Robot Learning

Vision Language Action Models

Prof. Songhwai Oh
ECE, SNU

RT-1: Robotics Transformer for Real-World Control at Scale

A. Brohan, et al.,
in Proc. of Robotics: Science and Systems (RSS), Jul, 2023.

RT-1

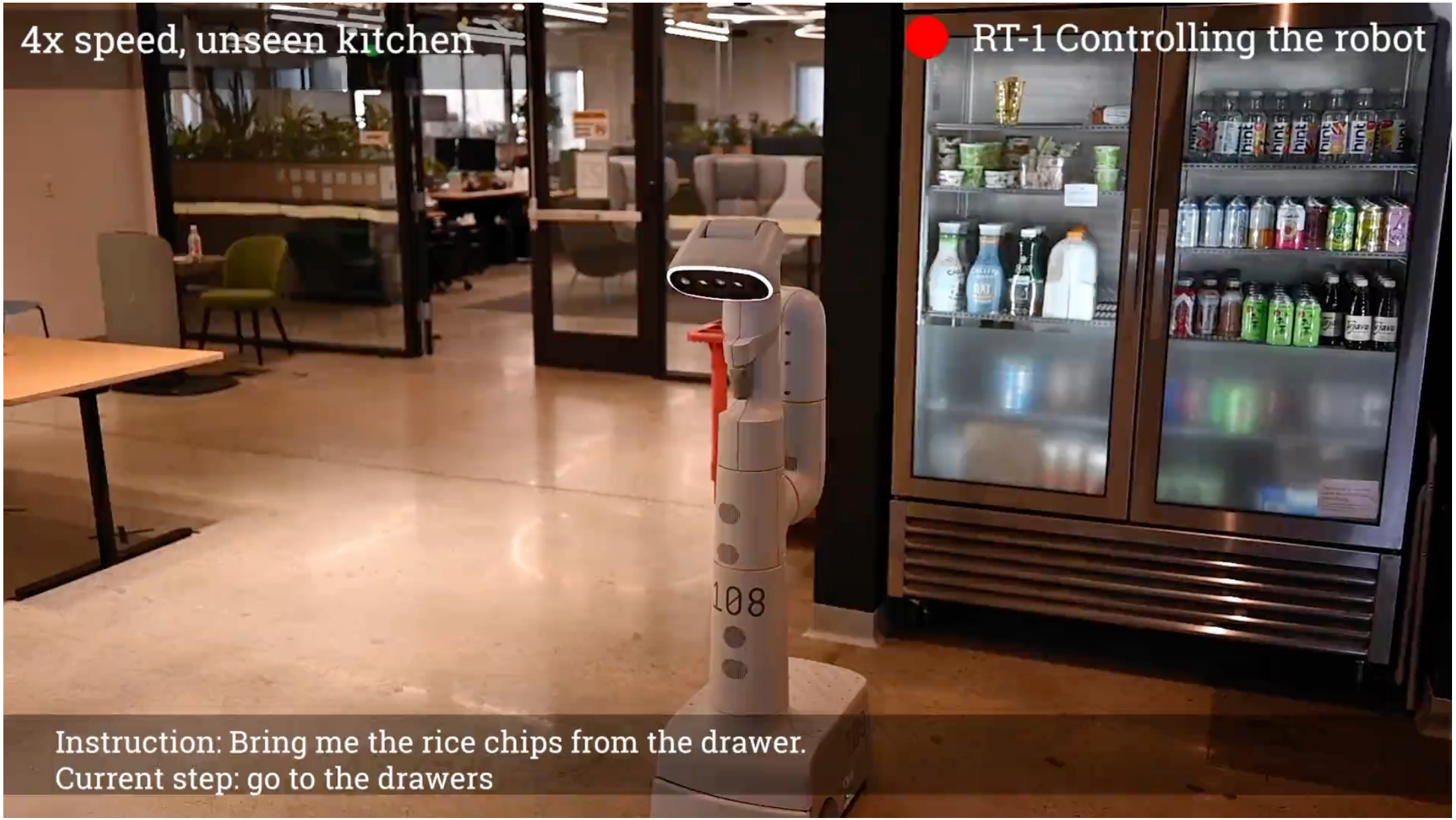
- Limitations of Existing Robot Policy Learning Methods
  - Collecting task-specific datasets in single-task or multi-task setting is narrowly tailored to individual tasks.
  - Such models often exhibit limited generalization to new tasks and environments.

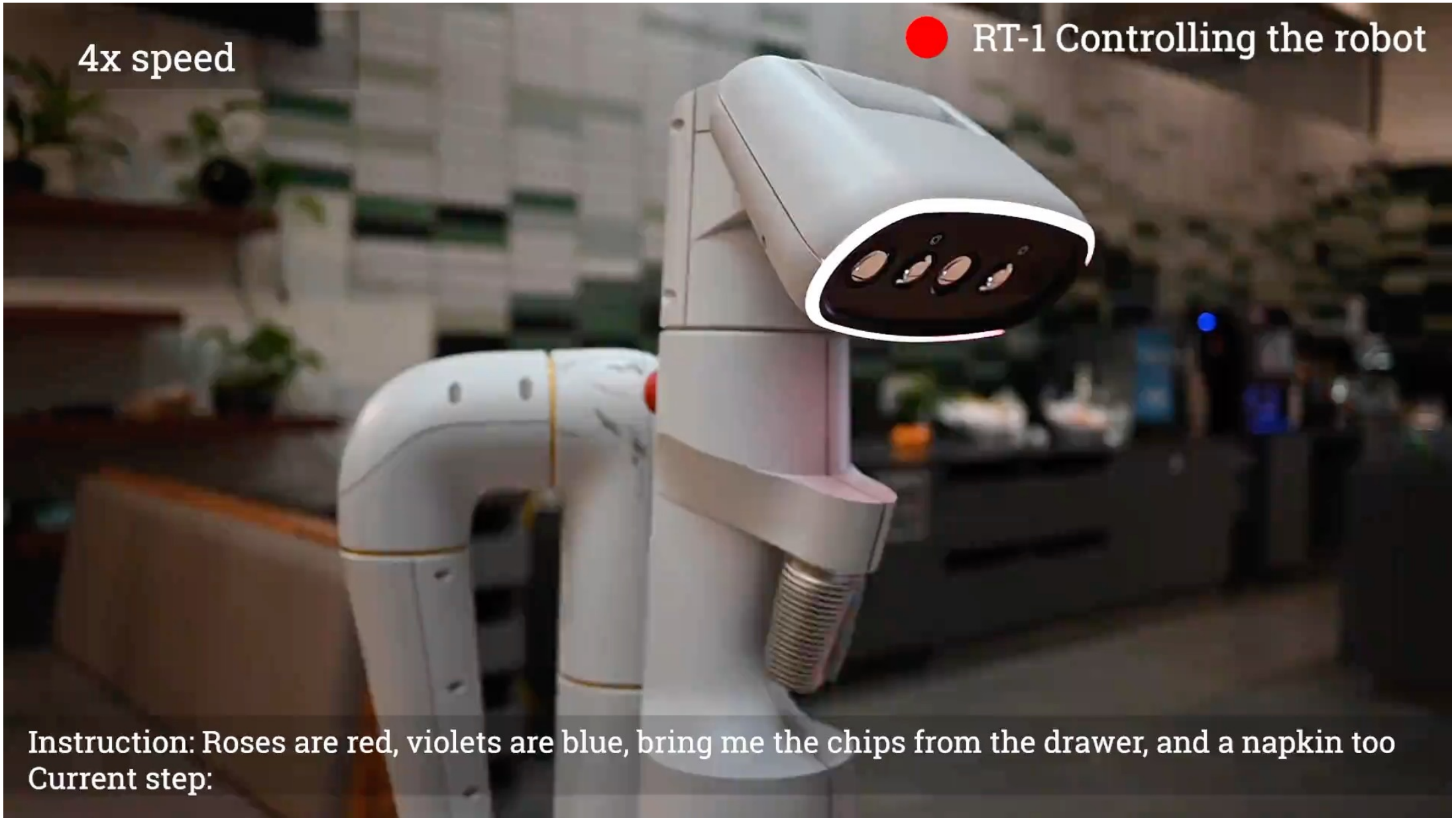
- Toward Scalable General Robot Learning
  - Can RT-1 (Robotics Transformer 1) train **a single multi-task backbone model** on data collected from a wide variety of real-world tasks and environments?
  - Does training RT-1 on a large, diverse real-world dataset enable **zero-shot generalization** to new tasks, objects, and environments?

RT-1: Architecture & Dataset

- Architecture
  - RT-1 does not rely on a pre-trained VLM
  - Model:
    - EfficientNet + FiLM (language-conditioned vision encoding)
    - Transformer (mapping vision-language tokens to discretized action tokens)
  - Model size:
    - ~35M parameters

- Dataset
  - ~130k episodes with 700+ language instructions
  - Collected using mobile manipulators with a 7-DoF arm

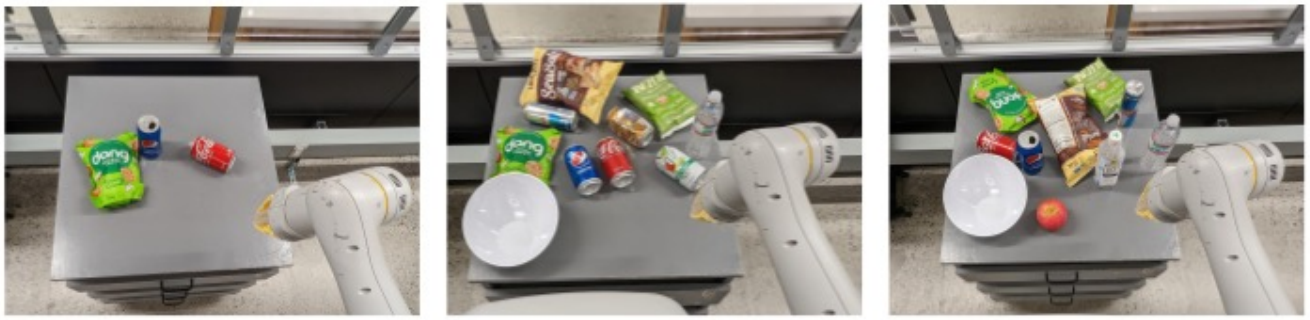

4x speed, unseen kitchen
RT-1 Controlling the robot
Instruction: Bring me the rice chips from the drawer.
Current step: go to the drawers


4x speed
RT-1 Controlling the robot
Instruction: Roses are red, violets are blue, bring me the chips from the drawer, and a napkin too
Current step:

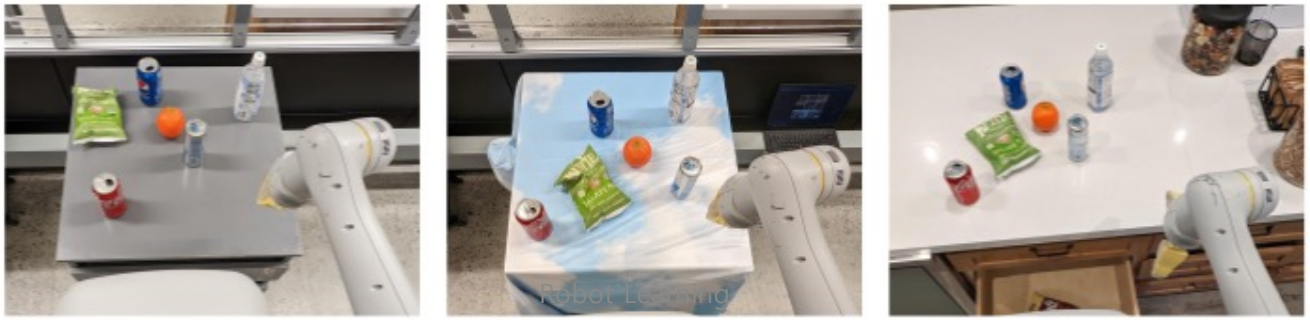
RT-1: Main Experiments

- Generalization & Robustness Experiments
  - Distractor: placing irrelevant objects around the target object
  - Background: varying kitchens, lighting conditions, and table surfaces

Distractor:

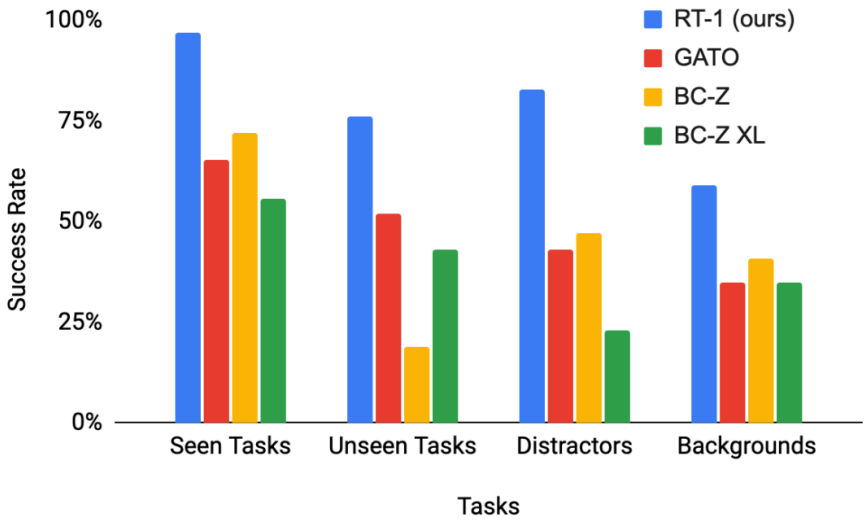


Background:



RT-1: Main Experiments

- Generalization & Robustness Experiments



Conclusion: Large-scale policies exhibit better generalization capabilities.

RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control

A. Brohan, et al.,
in Proc. of the Conference on Robot Learning (CoRL), Nov, 2023.

RT-2

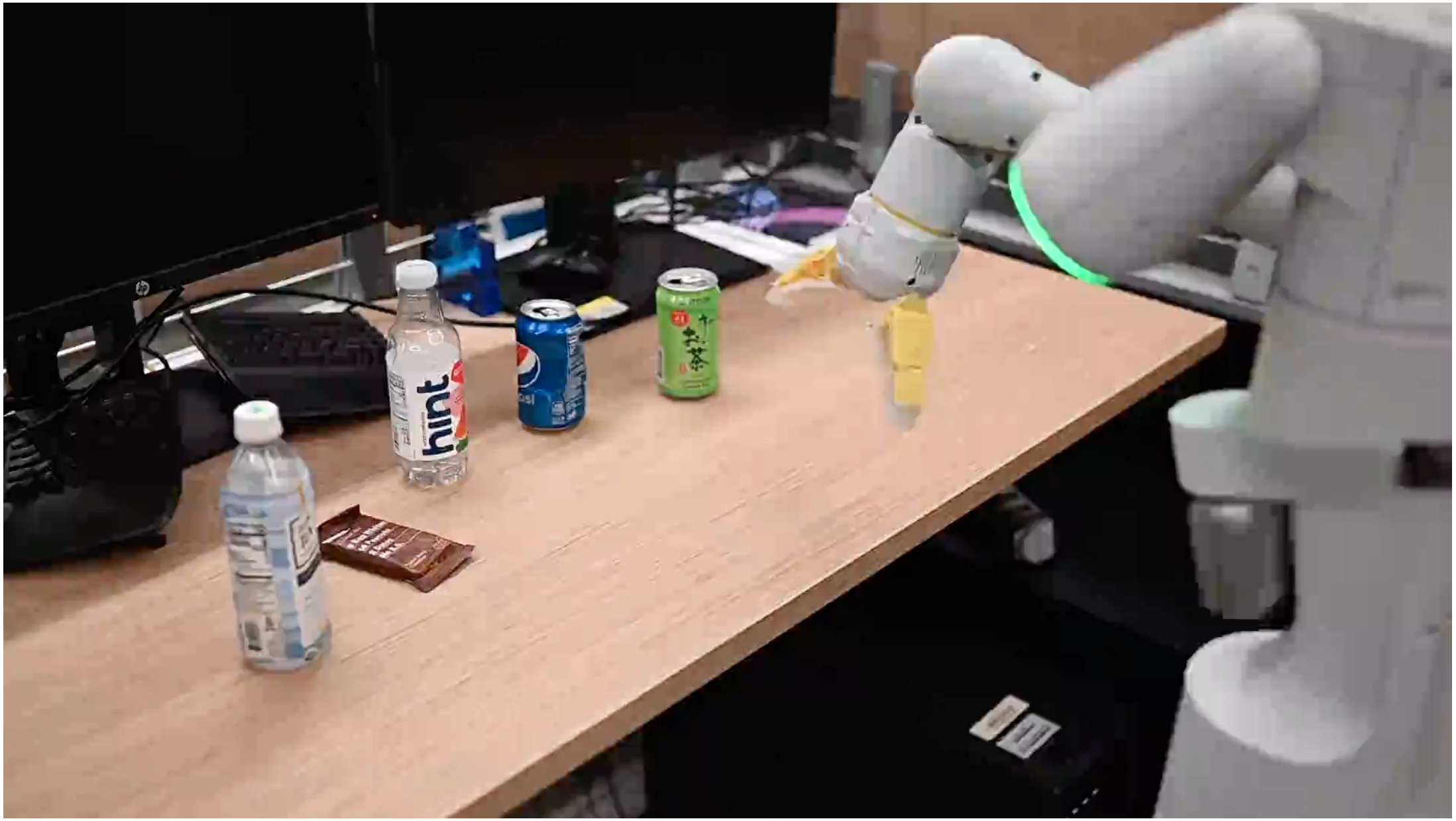
- Limitations of Existing Robot Policy Learning
  - Robot-only data limits generalization.
  - Existing approaches have limited use of VLMs:
    - Typically used only for high-level planning
    - Do not leverage rich semantic knowledge for low-level control

- Vision-Language-Action (VLA) Model
  - Leverage **pretrained VLMs** directly for robot control
  - Represent **robot actions as text tokens** in a shared language space
  - Train on both robot data and web-scale vision-language data via co-fine-tuning

RT-2: Architecture & Dataset

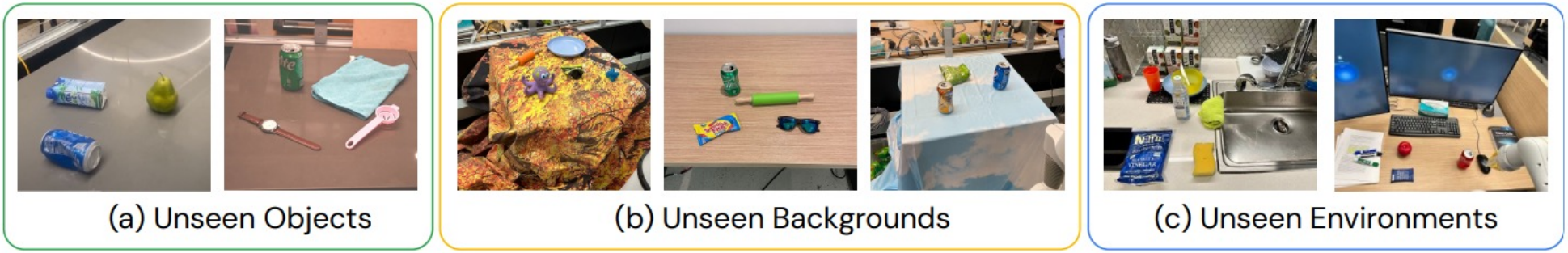
- Architecture
  - Pre-trained VLM
    - PaLI-X (5B, 55B) or PaLM-E (12B)
  - Output
    - Text tokens → Discretized actions

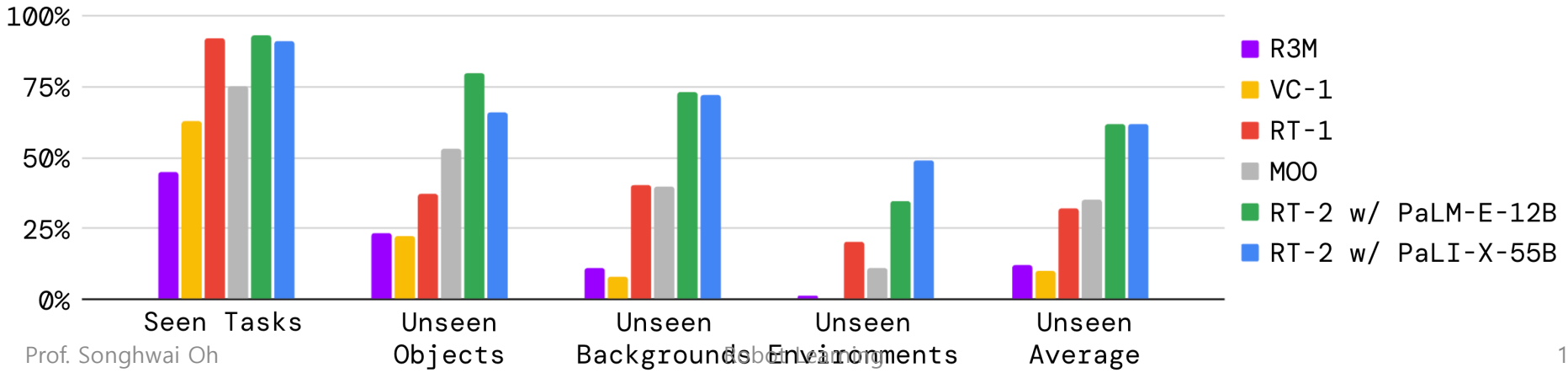
- Dataset
  - RT-1 robot dataset
  - Web-scale vision-language dataset (VQA, image captioning)


hint

RT-2: Main Experiments

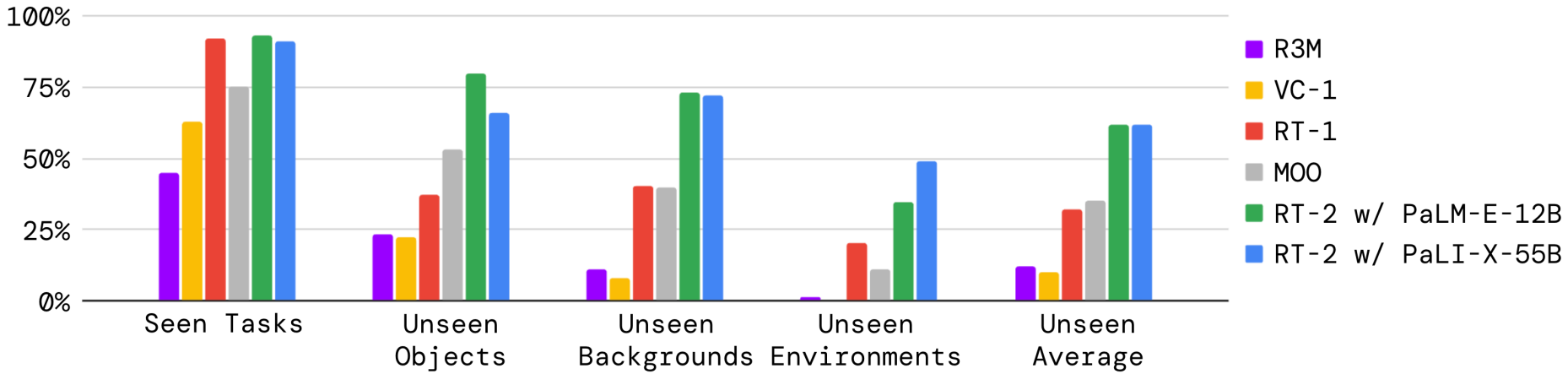
- Generalization Experiments





RT-2: Main Experiments

- Generalization Experiments



Conclusion: Pretrained VLMs with web-data co-fine-tuning improve generalization.

Open X-Embodiment: Robotic Learning Datasets and RT-X Models

A. O'Neill, et al.,
in Proc. of the IEEE International Conference on Robotics and Automation (ICRA), May, 2024.

RT-X

- Overview
  - Aggregate diverse robot embodiments to improve policy generalization

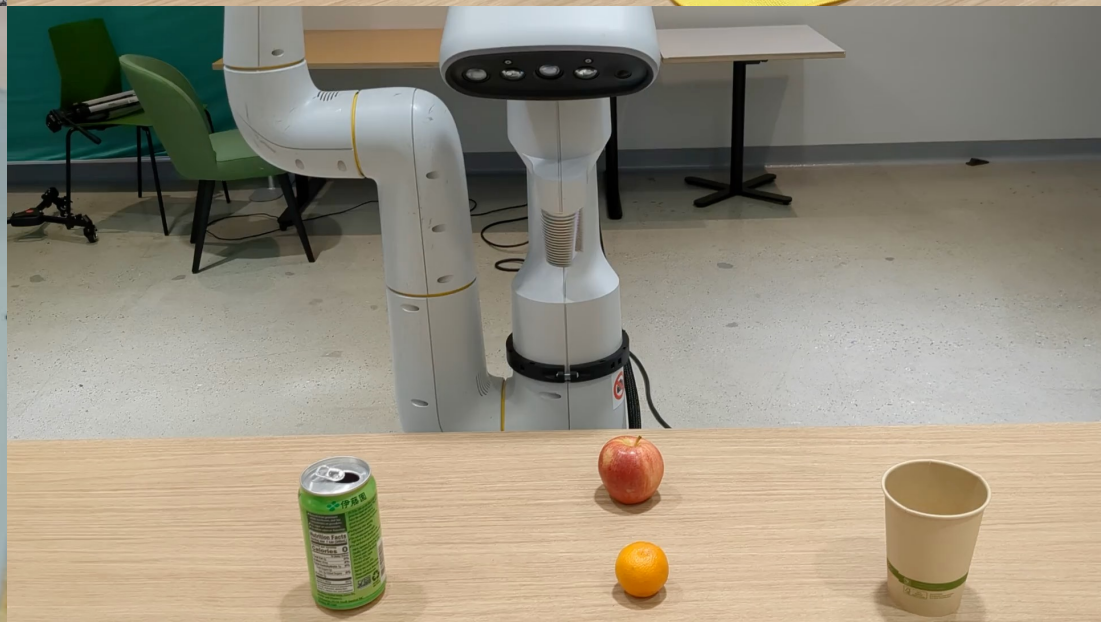
- Key Contributions
  - Construct the Open X-Embodiment (OXE) dataset by aggregating data from diverse robots, environments, and institutions
  - 1) Mitigate the embodiment gap
  - 2) Standardize heterogenous robot datasets into a unified format

RT-X: Architecture & Dataset

- Architecture
  - Reuse the RT-1 and RT-2 architectures

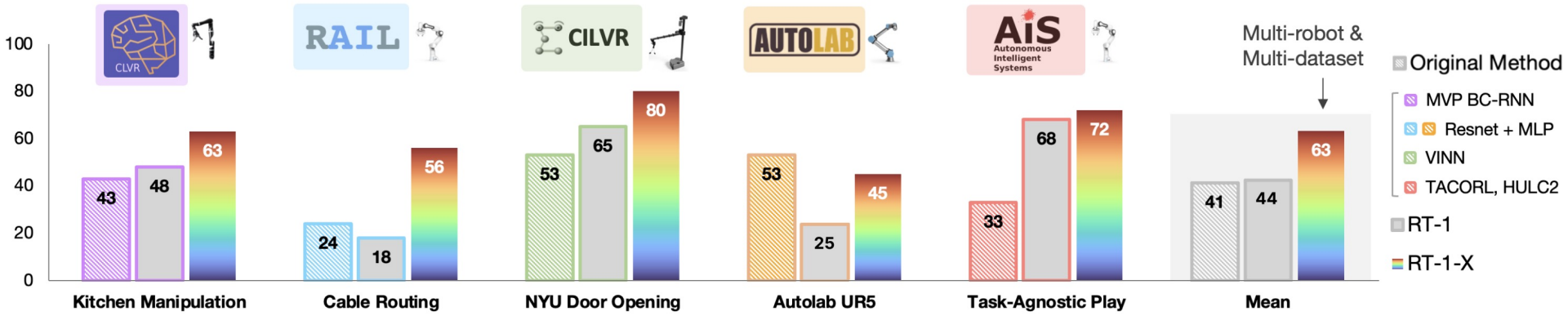
- Open X-Embodiment (OXE) Dataset
  - 1M+ trajectories
  - 22 robot embodiments
  - Aggregated from 60 heterogeneous robot datasets across multiple institutions



RT-X: Experiments

- Cross-Embodiment Transfer Experiments

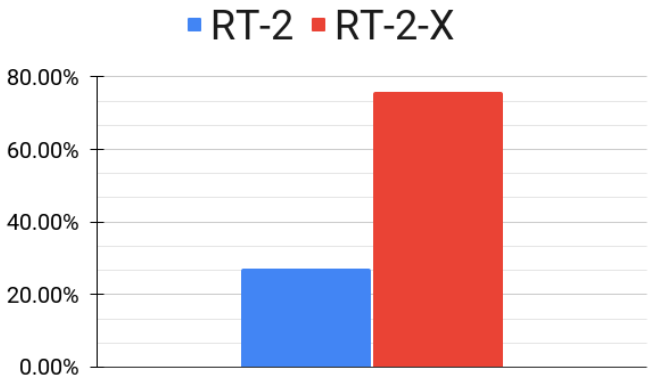


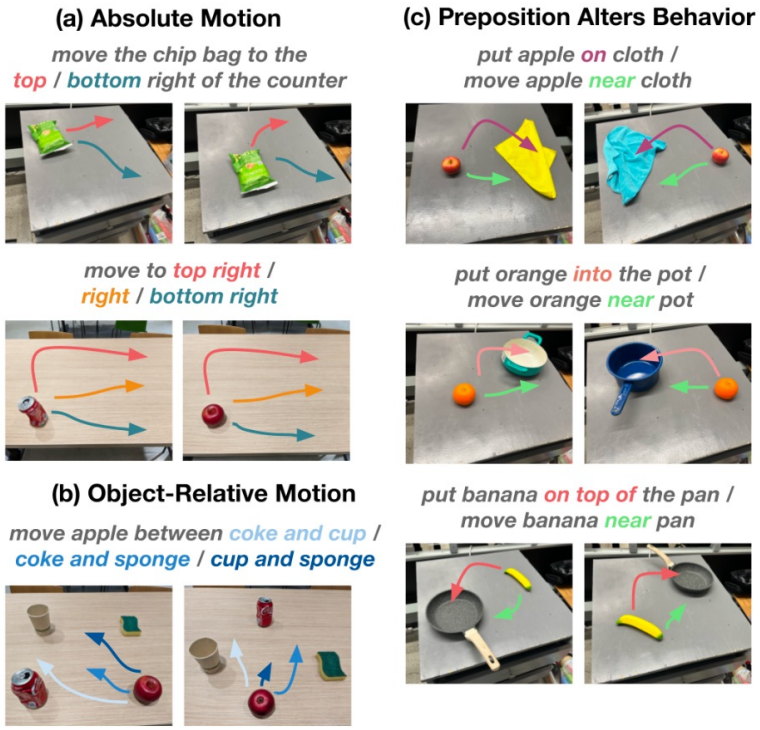
Result: RT-1-X achieves ~50% higher success rate than prior single-embodiment methods.

Conclusion: Training on multi-embodiment data enables positive transfer across robots.

RT-X: Experiments

- Generalization Experiments





Result: RT-2-X achieves ~3x better generalization compared to single-embodiment training.

Conclusion: Training on diverse robot embodiments significantly improves generalization

Octo: An Open-Source Generalist Robot Policy

D. Ghosh, et al.,
in Proc. of Robotics: Science and Systems (RSS), Jul, 2024.

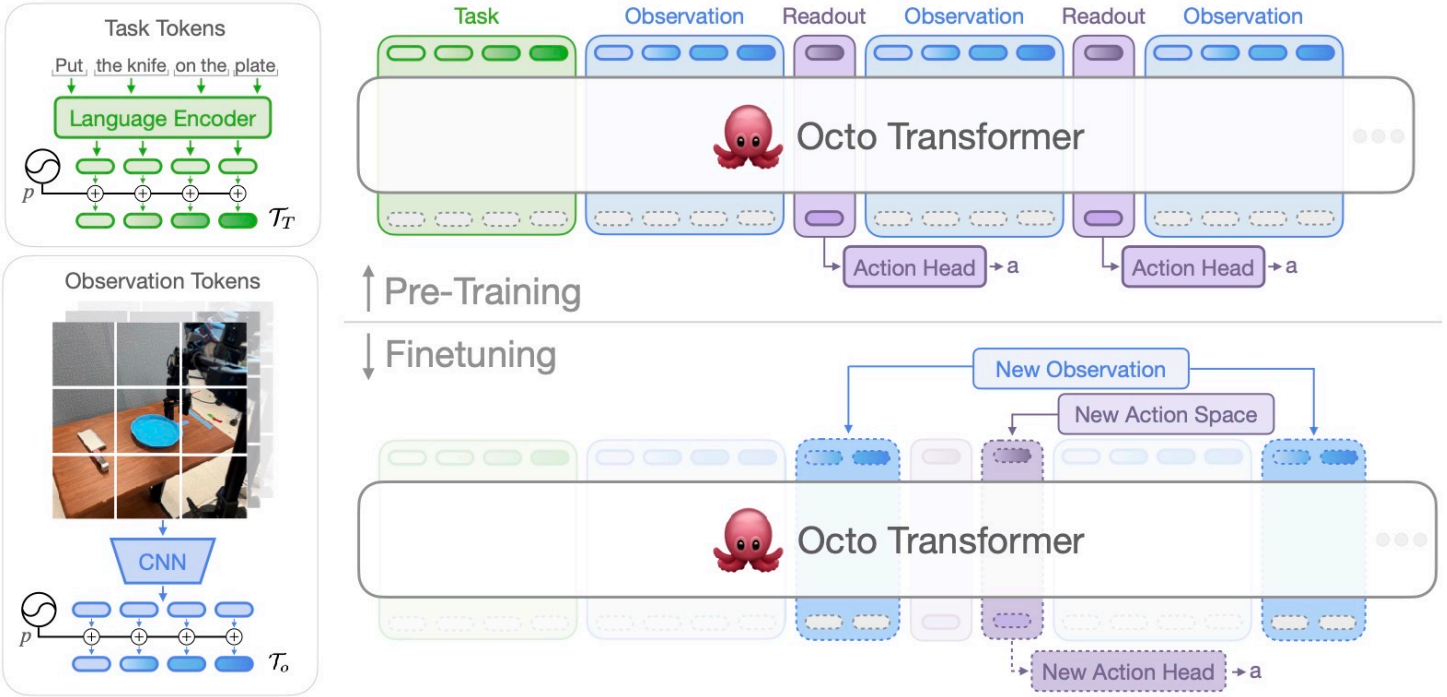
Octo

- Overview
  - Pretrain a generalist robot policy on large-scale, diverse robot data
  - Enable out-of-the-box control + efficient fine-tuning across robots, tasks, and setups.

- Key Contributions
  - Transformer-based generalist robot policy
  - Supports:
    - Flexible observations (multi-camera, proprio)
    - action spaces (joint, end-effector)
    - task specification (language, goal image)
  - Efficient fine-tuning to new robots / inputs / actions

Octo: Architecture & Dataset

- Architecture (27M or 93M parameters)
  - Token-based transformer policy



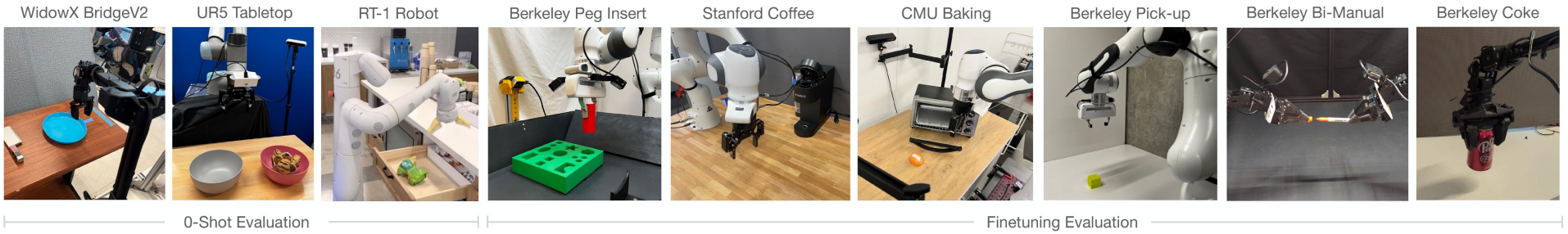
Octo: Architecture & Dataset

- Architecture (27M or 93M parameters)
  - Different modalities (language, goal image, observation) are processed by separate tokenizers.
  - Actions are generated via task-specific action heads.
  - During fine-tuning:
    - New tokenizers / heads can be added
    - And the entire model (including the transformer) is fine-tuned.

- Dataset
  - A curated subset of the Open X-Embodiment dataset
    - Octo uses ~800k robot trajectories.
    - Octo selects from 25 datasets within OXE

Octo: Experiments

- Zero-Shot & Fine-Tuning Experiments



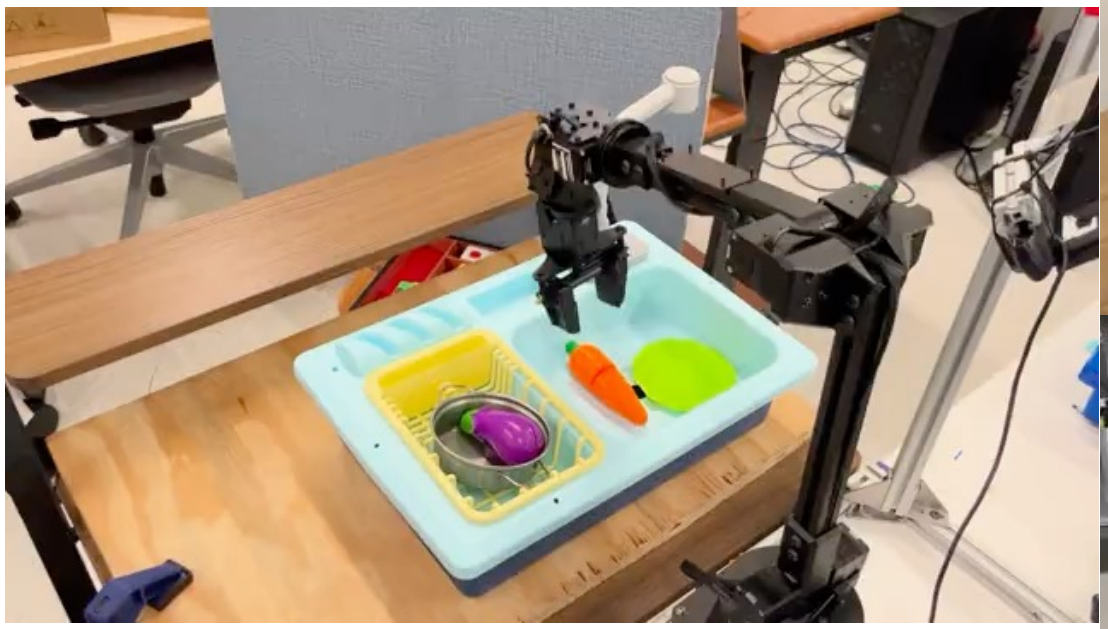
Zero-shot

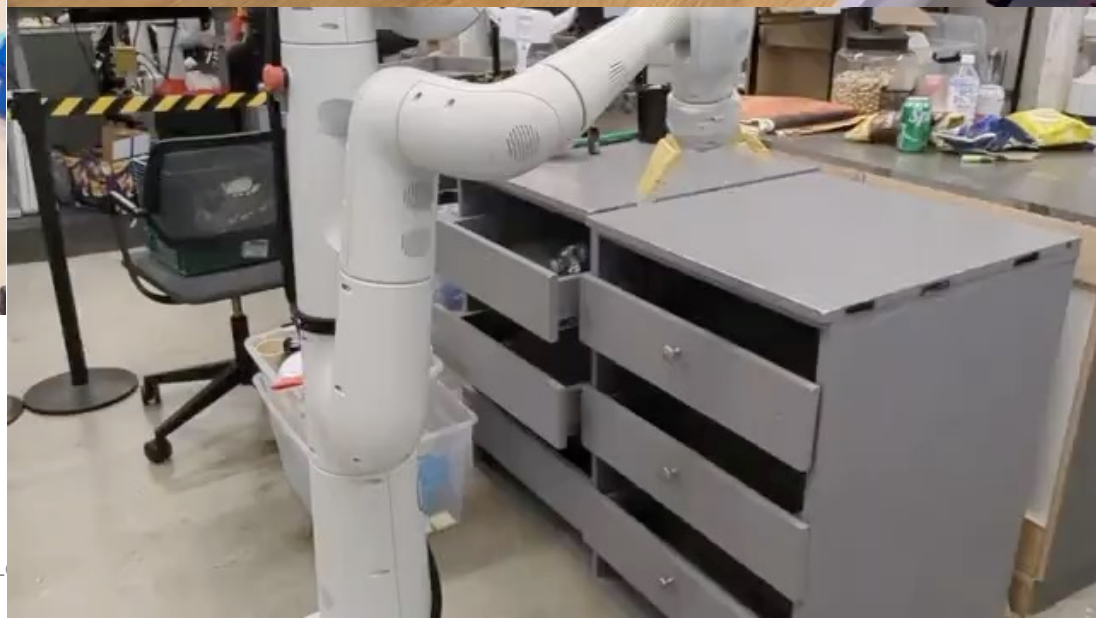
	WidowX	UR5	RT-1 Robot
RT-1-X	0.20	0.35	0.60
RT-2-X	**0.50**	—	**0.85**
Octo 🐙	**0.50**	**0.70**	0.80

Finetuning

	CMU Baking	Stanford Coffee	Berkeley Peg Insert*	Berkeley Pick-Up†	Berkeley Bimanual†	Berkeley Coke	Average
From Scratch	0.25	0.45	0.10	0.00	0.20	0.20	0.20
VC-1	0.30	0.00	0.05	0.00	0.50	0.10	0.15
Octo 🐙	**0.50**	**0.75**	**0.70**	**0.60**	**0.80**	**1.00**	**0.72**

*New observation input (force-torque proprioception) †New action space (joint position control)





Octo: Experiments

- Zero-Shot Experiments
  - **Result**: Octo outperforms RT-1-X by ~29% success rate & performs comparably to RT-2-X (much larger model).
  - **Conclusion**: Large-scale pretraining enables strong zero-shot multi-robot control.

- Fine-Tuning Experiments
  - **Result**: Octo outperforms baselines by ~52% on average
  - **Conclusion**: Octo provides a strong initialization for efficient fine-tuning.

OpenVLA: An Open-Source Vision-Language-Action Model

M. Kim, et al.,
in Proc. of the Conference on Robot Learning (CoRL), Nov, 2024.

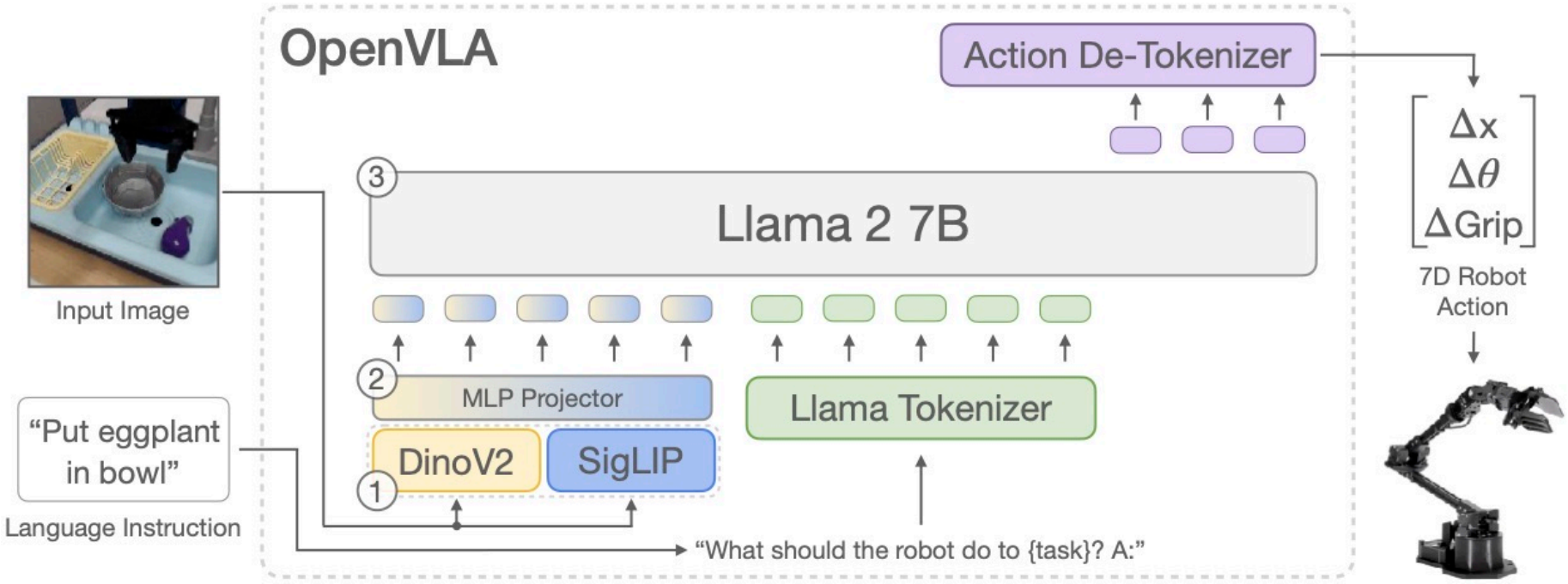
OpenVLA

- Overview
  - OpenVLA is a 7B open-source VLA model
  - It fine-tunes pretrained VLMs for end-to-end robotic control.

- Key Contributions
  - Open-source 7B VLA model (model, code, data)
  - VLM-based policy with actions represented as tokens
  - Trained on ~970k trajectories from OXE
  - Supports efficient fine-tuning (e.g., LoRA) on consumer GPUs

OpenVLA: Architecture & Dataset

- Architecture



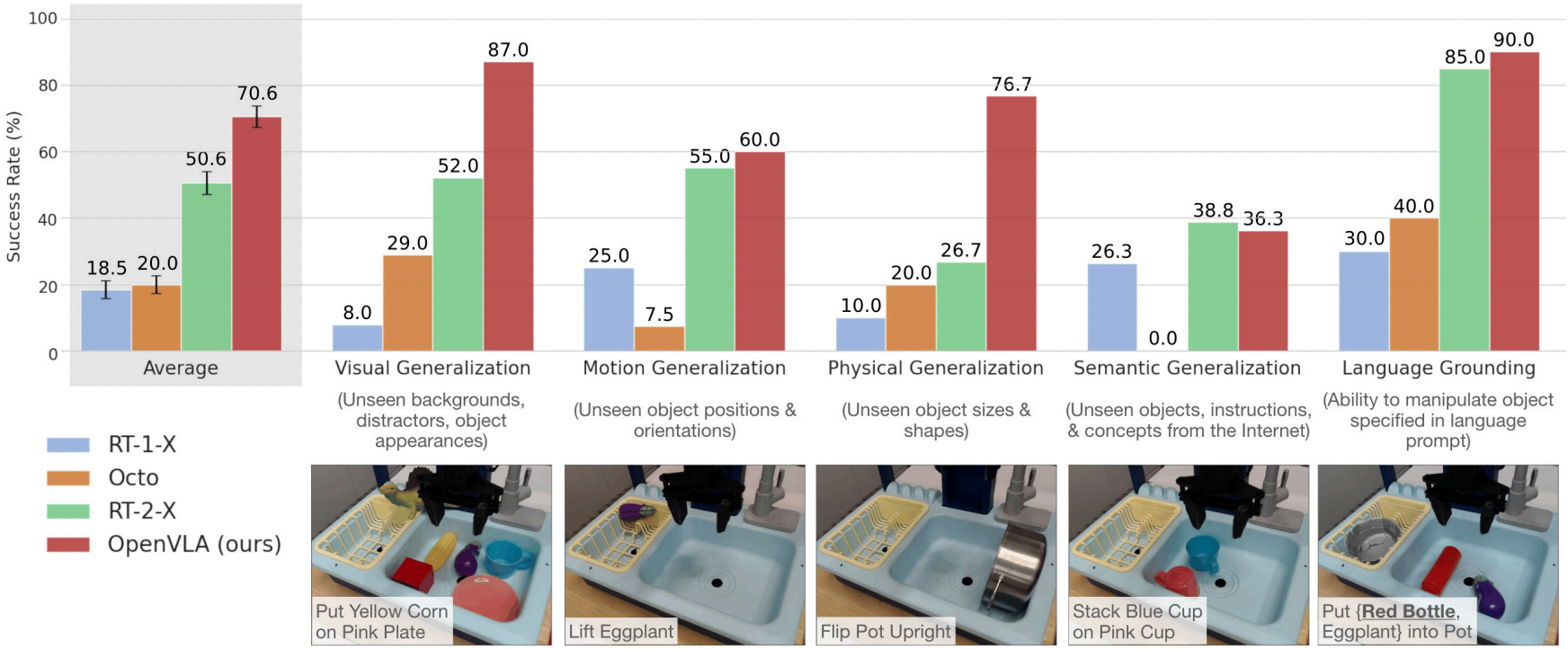
OpenVLA: Architecture & Dataset

- Architecture
  - Backbone: Pretrained prismatic VLM (7B)
    - Vision encoder: DINOv2 + SigLIP → 2-layer MLP projector
    - Language encoder: LLaMA-2 tokenizer
    - Action prediction: LLaMA-2
  - LLaMA-2 predicts discretized action tokens (corresponding to a single-step action) conditioned on vision-language tokens.

- Dataset
  - A curated subset of the OXE dataset
    - ~970k trajectories

OpenVLA: Main Experiments

- Direct Evaluations on Multiple Robot Platforms



OpenVLA: Main Experiments

- Direct Evaluations on Multiple Robot Platforms
  - **Result**: OpenVLA outperforms RT-1-X and Octo, and matches or exceeds RT-2-X (55B) despite being much smaller.
  - **Conclusion**: OpenVLA substantially improves performance and generalization while using fewer parameters.

CogACT: A Foundational Vision-Language-Action Model for Synergizing Cognition and Action in Robotic Manipulation

Q. Li, et al.,
arXiv, 2024.

CogACT: Overview

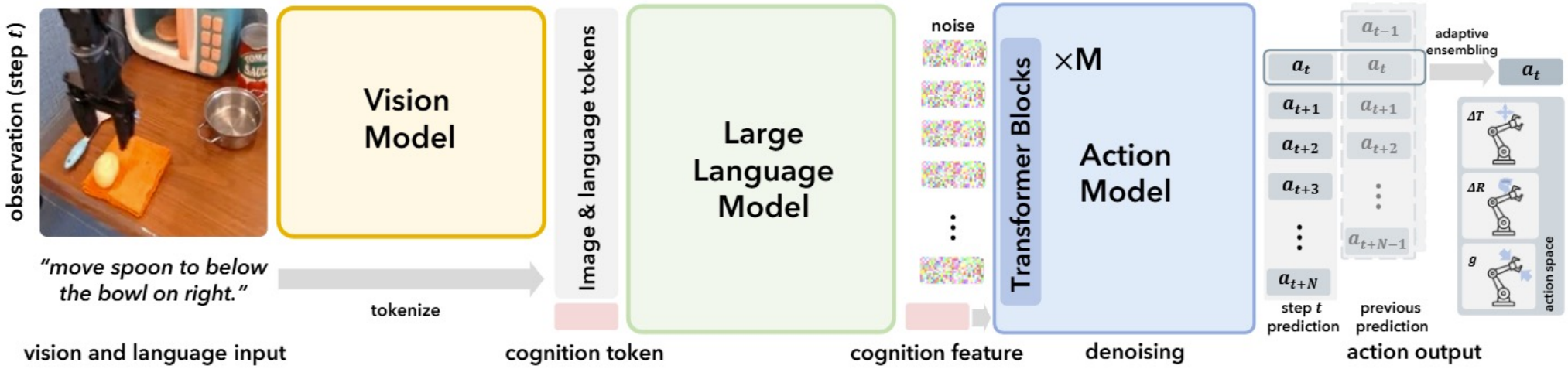
- Overview
  - CogACT proposes a componentized VLA architecture.

  ⇒ It uses VLM for semantic reasoning and diffusion-based action module for precise control.

- Kye Contributions
  - Diffusion transformer action module
    - It models continuous + multi-modal + temporal actions
  - Scaling insight
    - Action module scaling → Strong performance gains
  - Adaptive Action Ensemble (AAE)
    - AAE aggregates current and past action predictions using similarity-based weighting to produce the final action.

CogACT: Architecture & Dataset

- Architecture (7B VLM + 300M Action Module)

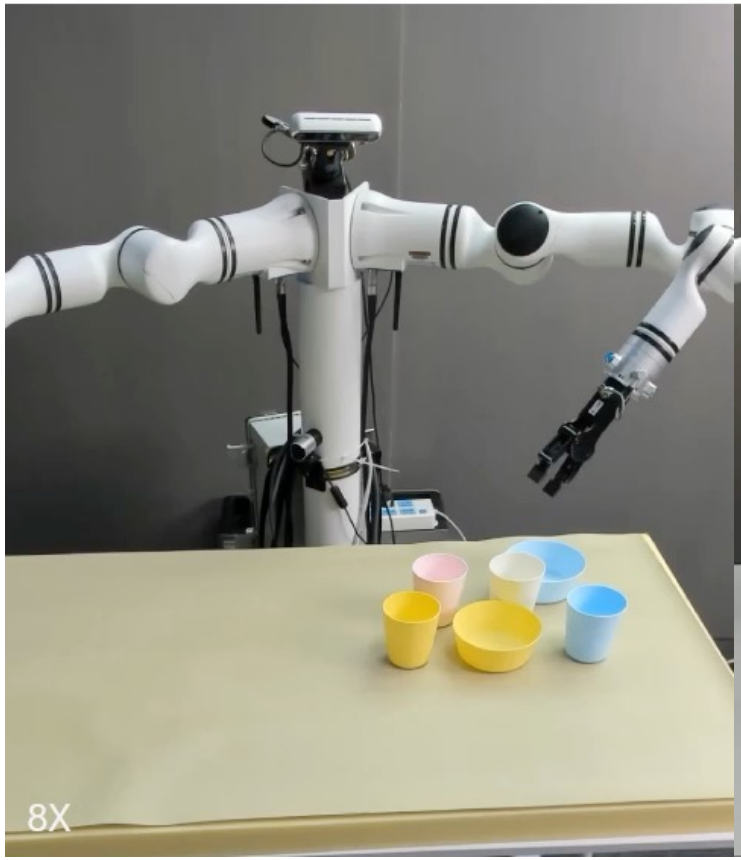


CogACT: Architecture & Dataset

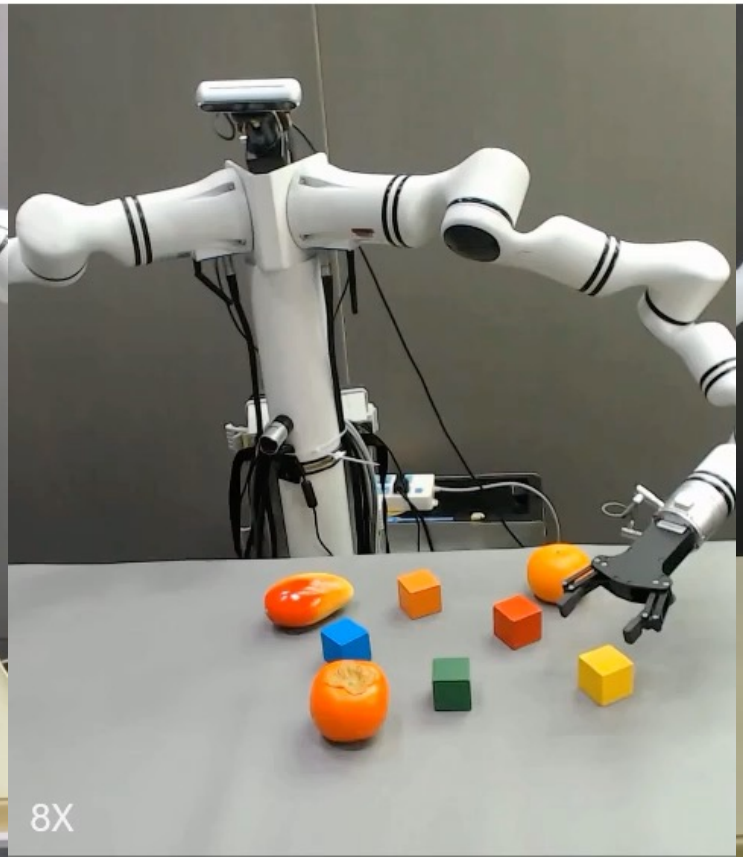
- Architecture (7B VLM + 300M Action Module)
  - Vision encoding: DINOv2 + SigLIP
  - Language encoding: LLaMA-2 tokenizer
  - Vision + language joint processing: LLaMA-2
  - Action module: diffusion transformer (DiT)

- Dataset
  - A curated subset of the OXE dataset
    - ~22.5M frames

Pick up the pink cup;
Put the pink cup into the yellow cup;
Pick up the white cup;
Put the white cup into the pink cup;
Pick up the blue cup;
Put the blue cup into the white cup.



Pick up the yellow block;
Put the yellow block on the green block;
Pick up the red block;
Put the red block on the yellow block.



Pick up the banana;
Put the banana into the white plate;
Pick up the avocado;
Put the avocado into the blue plate.

CogACT: Main Experiments

- Real-World Experiments

Method	Pick				Stack			Place			Task (All)
	Banana	Lemon	Avocado	Avg.	Cup	Bowl	Avg.	Pick	Stack	Avg.	Avg.
Octo-Base	25.0	0.0	0.0	8.3	0.0	0.0	0.0	12.5	0.0	6.3	4.9
OpenVLA	12.5	12.5	0.0	8.3	25.0	6.25	15.6	25.0	4.2	12.5	12.1
Ours	**75.0**	**50.0**	**87.5**	**70.8**	**95.8**	**68.8**	**82.3**	**87.5**	**33.3**	**60.4**	**71.2**

Result: CogACT significantly outperforms prior VLAs (e.g., +35% sim, +55% real vs OpenVLA)

Conclusion: A specialized diffusion-based action module greatly improves task performance.

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

K. Black, et al.,
in Proc. of the Conference on Robot Learning (CoRL), Sep, 2025.

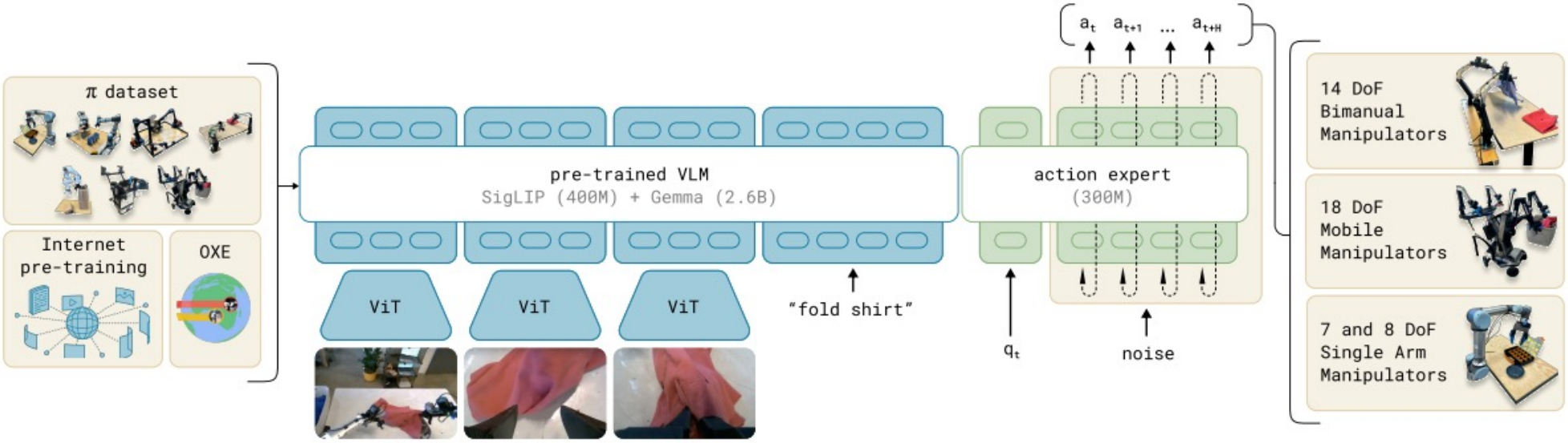
π_0

- Overview
  - π_0 is a VLA model with a **flow-matching** action generator.
  - It aims for **general + dexterous** robot control

- Key Contributions
  - Flow-matching action expert module
    → Separate from VLM backbone & Generate continuous actions
  - Action chunking + High-frequency control (~50 Hz)
  - Pretraining + Post-training recipe
    → Generalization + Task specialization

π_0: Architecture & Dataset

- Architecture



π_0: Architecture & Dataset

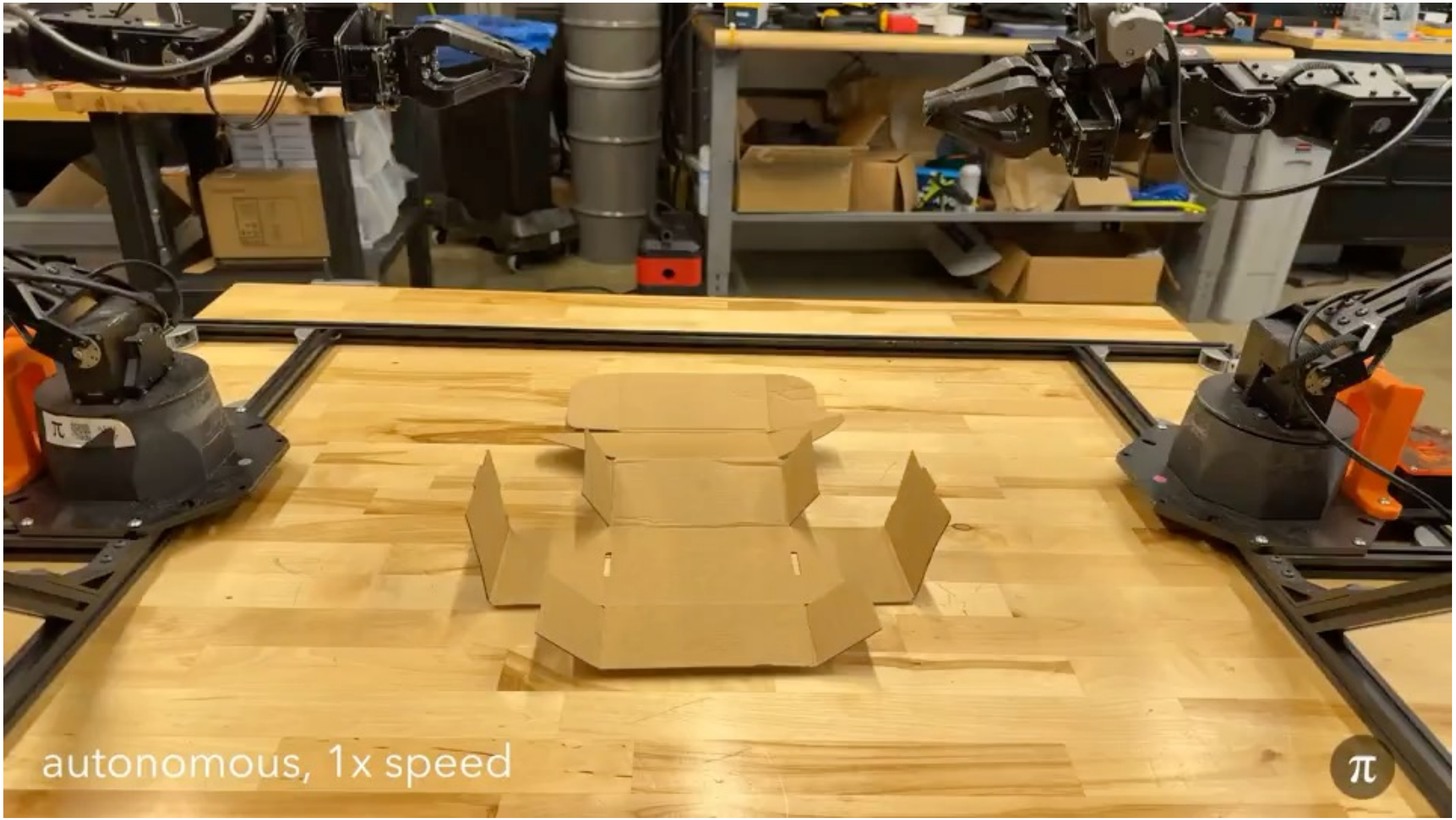
- Architecture
  - Backbone: Pretrained VLM (3B PaliGemma + 300M action expert)
  - Forward Pass:
    - Vision + language → VLM (joint processing)
    - Proprioception → Separated encoder
    - VLM features + state features → Action expert (flow matching)

- Dataset
  - Subset of OXE + Large in-house manipulation dataset
    - 10,000+ hours of robot data & 68 tasks
    - 7 robot configurations (single-arm robots, dual-arm robots, mobile manipulators)


autonomous, 2x speed

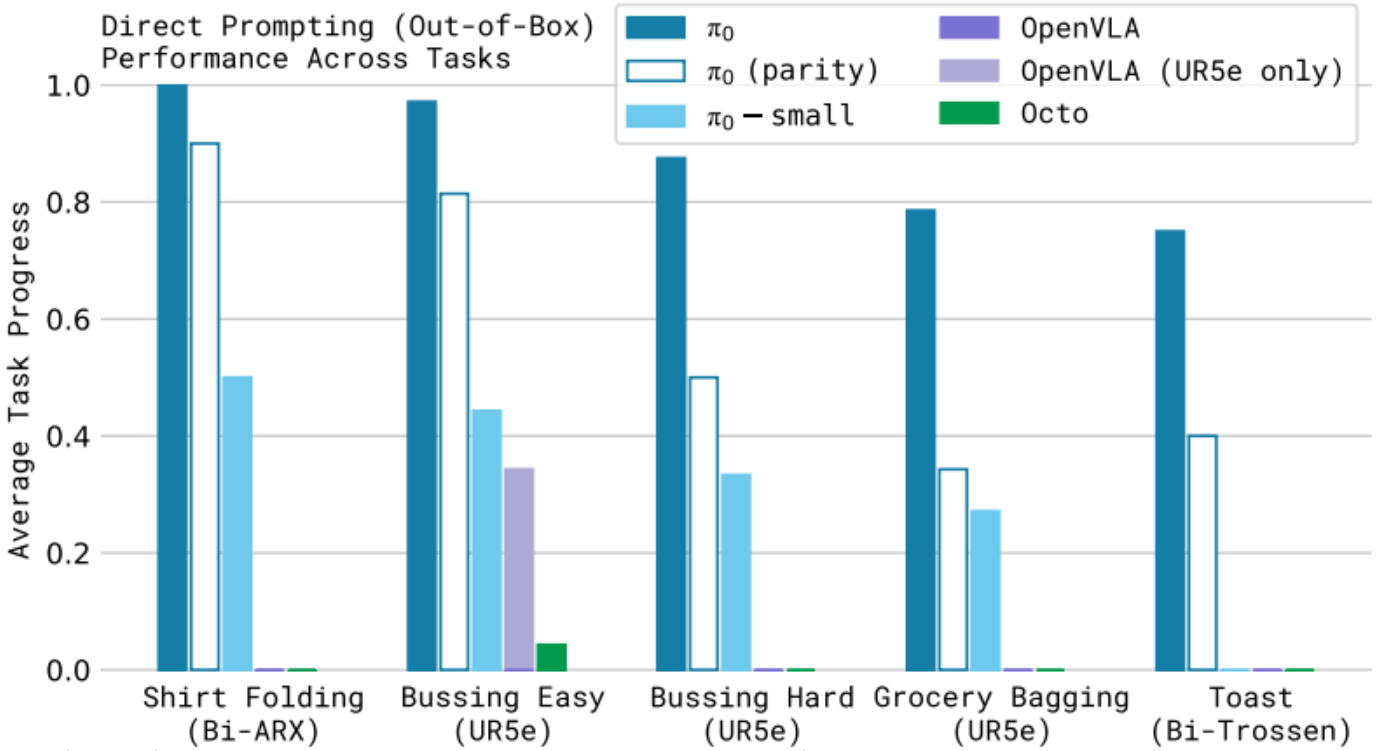

autonomous, 2x speed
π


autonomous, 2x speed
π


autonomous, 1x speed

π_0: Main Experiments

- Dexterous Manipulation Tasks across Multiple Robots



π_0: Main Experiments

- Dexterous Manipulation Tasks across Multiple Robots

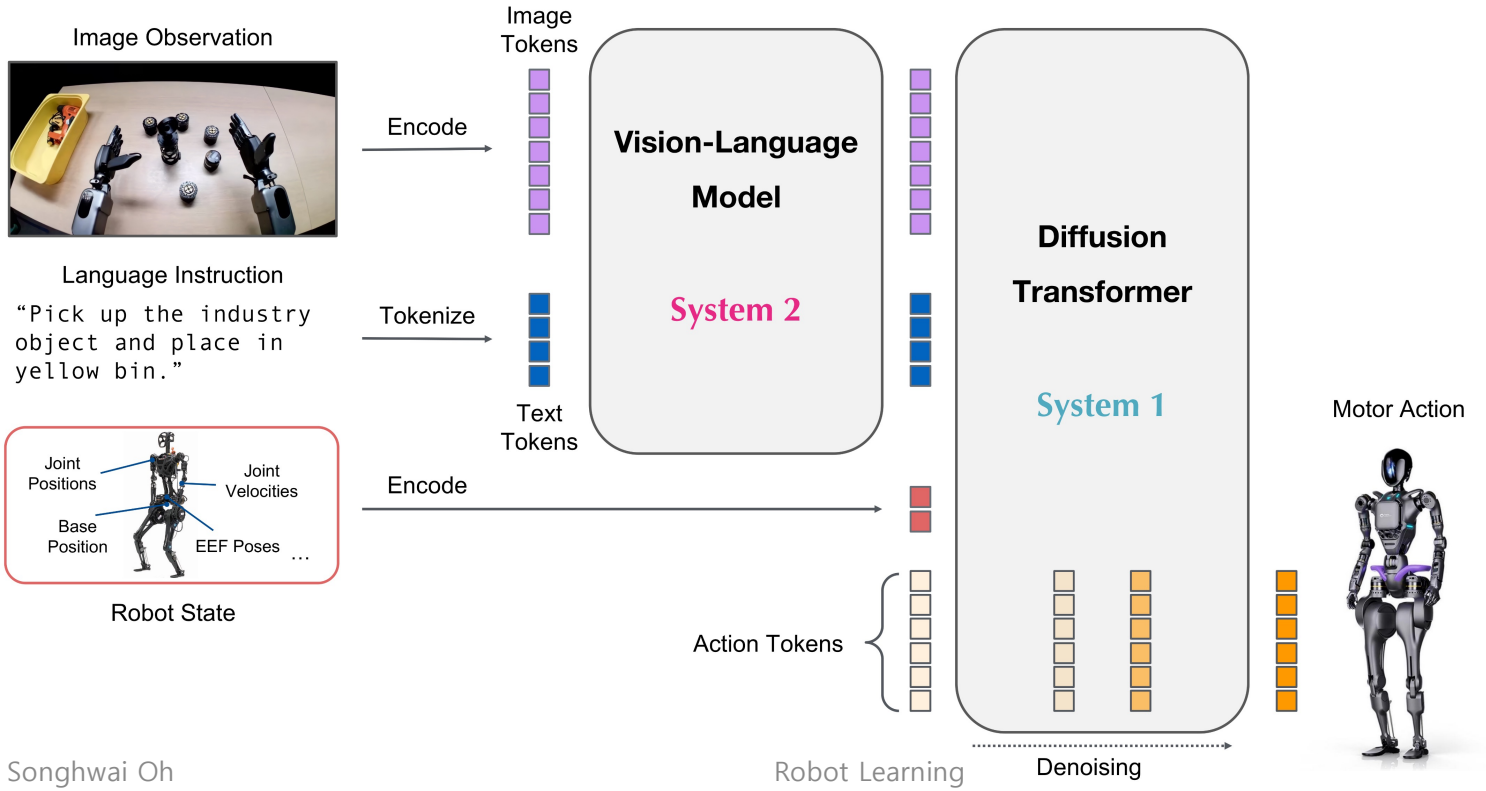
  - **Result**:
    - Significantly outperforms prior VLAs (RT-2, OpenVLA, Octo)
    - Strong gains on dexterous and long-horizon tasks
  - **Conclusion**:
    - Flow-based continuous action modeling enables precise and robust robot control.

GR00T N1: An Open Foundation Model for Generalist Humanoid Robots

NVIDIA,
arXiv, 2025.

GR00T N1: Architecture

- Architecture (1.34B VLM + ~900M action module)



GR00T N1: Architecture

- Architecture (1.34B VLM + ~900M action module)
  - Dual-system VLA
    - System 2 (Reasoning): Eagle-2
    - System 1 (Action): Diffusion transformer + Flow matching
  - Generates continuous action chunk (H=16)

- Data Pyramid
  - Base (Large scale):
    - Web data + Human videos
  - Middle:
    - Synthetic data (Simulation + Neural-generated trajectories)
  - Top (Smallest scale):
    - Real robot data

GR00T N1: Main Experiments

- Real-World Humanoid Manipulation Tasks

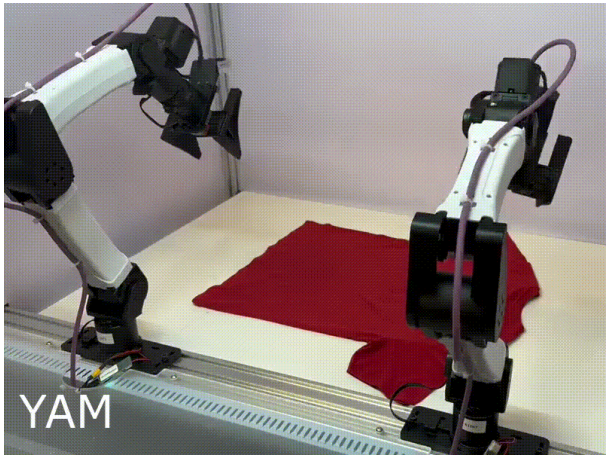
	Pick-and-Place	Articulated	Industrial	Coordination	Average
Diffusion Policy (10% Data)	3.0%	14.3%	6.7%	27.5%	10.2%
Diffusion Policy (Full Data)	36.0%	38.6%	61.0%	62.5%	46.4%
GR00T-N1-2B (10% Data)	35.0%	62.0%	31.0%	50.0%	42.6%
GR00T-N1-2B (Full Data)	**82.0%**	**70.9%**	**70.0%**	**82.5%**	**76.8%**

Result: GR00T N1 achieves strong performance across diverse real-world tasks.

Conclusion: GR00T N1 enables effective real-world humanoid manipulation with strong generalization and data efficiency.

GR00T N1: Main Experiments

- Experiment Videos



CoT-VLA: Visual Chain-of-Thought Reasoning for Vision-Language-Action Models

Q. Zhao, et al.,
in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun, 2025.

CoT-VLA

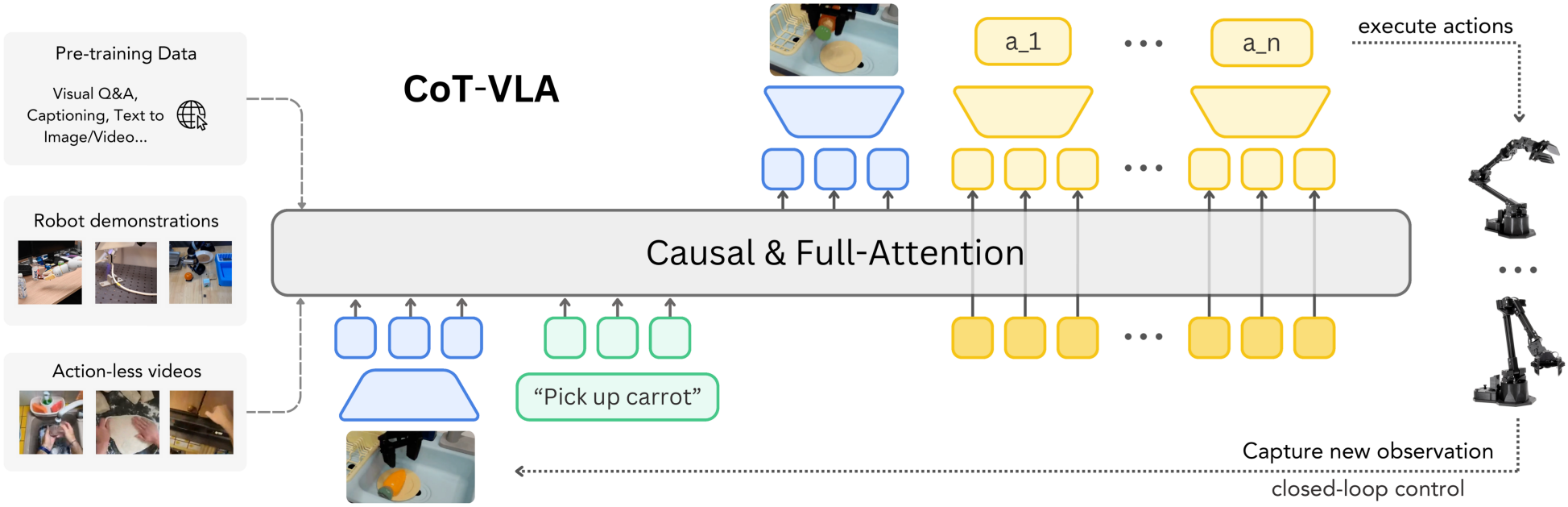
- Overview
  - CoT-VLA introduces visual chain-of-thought (CoT) reasoning into VLA models.
  - CoT-VLA first generates a subgoal image and then predicts actions to achieve it.

- Key Idea
  - CoT-VLA uses future images (subgoals) as intermediate reasoning steps.

  → Enabling explicit visual planning before action generation.

CoT-VLA: Architecture & Dataset

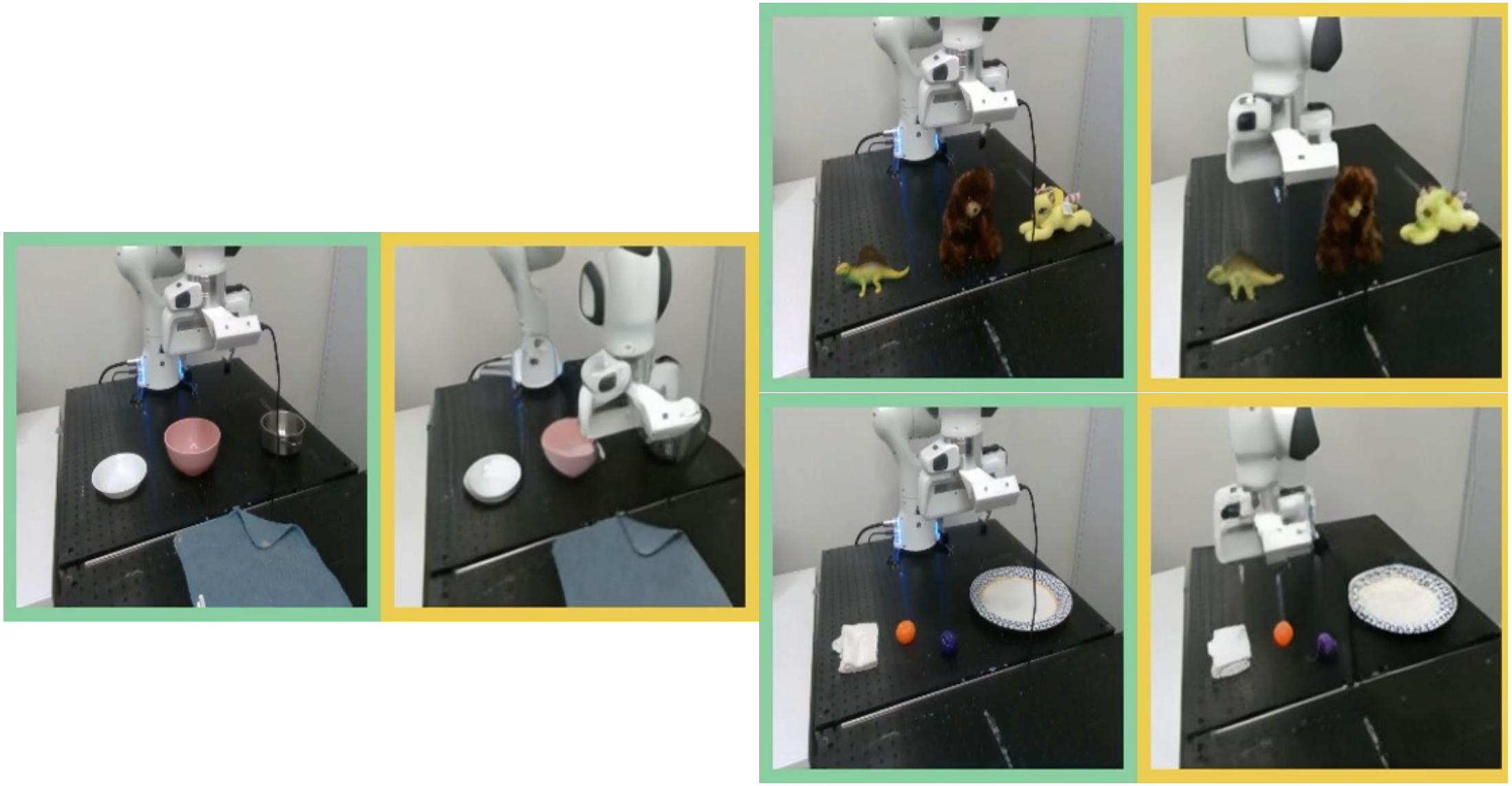
- Architecture



CoT-VLA: Architecture & Dataset

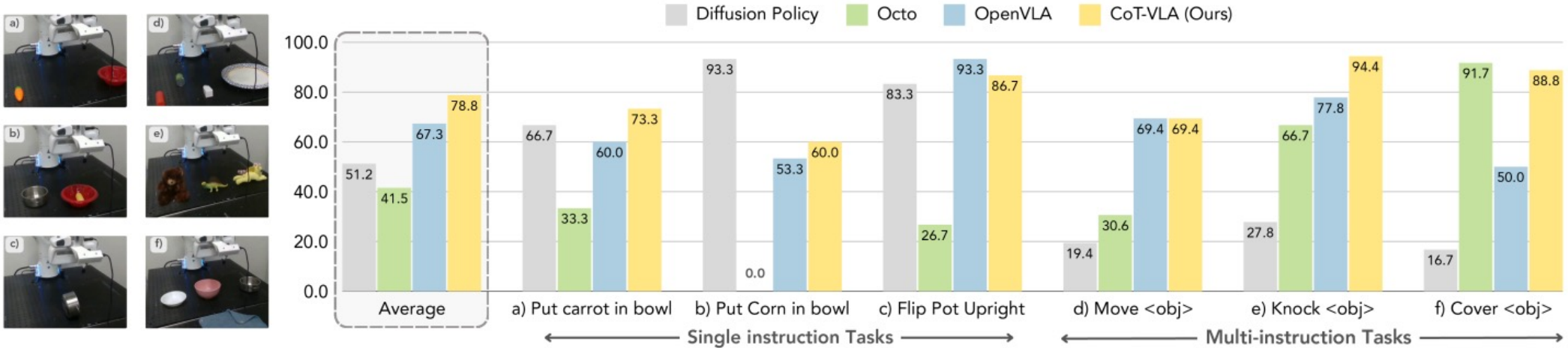
- Architecture
  - Backbone: VILA-U (7B VLM)
  - Forward Pass
    - Step 1: Visual CoT
      - CoT-VLA generates a subgoal image using causal attention.
    - Step 2: Action Generation
      - Input: Current image + Subgoal image + Language
      - Output: Discrete action sequence (Action chunk)
      - Attention: Full attention for action tokens

- Dataset
  - Robot demonstrations: OXE dataset
  - Action-less videos: EPIC-KITCHENS, Something-Something V2



CoT-VLA: Main Experiments

- Franka-Tabletop Benchmark



- **Result**: CoT-VLA achieves the best average performance across all tasks.
- **Conclusion**: Visual CoT reasoning improves performance, especially for complex multi-step tasks.

What matters in Building Vision-Language-Action Models for Generalist Robots

X. Li, et al.,
Nature Machine Intelligence, vol. 8, pp.158-172, 2026.

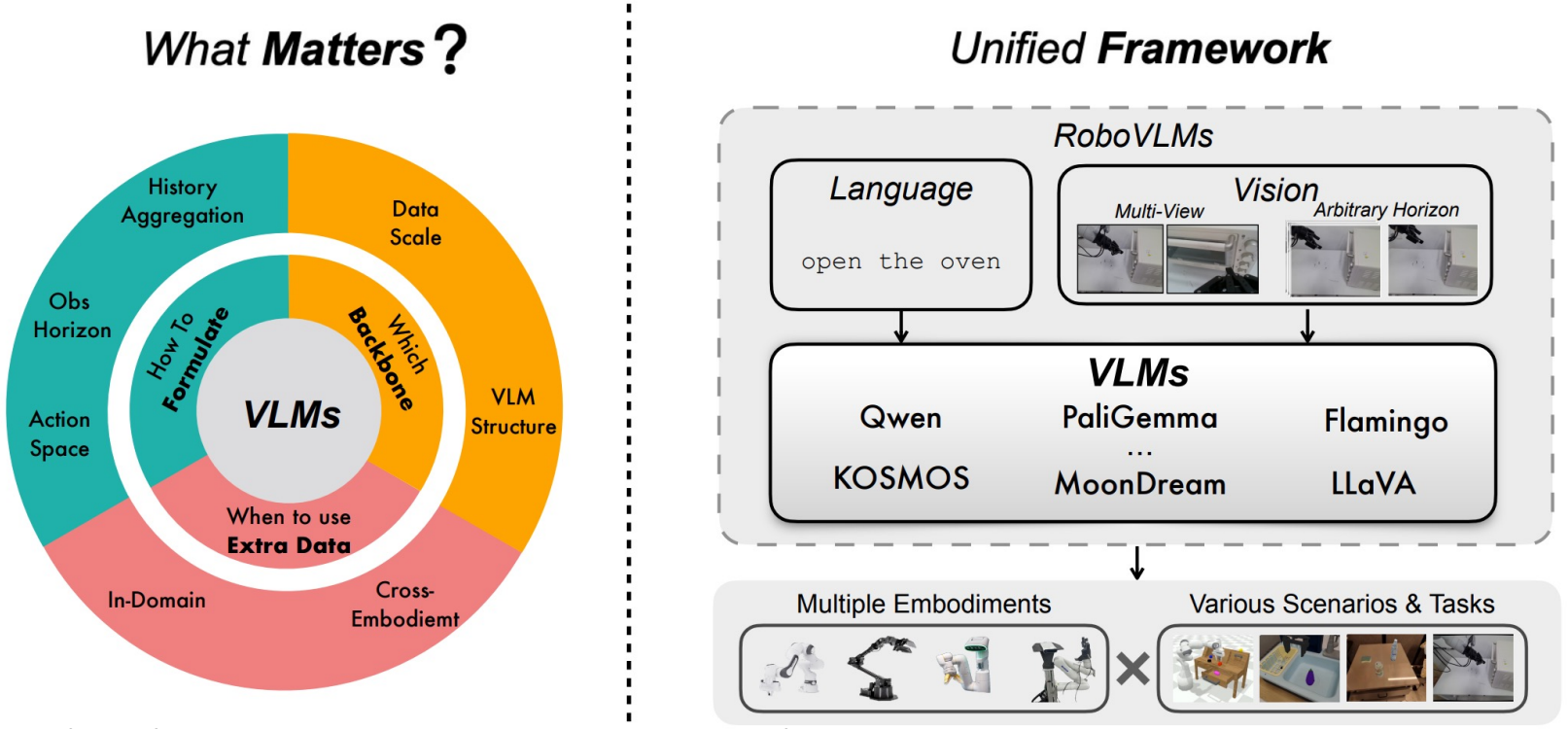
RoboVLMs

- Overview
  - Study what matters in building VLA models for generalist robots
  - Systematically analyze:
    - VLM backbone
    - VLA architecture
    - Robot data (Scale & Mixture)

- Key Contributions
  - Systematic study of VLA design space
    - 8+ VLM backbones & 4 policy architecture & 600+ experiments
  - Identify key factors for VLA performance
  - Propose RoboVLMs framework
    - Flexible VLA design & Easy integration of different components

RoboVLMs: Key Design Choices

- Flexible VLA Framework (Modular Design)



RoboVLMs: Key Design Choices

- Flexible VLA Framework (Modular Design)
  - How to formulate the VLA
    - Action space (Continuous vs Discrete)
    - Observation horizon (One-step vs History)
    - History aggregation (Interleaved vs Policy head)
  - Which VLM backbone to use
    - Flamingo / LLaVA / Qwen-VL / MoonDream / Uform / KosMos / PaliGemma
  - When / how to use data
    - Cross-embodiment data
    - In-domain data
    - Pretraining vs Post-training

RoboVLMs: Key Findings

- VLM backbone significantly affects performance.
  - KosMos and PaliGemma outperform other backbones.
- Continuous actions outperform autoregressive discrete actions.
  - Continuous formulations show consistently better results.
- Policy-head formulation performs best among tested structures.
  - Maintaining the original VLM processing with a separate policy head is most effective.
- Incorporating historical context improves performance.
  - History-aware models outperforms one-step formulations.

RoboVLMs: Key Findings

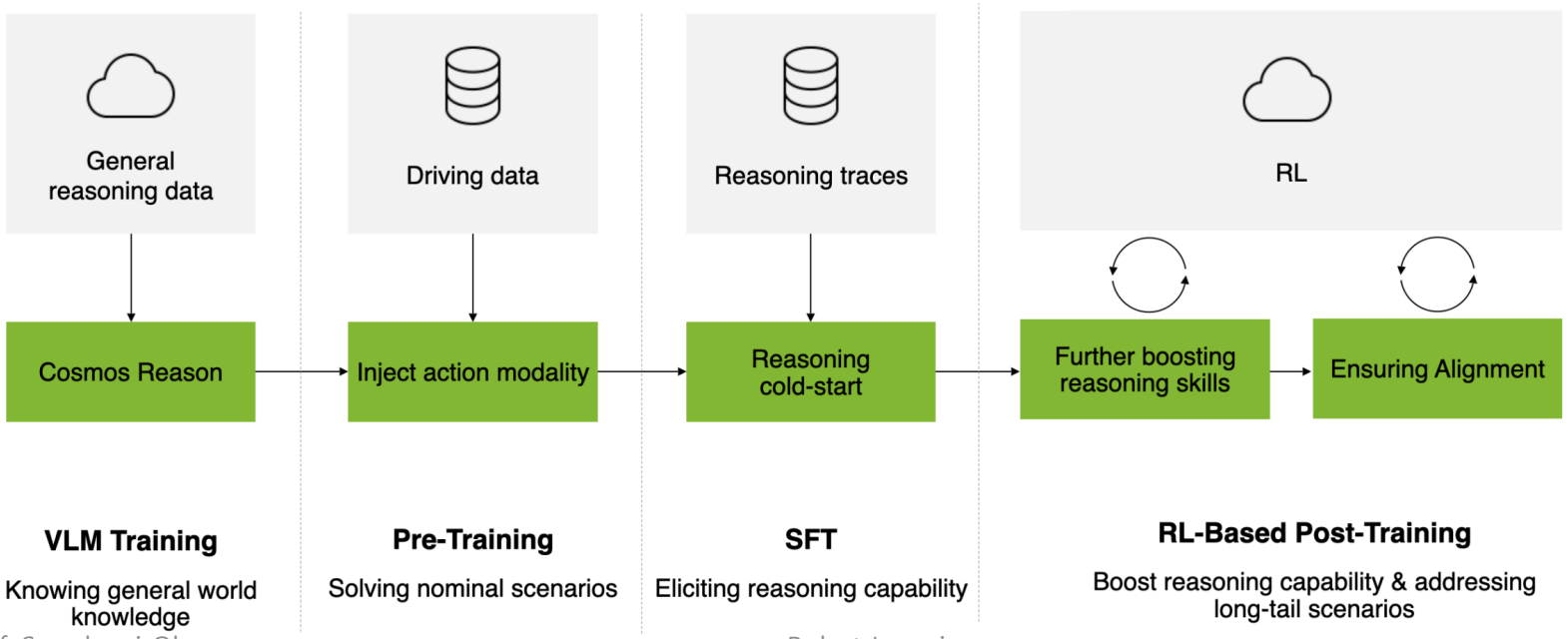
- Cross-embodiment pretraining alone does not consistently improve performance.
  - Post-training (fine-tuning on target data) leads to better results.
- In-domain data provides clear performance gains.
  - Data from the same robot or task improves performance.
- Larger VLMs improves data efficiency.
  - Larger models achieve better performance with less data.

Alpamayo-R1: Bridging Reasoning and Action Prediction for Generalization Autonomous Driving in the Long Tail

NVIDIA,
arXiv, 2026.

Alpamayo-R1

- Overview
  - Alpamayo-R1 is a VLA that integrates causal reasoning with trajectory prediction for robust autonomous driving.



Alpamayo-R1

- Key Contributions:
  - Chain of Causation (CoC) dataset
    - CoC data uses **a hybrid pipeline** with auto-labeling and human verification.
    - It selects **key decision moments** and annotates **causes** and **explicit driving decisions**.
  - Modular reasoning-action VLA architecture
    - Alpamayo-R1 uses **Cosmos-Reason (VLM)** to generate **reasoning traces** and conditions action prediction on them.
    - It generates low-level actions via **a diffusion-based trajectory decoder** using flow-matching.
  - Training strategy (SFT + RL)
    - Alpamayo-R1 uses supervised fine-tuning to learn reasoning from CoC data.
    - If further applies reinforcement learning to improve reasoning quality and enforce reasoning-action consistency.

Alpamayo-R1

- Examples of CoC Reasoning Traces
  - Green: Driving decision / Blue: Critical component

Handling high uncertainty ODD

Nudge right within lane to keep safe clearance to the **oncoming vehicle** in the opposite lane due to **limited visibility at night**.

Optimizing mission-level progress

Change lane to the left to avoid the **bike** ahead and stop at the **pedestrian crossing** for the **red traffic light**.


Reasoning
Nudge left to
Trajectory Prediction
History
Prediction
Ego Vehicle
Longitudinal (m)
Lateral (m)
80
60
40
20
0
-20
-10
0
10
20

Alpamayo-R1: Main Experiments

- Closed-Loop and Open-Loop Evaluation on Long-Tail, Safety-Critical Driving Scenarios
  - **Result**: Alpamayo-R1 improves planning accuracy (+12%) and reduces close encounters (-35%) compared to trajectory-only baselines.
  - **Conclusion**: Causally grounded reasoning significantly enhances robustness and safety in challenging driving scenarios.