PD controller
    5. Obtain updated robot's current state
2. Compute reward (high much close to the target) for all elements in sim_output
3. For some elements in sim_output, increase probability of the output action with high reward

Loss function

Robot Training with RL

- There are two problems

1. Roll out
    - Too slow
    - We need to run for loop in physics simulator
    - That makes the training really slow
2. Loss function
    - No explicit target for the output action
    - RL: “do the estimated action with high reward again!”
    - Supervised learning: “do the ‘target’ action instead of the estimated action!”

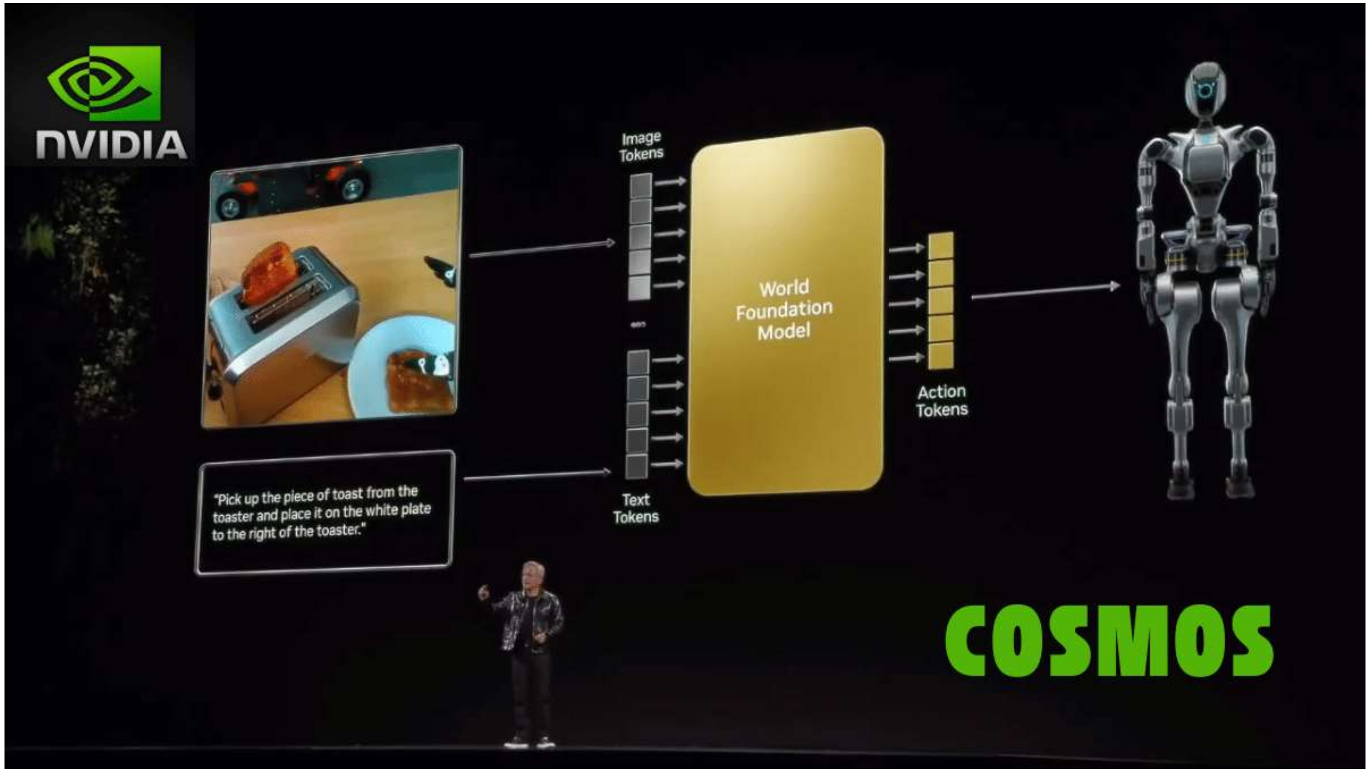
Why Reinforcement Learning?

Slide from Lecture 18

- If you're in the two situations, you can try RL
  - Infeasible data collection
  - Non-differentiable output function
- Representative examples are agents in video game and robots

Vision-language-action (VLA) models

- The good points of RL becomes limitations of RL at the same time
- How to solve this?
    - Vision-language-action (VLA) models

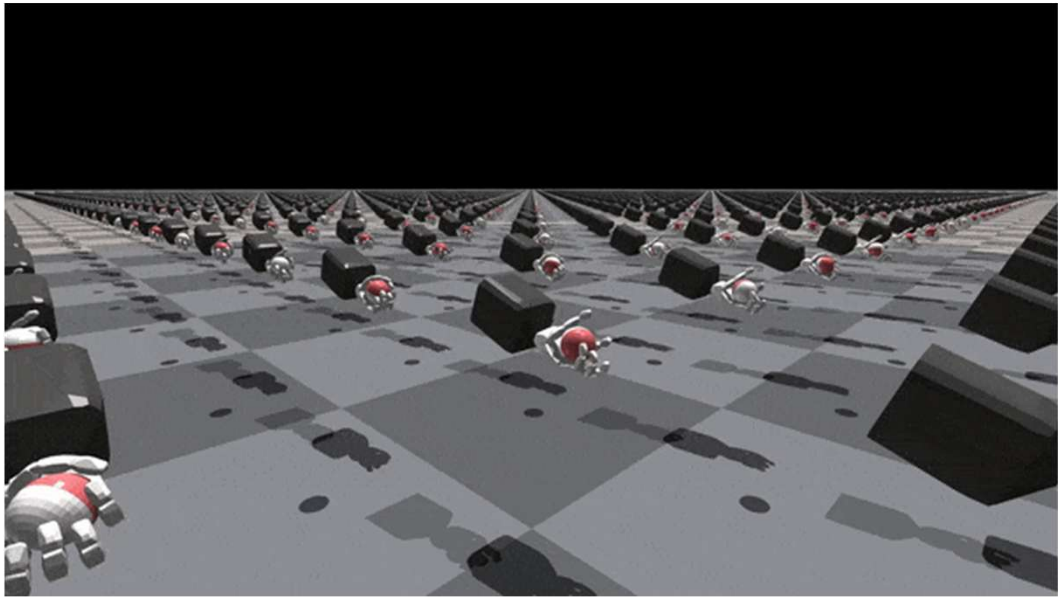


Vision-language-action (VLA) models

- Use supervised learning instead of RL for the robot training
- How to get data?

Real robot data (telooperations)

RL in physics simulators (we've learned so far)



[1] Fu, Zipeng, Tony Z. Zhao, and Chelsea Finn. "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation." CoRL. 2024.
[2] Andrychowicz, OpenAI: Marcin, et al. "Learning dexterous in-hand manipulation." The International Journal of Robotics Research 39.1 (2020): 3-20.

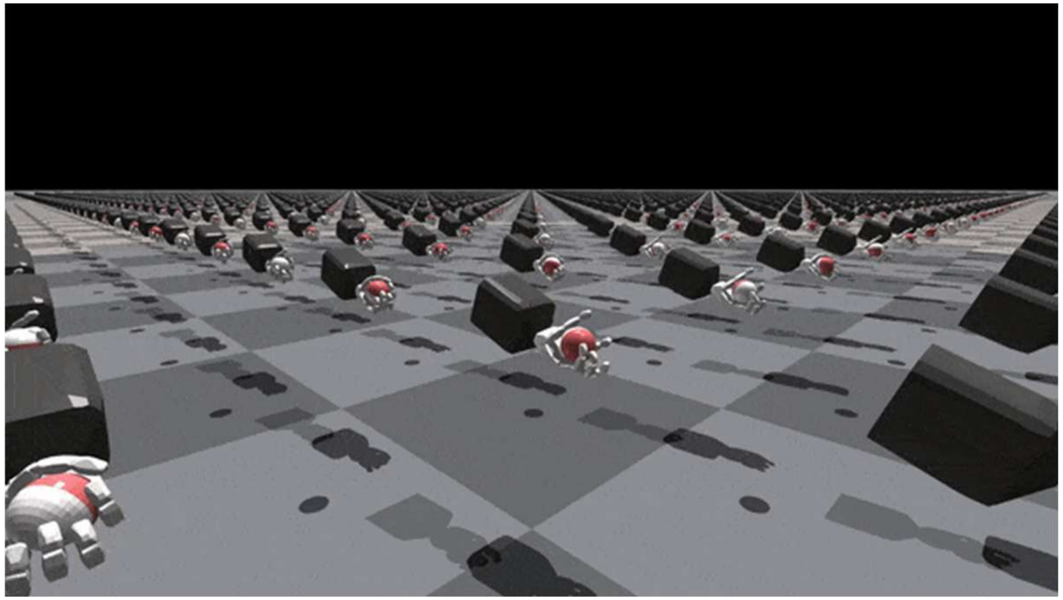
Teleoperations

- Use real robots to collect data
- Humans control robots to get robot trajectories
- Costly and hard to scale up

[1] Fu, Zipeng, Tony Z. Zhao, and Chelsea Finn. "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation." CoRL. 2024.

RL in Physics Simulators

- Use RL for each task instead of building a universal policy network
- For example, train a policy network with RL only for hand-obj grasp
- In this way, we can avoid scale-up issue of RL
- Sim2Real gap: Gap between simulation environments and real worlds

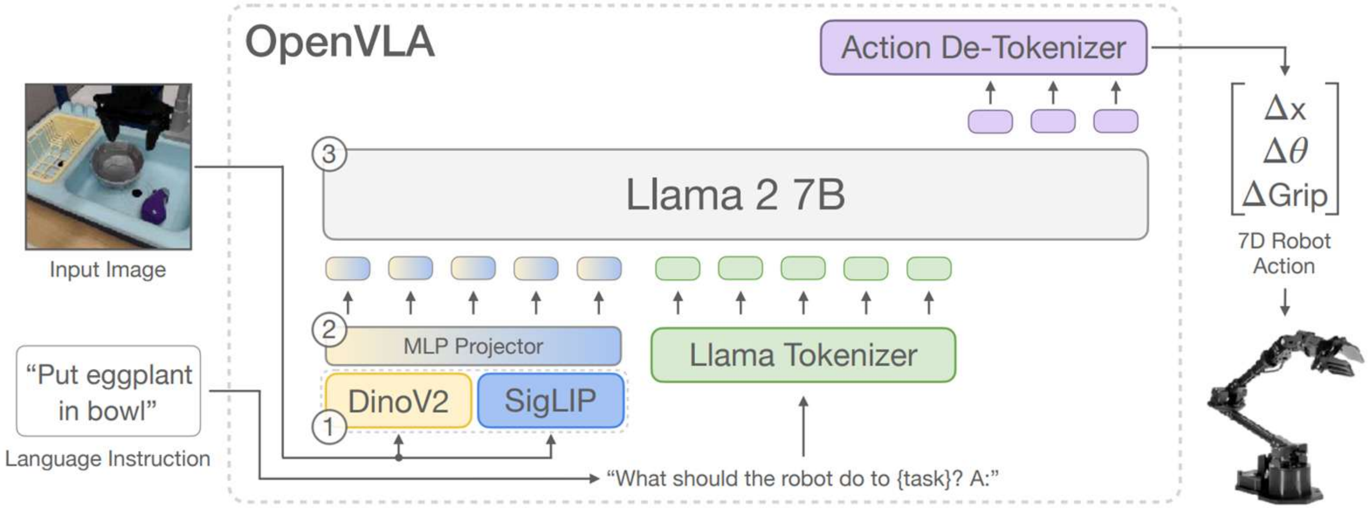


[1] Andrychowicz, OpenAI: Marcin, et al. "Learning dexterous in-hand manipulation." The International Journal of Robotics Research 39.1 (2020): 3-20.

Vision-language-action (VLA) models

- Use supervised learning instead of RL for the robot training
- *Now, we have data (pairs of (state, action))*
- Loss = distance(net(state) – action_target)
- No roll out
- With explicit target

Vision-language-action (VLA) models



processed into a sequence of tokens, OpenVLA is trained with a standard next-token prediction objective, evaluating the cross-entropy loss on the predicted action tokens only. We discuss key design decisions for implementing this training procedure in Section 3.4. Next, we describe the robot dataset we use for OpenVLA training.

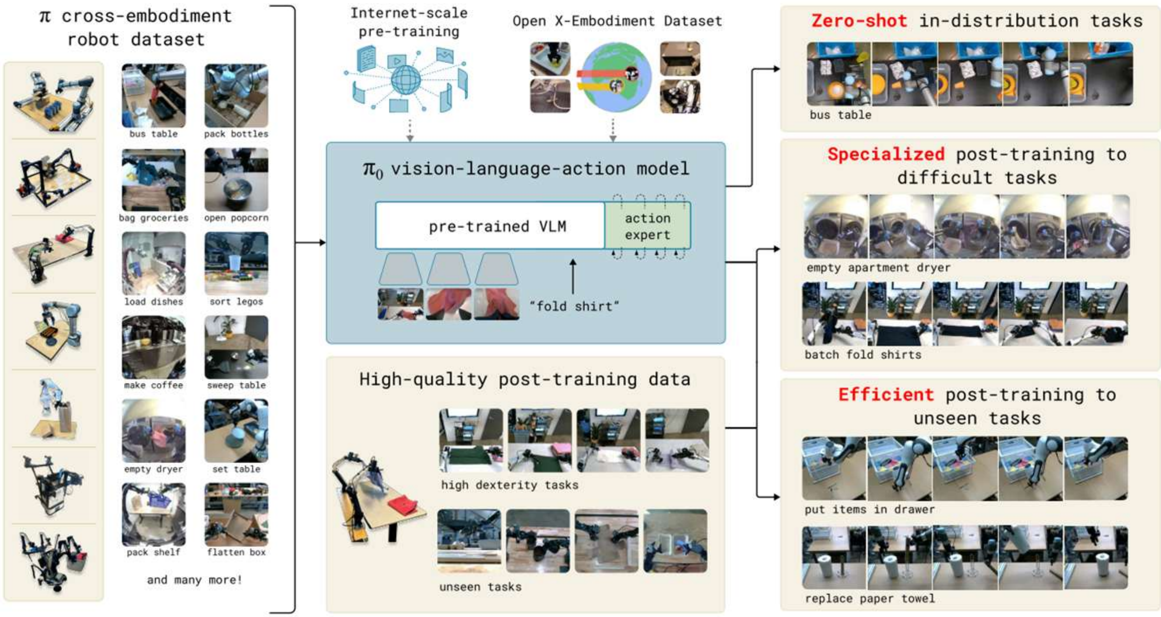
I guess he's Korean..? He's a Ph.D. candidate in Stanford!

[1] Kim, Moo Jin, et al. "OpenVLA: An open-source vision-language-action model." CoRL. 2024.

Vision-language-action (VLA) models

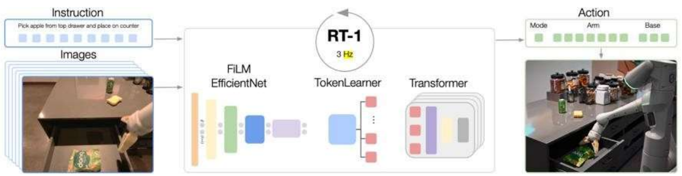
π_0: A Vision-Language-Action Flow Model for General Robot Control

Physical Intelligence
Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, Ury Zhilinsky
https://physicalintelligence.company/blog/pi0

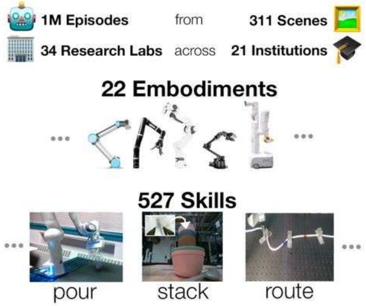


- Combination of VLA model and diffusion generative models
- Denoise action conditioned on image, language, and robot state

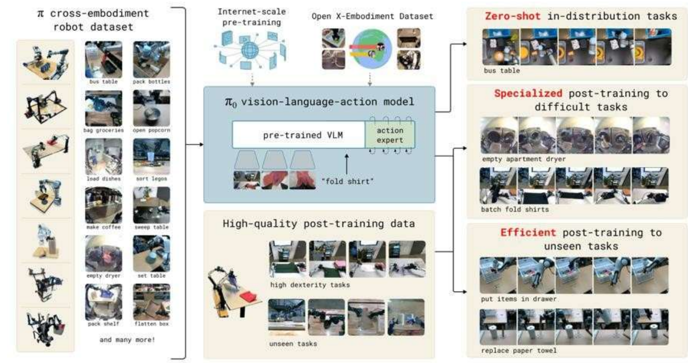
Robotic Foundation Models (fancy name of VLA)



RT-1 (Dec. 2022)

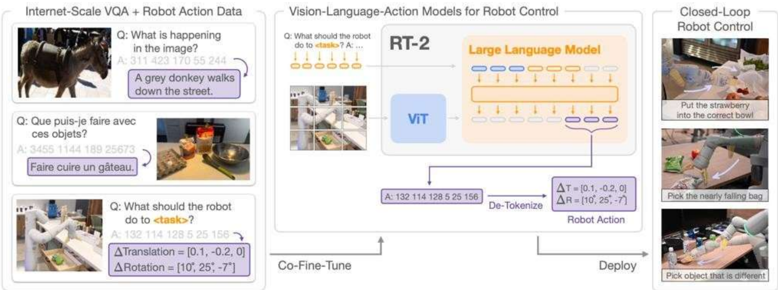


RT-X (Oct. 2023)

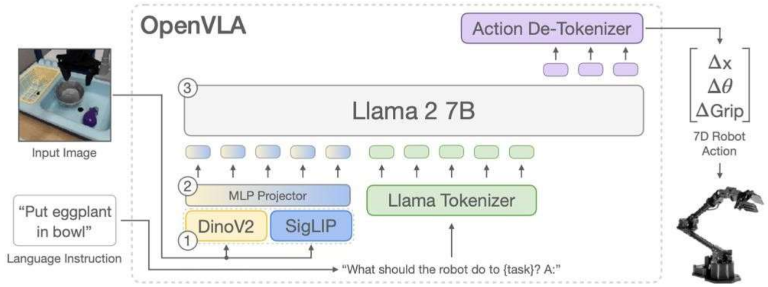


Pi-Zero (Oct. 2024)

RT-2 (Jul. 2023)



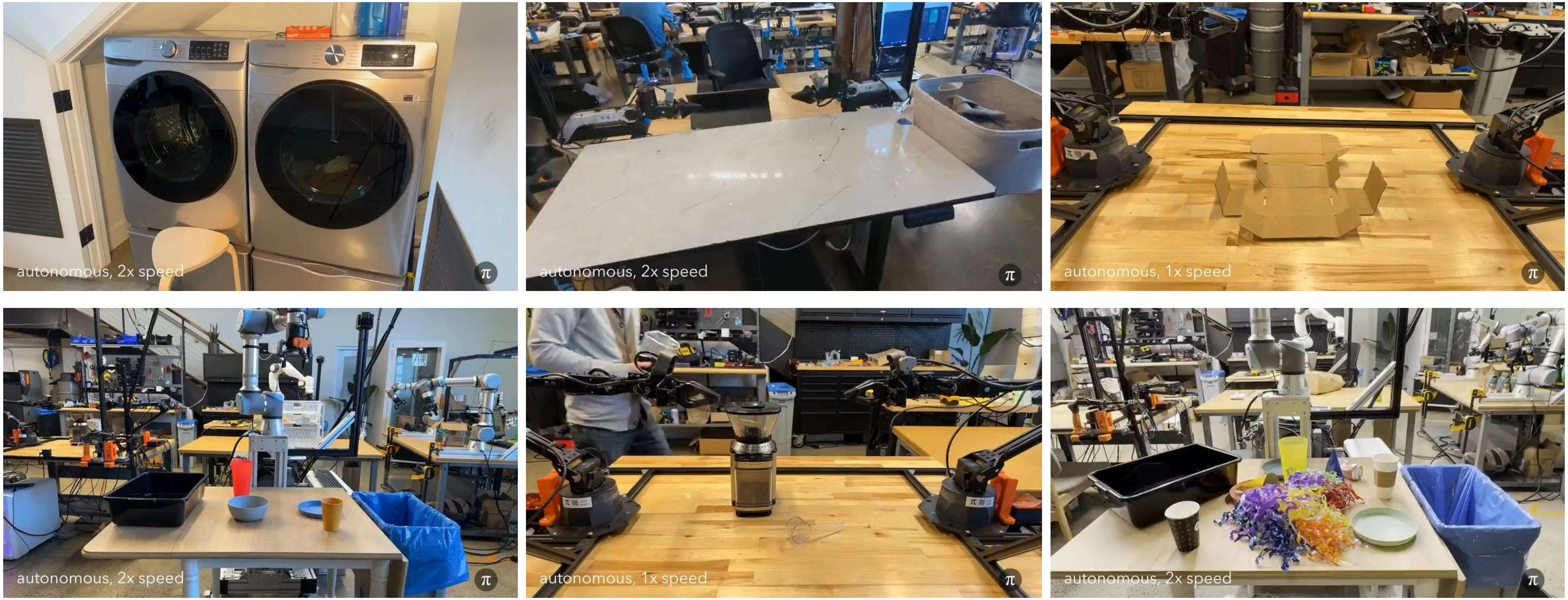
OpenVLA (Jun. 2024)



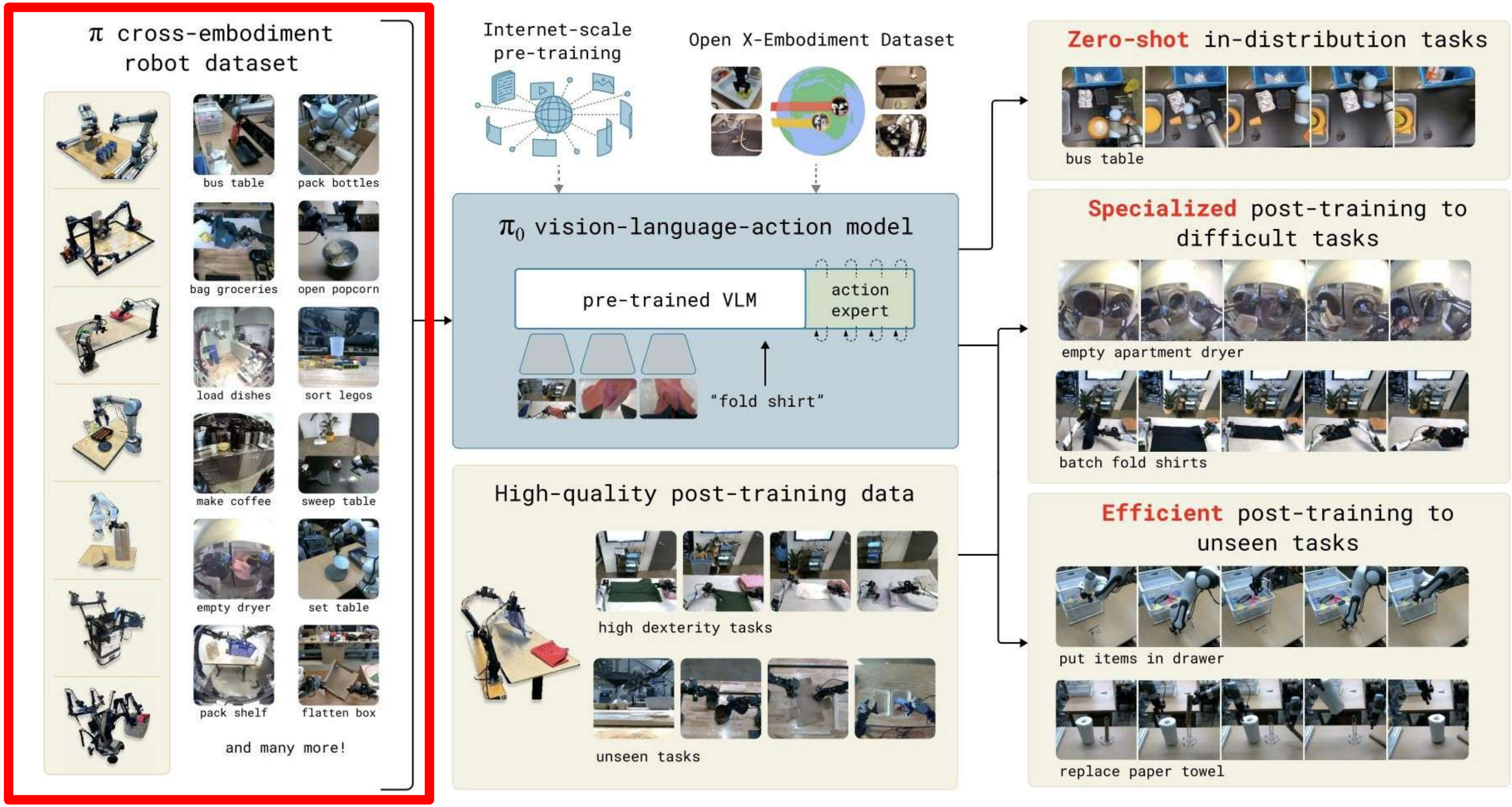
Helix (Figure)
Hi-Robot (PI)
Gemini Robotics
Pi-0.5 (PI)
GR00T (Nvidia)
DYNA-1
…

Pi-Zero by Physical Intelligence

https://www.physicalintelligence.company/blog/pi0



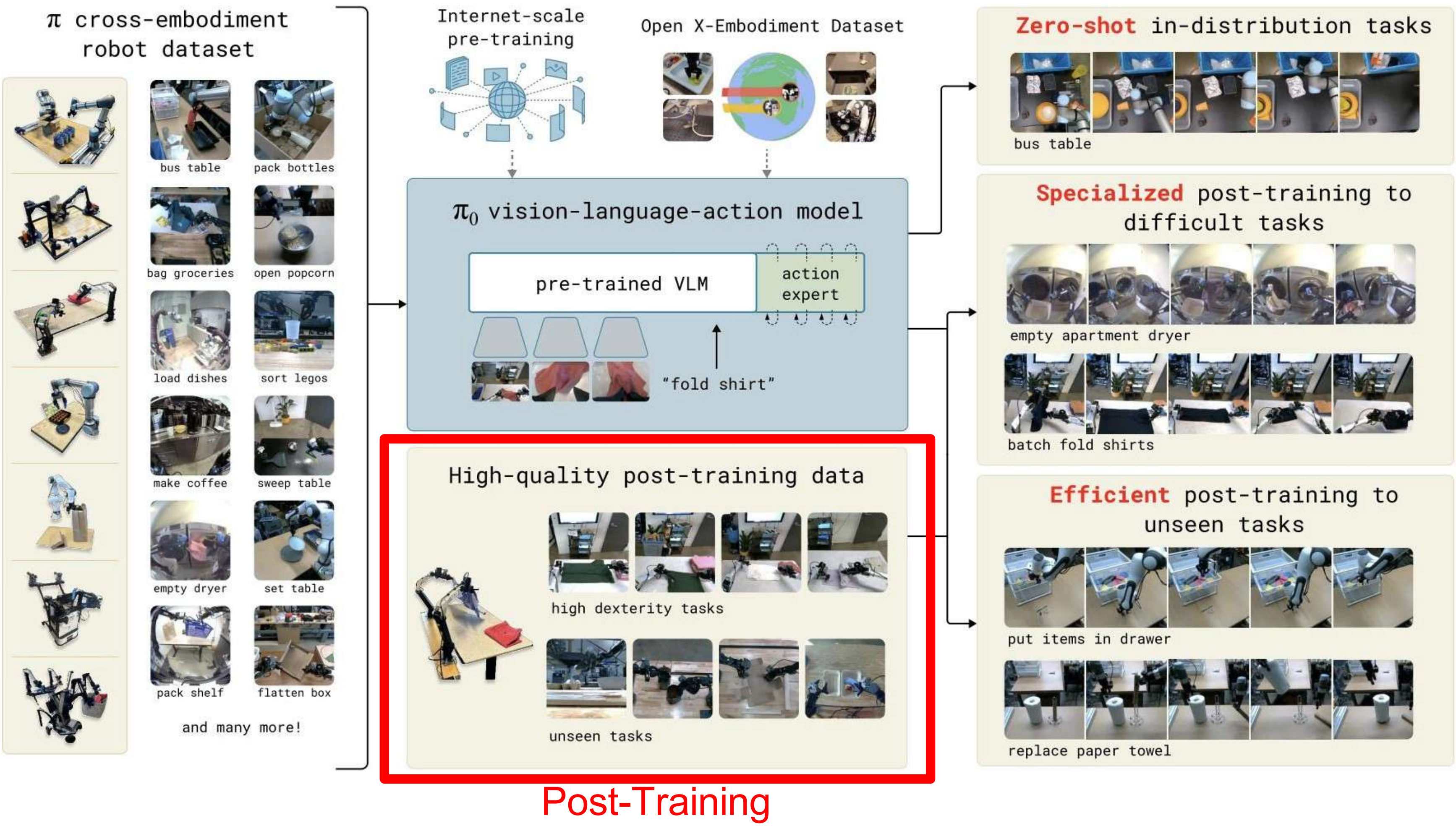
Pi-Zero by Physical Intelligence



Cross-Embodiment Dataset

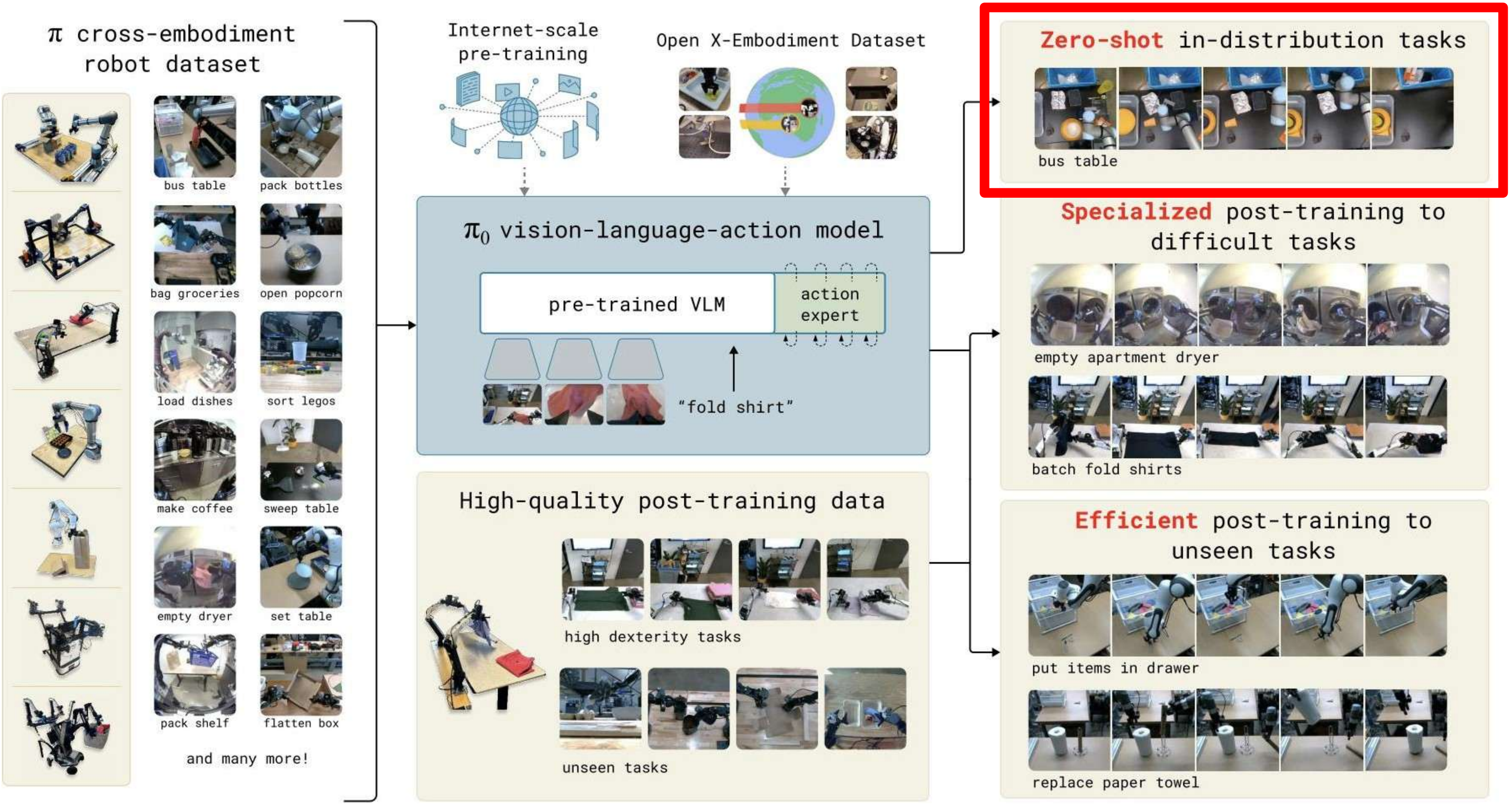
Pi-Zero by Physical Intelligence

Pi-Zero by Physical Intelligence



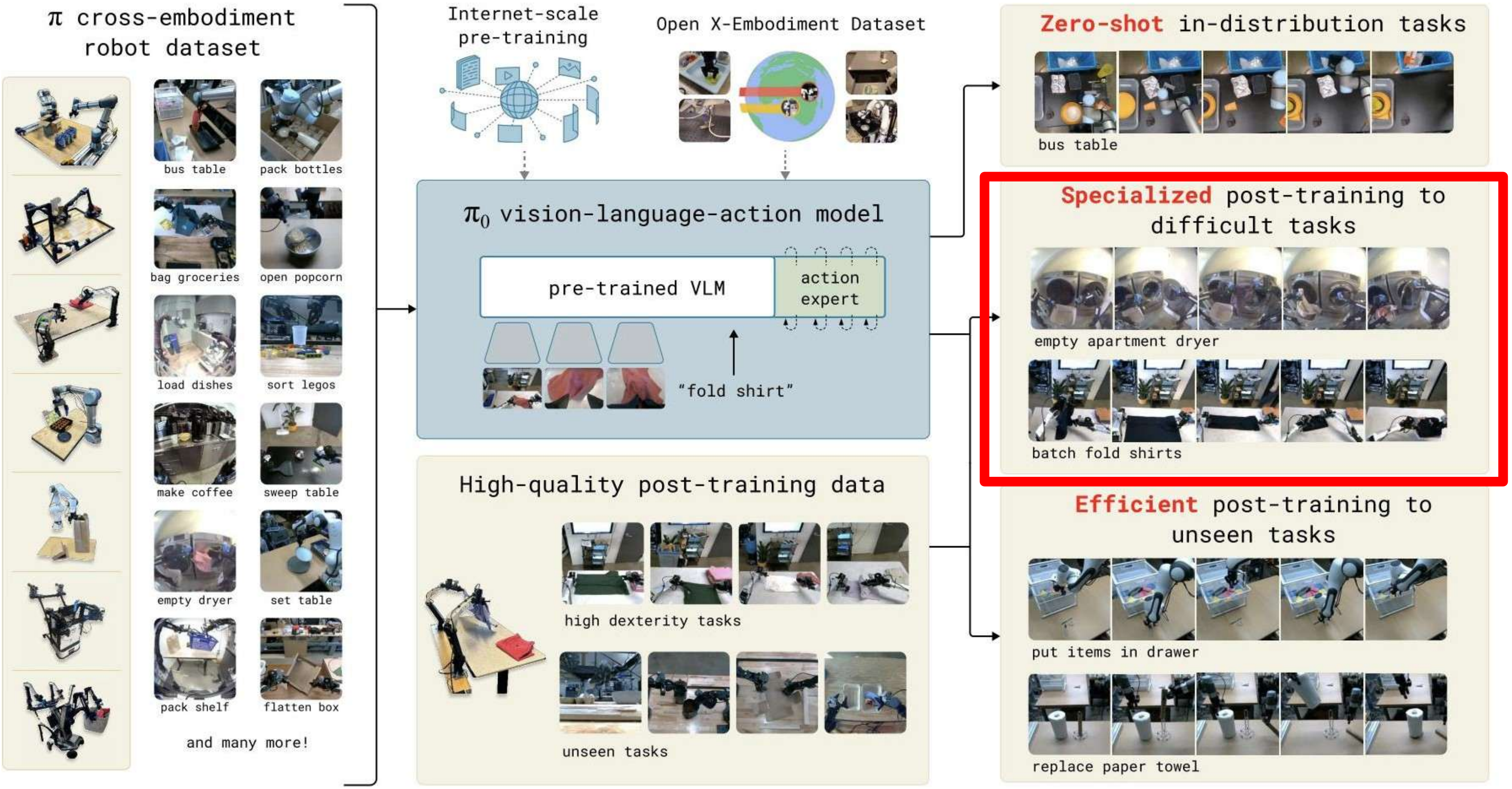
Pi-Zero by Physical Intelligence

Simple in-distribution tasks

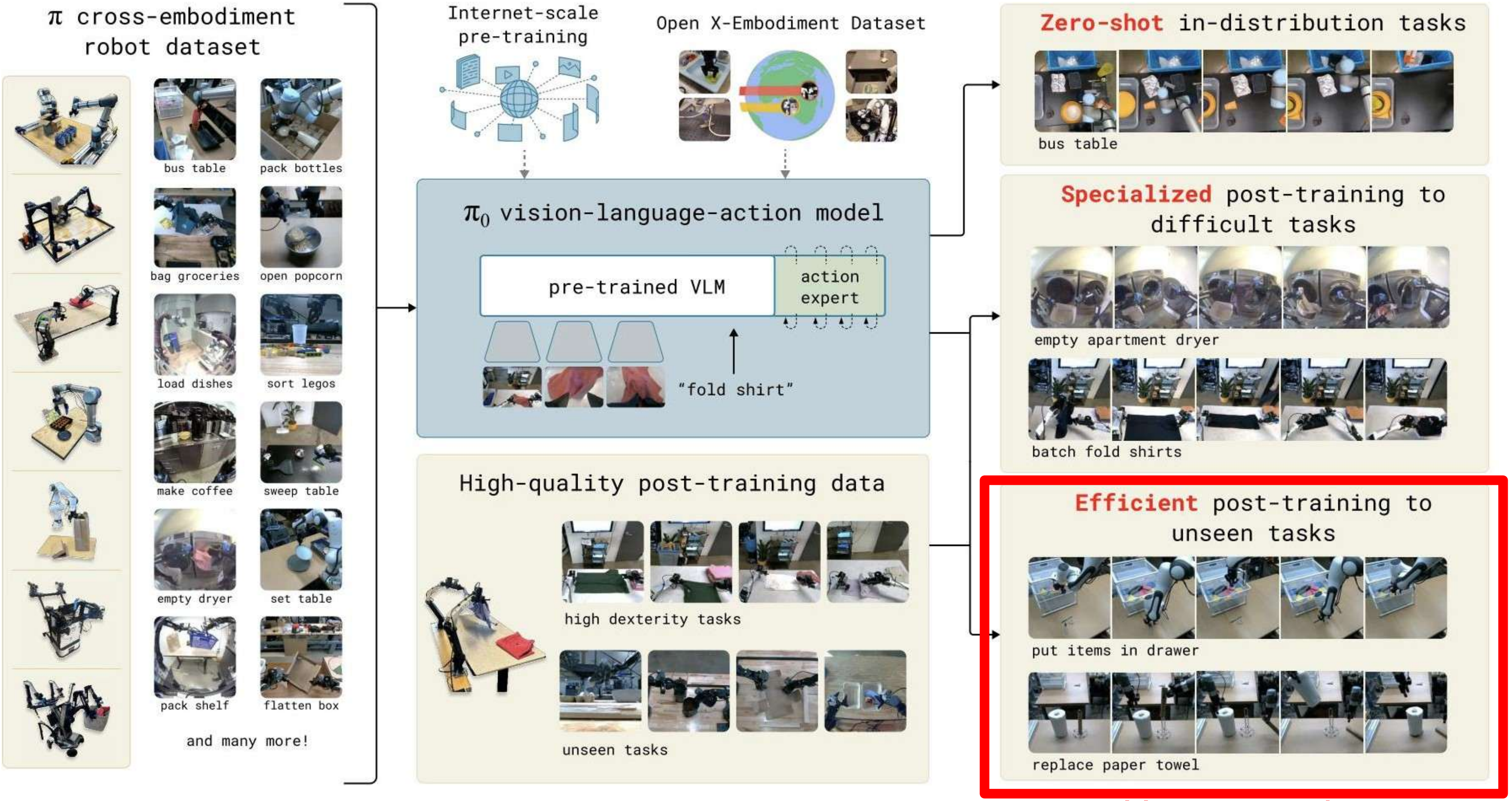


Pi-Zero by Physical Intelligence

Complicated in-distribution tasks



Pi-Zero by Physical Intelligence



Unseen tasks

Pi-Zero by Physical Intelligence

Physical Intelligence (π)

Open Sourcing π_0

Published	February 4, 2025
Email	research@physicalintelligence.company
Repo	Physical-Intelligence/openpi

README Apache-2.0 license

openpi

openpi holds open-source models and packages for robotics, published by the Physical Intelligence team.

Currently, this repo contains two types of models:

- the π_0 model, a flow-based diffusion vision-language-action model (VLA)
- the π_0-FAST model, an autoregressive VLA, based on the FAST action tokenizer.

For both models, we provide *base model* checkpoints, pre-trained on 10k+ hours of robot data, and examples for using them out of the box or fine-tuning them to your own datasets.

This is an experiment: π_0 was developed for our own robots, which differ from the widely used platforms such as ALOHA and DROID, and though we are optimistic that researchers and practitioners will be able to run creative new experiments adapting π_0 to their own platforms, we do not expect every such attempt to be successful. All this is to say: π_0 may or may not work for you, but you are welcome to try it and see!

Evaluation of the Robot Learning Models

- Evaluation is primarily conducted in the real world
- Real-world evaluation is costly and noisy
  - "We have large enough budget such that we can still make progress."
- Weak correlation between training loss and real-world success rate.
  - Training objectives vs task-specific metrics, training vs testing horizons

ALOHA 2

Evaluation of the Robot Learning Models

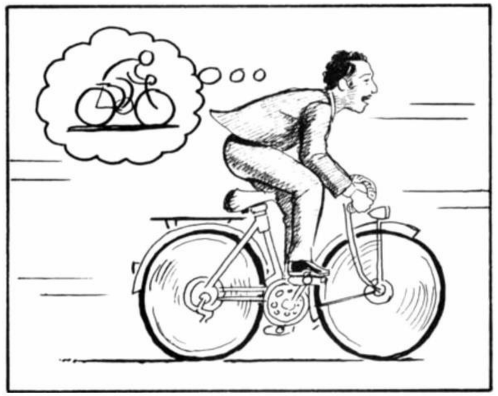
What about evaluation in simulation?

- Sim-to-real gap: rigid / deformable / cloth
- Efficient asset generation
- Digitalization of the real world
- Procedural generation of realistic and diverse scenes
- Correlation between sim and real

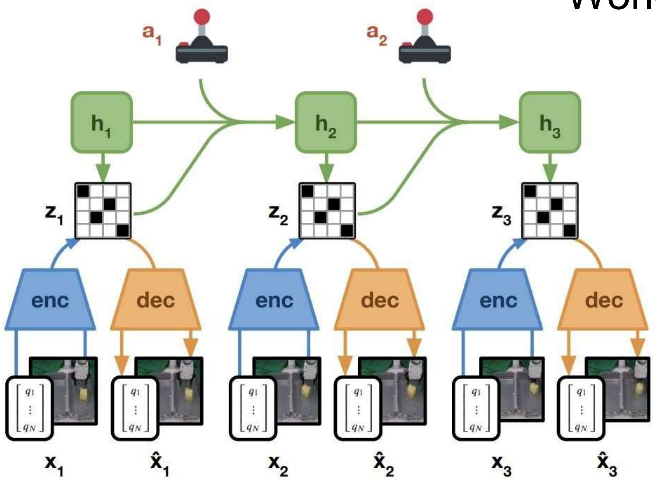
ImageNet in Embodied AI?

Habitat 3.0

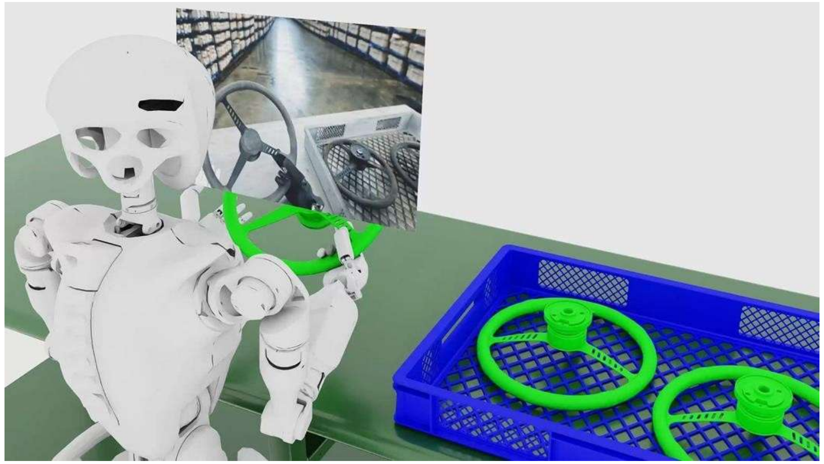
Robotic Foundation Model + World Models


World Models


1X World Models



DayDreamer

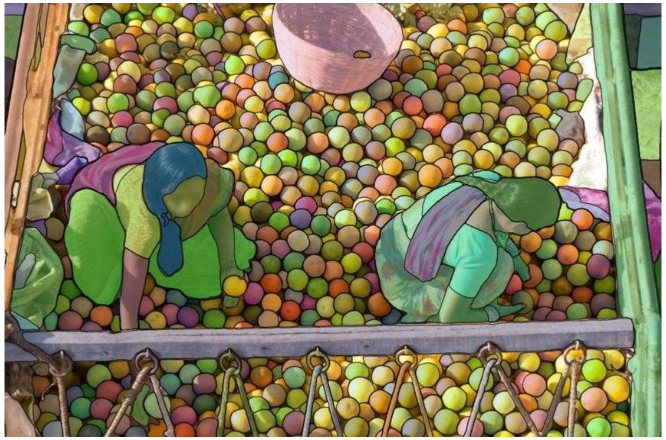

Nvidia Cosmos - World Foundation Model

- 3D?
- Structural Prior?
- Learning + Physics?
- Corr. w/ Real World

Foundation Models for Embodied Agents

- Current foundation models are not tailored for embodied agents
  - LLM/VLM can fail in embodied-related tasks
  - Limited understanding of geometric / embodied / physical interactions
  - Reinforcement learning (RL) from human feedback → RL from Embodied Feedback

GPT



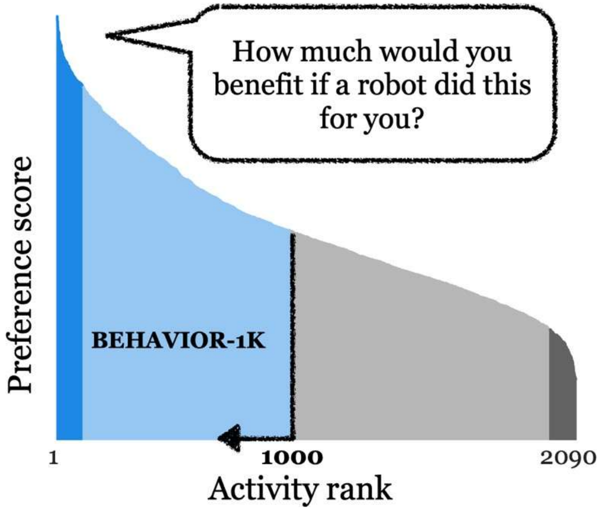
Segment Anything

DINOv2

Adaptation / Life-Long Learning

- Adapt to new scenarios
- Adapt to human preferences
- Self improve / life-long learning

Adapt to new scenarios



Adapt to human preferences

Improve through experience