RT-2 → MolmoAct2

CoRL 2023 → arXiv preprint

2026

Google DeepMind → **Ai2 & University of Washington**

Georgia Tech

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

[Brohan, Anthony, et al.]
[CoRL 28 July 2023]

[Nadira Mahamane, Josh Traynelis, Zihao Zhao]

Georgia Tech

Outline

- **Part 1: Seminal Paper (RT-2)**
  - Problem Statement
  - Related Works
  - Approach
  - Experiments & Results
- **Part 2: Frontier Paper (MOLMO-ACT2)**
  - Problem Statement
  - Contributions
  - Related Works
  - Approach
  - Experiments & Results
  - What persisted and what changed?
  - Limitations and Implications
  - Discussion Questions

Georgia Tech

Problem Statement: RT-2

- Robot data is currently limited. VLMs output text, while having significant amounts of training data available
- Previous methods such as RT-1 don't perform strongly on unseen objects

Can large pretrained VLMs be integrated into low-level robotic control to improve generalization and also enable emergent semantic reasoning?

RT-2 Contributions

- Robot actions are written as text tokens in the model's output.
- During inference, the text tokens are de-tokenized into robot actions, enabling closed loop control.
- For RT-2, VLM is **co-fine-tuned** on both internet scale Vision Language datasets, including vision question answering and robot trajectories
- Explores **CoT reasoning**

Georgia Tech

Related Works for RT-2

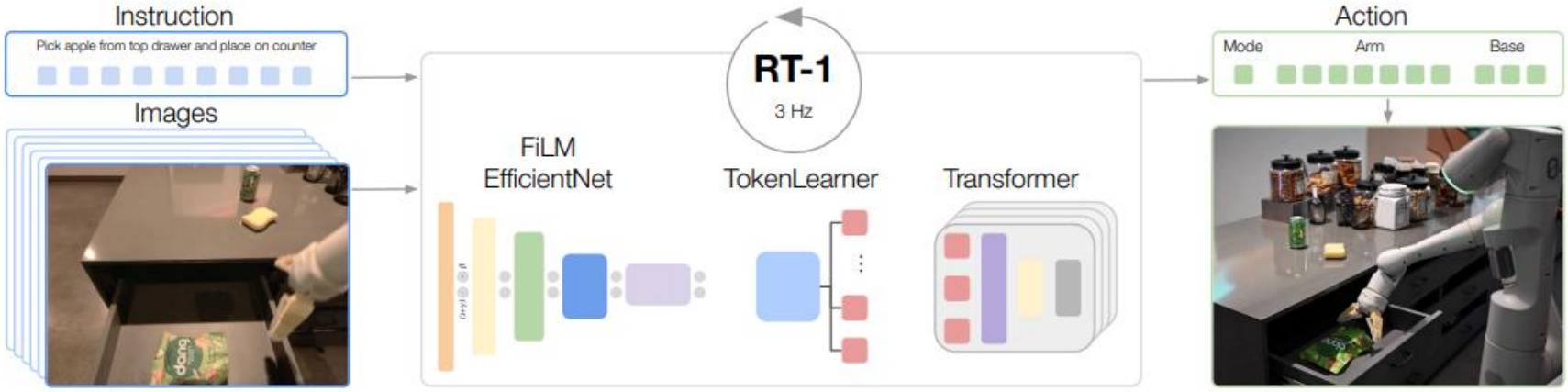
Category	Method/Paper	What it does	Drawbacks
Vision Language Models	Flamingo, Pali-X, Palm-E	VLM models, image + text → text	Output language, not robot actions
Generalization for Robot Data Training	RT1	Provides larger Robotic datasets	Trained on robotic data, without web data based pre-training
Pre-trained VLMs in Robotics	MOO, CLIPort	Integrate VLMs into end-to-end policies	Add structure but less generalizable

RT-2: fine-tunes a VLM end-to-end model that completes both understanding and action, outputting action tokens, distinct from prior works which separate these steps

Georgia Tech

Overview RT-1 and Gap

RT-1: Performs poorly on unseen objects



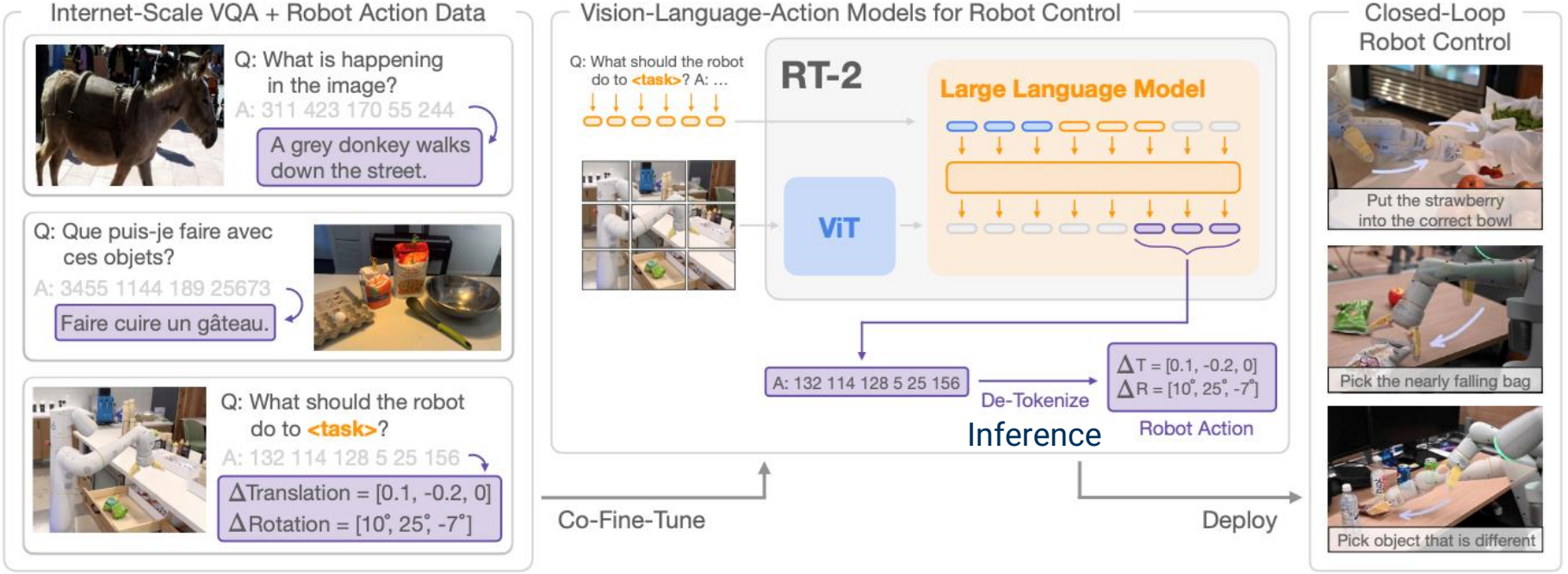
Gap: Robots could act but not understand. VLMs could understand but not act

RT-1: Learns from robotic data but not web-scale vision language data

Georgia Tech

RT2 Overview

Robot Actions → Text tokens



Delta position and Delta rotation

Co-fine-tune on both web and robotics data

Georgia Tech

Approach: Pre-trained VLMs PaLI-X/PaLM-E

PaLI-X (5B and 55B)

- ViT-22B Image Encoder and 32B Encoder-decoder language model
- Trained on multi-lingual web-image text data
- PaLI-X supports vision-language reasoning

PaLM-E

- Decoder only based PALM LLM
- Focused on embodied reasoning.
- RT-2 uses 12B version
- PaLM-E for linking perception to physical action

Challenge: weights and data are closed-source, motivating approaches such as OpenVLA and MolmoAct2

RT-2 uses pre-trained VLMs PaLI-X and PaLME

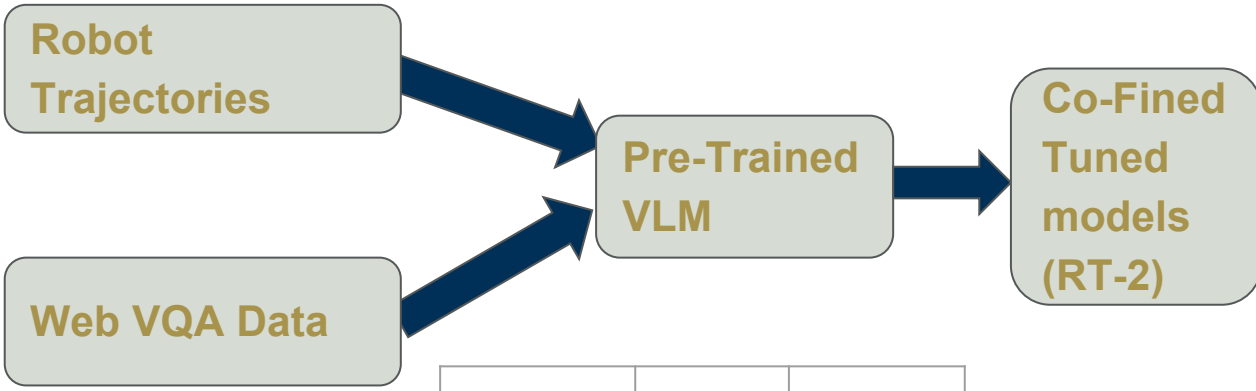
Approach: RT-2 Co-Fine-Tuning

- **Co-fine-tuning** the model demonstrates stronger generalized performance compared to robot-only fine-tuning the model.

- Pre-trained VLM Model Backbones:
  - PaLI-X 5B / 55B
  - PaLM-E 12B

- **Co-fine-tuned models**:
  - RT-2-PaLI-X: 5B, 55B
  - RT-2-PaLM-E: 12B
  - Compared against baselines

Why: Fine tuning only on robot data leads to forgetfulness for web based data

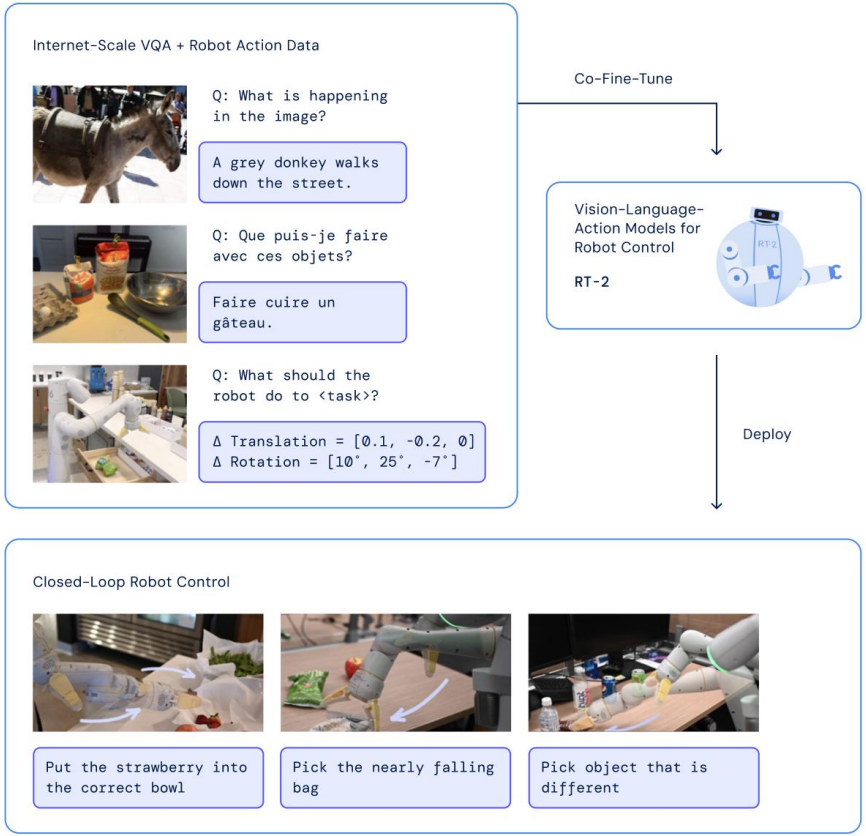


	Robot Only Fine-tuning	Co-fine-tuning
5B	42%	44%
55B	52%	63%

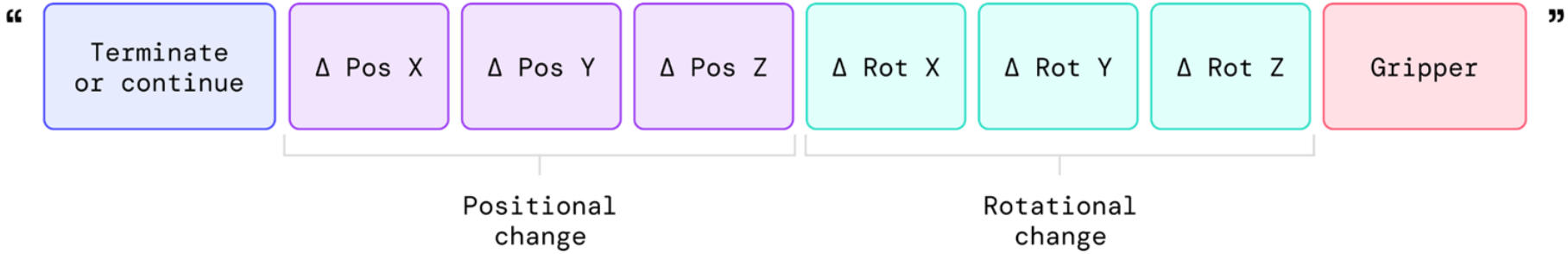
Georgia Tech

Approach: Web VQA and Robot Data

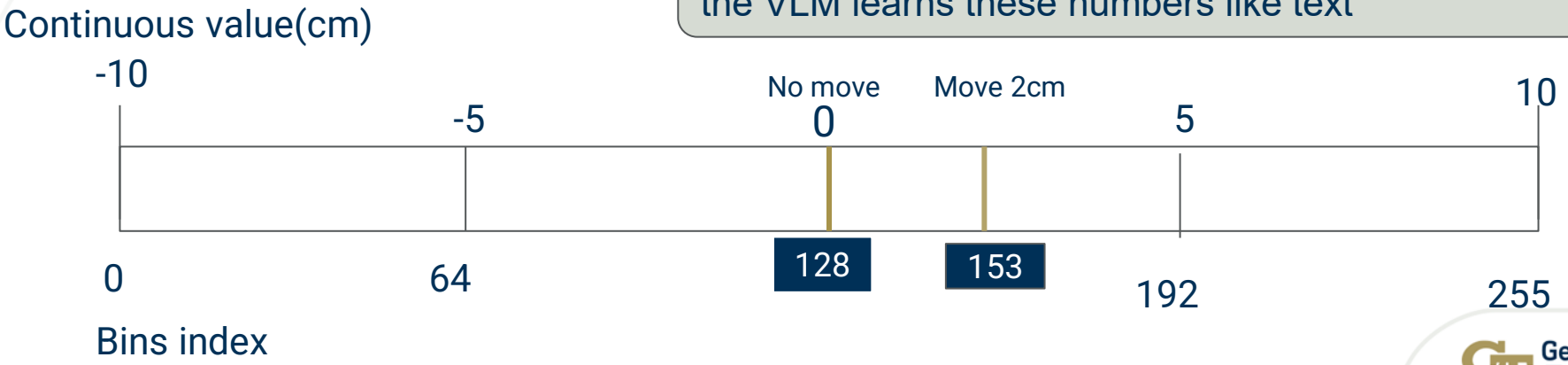
RT-2: Web VQA and Robot data are put into the same format, so a robot action represents another answer for the model



Approach: Action Representation

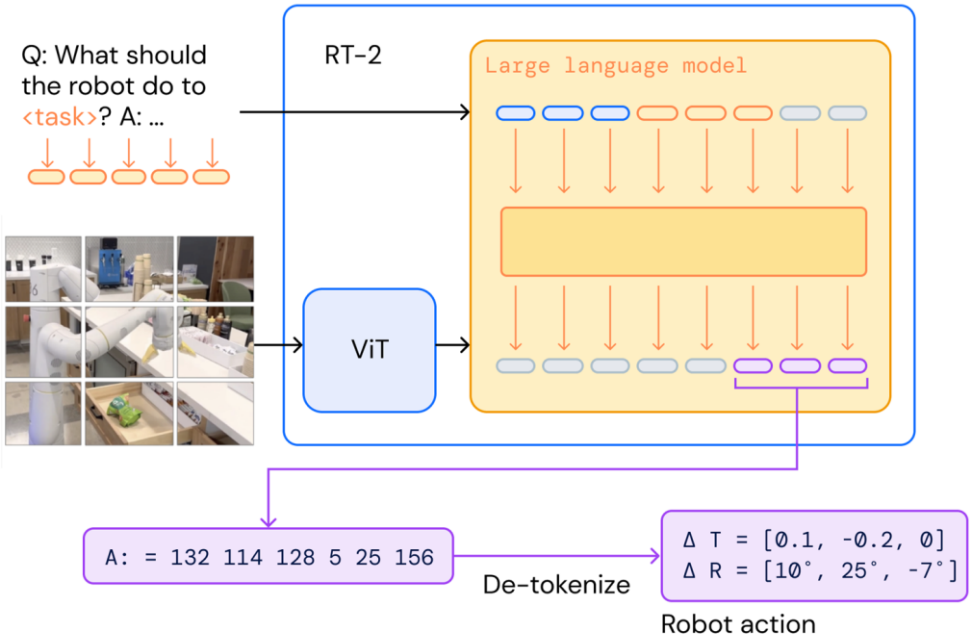


Takeaway: Action is written as eight numbers, and the VLM learns these numbers like text



Georgia Tech

Approach: Real Time Inference



- Deploy RT2 models in a multi TPU cloud service
- Query the service over the network
- Serve multiple robots over the same cloud
- Frequency : 1-3Hz -> RT2-PaLI-X-55B
- Frequency: 5Hz -> RT2-PaLI-X-5B

Georgia Tech

Action as Text Tokens-Pro

Requires no new architecture
Same VLM, same next -token loss

Keep web knowledge
One network for meaning and motion

Enables co-fine-tuning
Same format for robot and web data

Unifies reasoning and acting
Plan then actions

Georgia Tech

Action as Text Tokens-Limitations/Solutions

Coarse precision 256 levels per dimension	→	**OpenVLA** Percentile bin range
Slow inference 8tokens per step -> 1-3Hz	→	**FAST** Compressed tokens
Single step prediction Jerky motion, no trajectory	→	**Action Chunking** Multi step outputs
order -blind loss Same penalty for bin 152 or 20	→	**Π0** Continuous actions expert

Georgia Tech

Experiments

Objective: Real World generalization and Investigation of Real World capabilities

RQs:

1. How does RT-2 perform on seen tasks and more importantly, generalize over new objects, backgrounds, and environments?
2. Can we observe and measure any emergent capabilities of RT-2?
3. How does the generalization vary with parameter count and other design decisions?
4. Can RT-2 exhibit signs of chain-of-thought reasoning similarly to vision-language models

Georgia Tech

Experimental Setup

- **Scale:** ~6,000 real-world evaluation trials using a 7DoF mobile manipulator
- **Seen tasks:** 200+ instructions from the training distribution (pick, move near, knock over, place upright, open/close drawer, place into drawer)
- **Unseen tasks:** 280+ instructions with new objects, backgrounds and environments, each split into easy and hard

Baselines

- **RT-1:** transformer policy trained only on robot data, the previous state of the art on these tasks. Tests whether VLM pre-training matters.
- **VC-1:** robotics-specific visual foundation model (ViT-L), with language added through a Universal Sentence Encoder and an RT-1 policy head. Tests visual-only pre-training as an alternative to a VLM.
- **R3M:** visual representation (ResNet-50) pretrained on Ego4D human-activity videos, set up the same way as VC-1. Tests pre-training on human video as an alternative to a VLM.
- **MOO:** a VLM marks the target object with a single colored pixel, then a separate policy acts. Tests using a VLM only as a perception helper, not as the policy.

Results (RQ1-Generalization on unseen settings)



Unseen objects

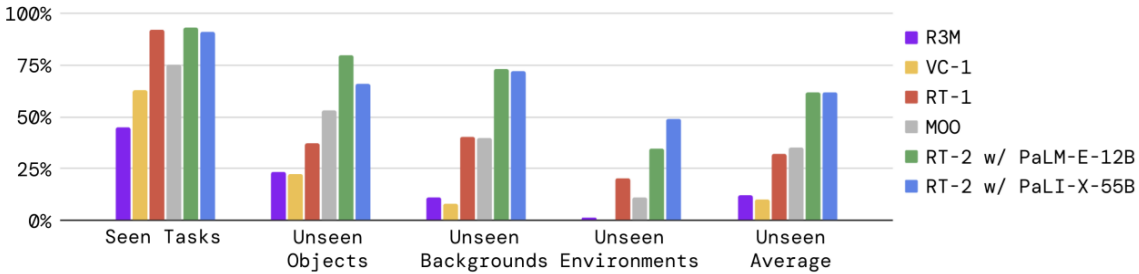


Unseen backgrounds



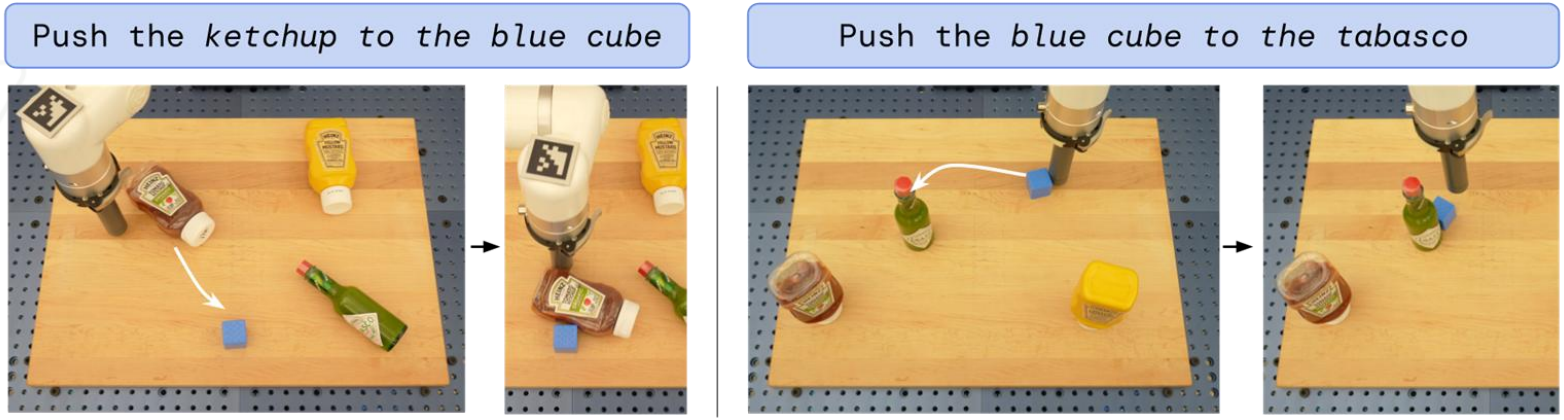
Unseen environments

Model	Seen Tasks	Unseen Objects		Unseen Backgrounds		Unseen Environments		Unseen Average
		Easy	Hard	Easy	Hard	Easy	Hard	
R3M (Nair et al., 2022b)	45	32	14	13	9	0	2	12
VC-1 (Majumdar et al., 2023a)	63	34	10	13	3	0	0	10
RT-1 (Brohan et al., 2022)	**92**	31	43	71	9	26	14	32
MOO (Stone et al., 2023)	75	58	48	38	41	19	3	35
RT-2-PaLI-X-55B (ours)	**91**	**70**	**62**	**96**	**48**	**63**	**35**	**62**
RT-2-PaLM-E-12B[1] (ours)	**93**	**84**	**76**	**75**	**71**	**36**	**33**	**62**



Takeaway: both backbones average 62% on unseen tasks, so PaLM-E-12B matches PaLI-X-55B with about 4.5× fewer parameters. Size isn't everything

Results (RQ1- Language Table)

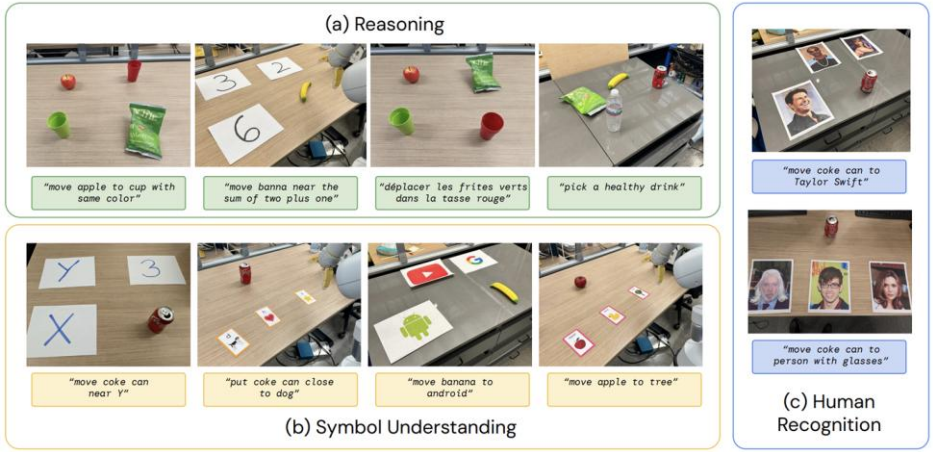


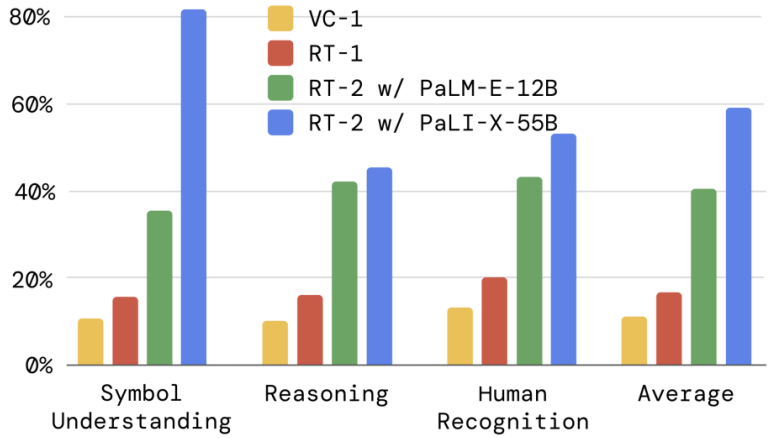
Model	Language-Table
BC-Zero (Jang et al., 2021)	72 ± 3
RT-1 (Brohan et al., 2022)	74 ± 13
LAVA (Lynch et al., 2022)	77 ± 4
RT-2-PaLI-3B (ours)	**90 ± 10**

Table 1 | Performance on the simulated Language-Table tasks (Lynch and Sermanet, 2020).

Georgia Tech

Results (RQ2- Emerging Capabilities)

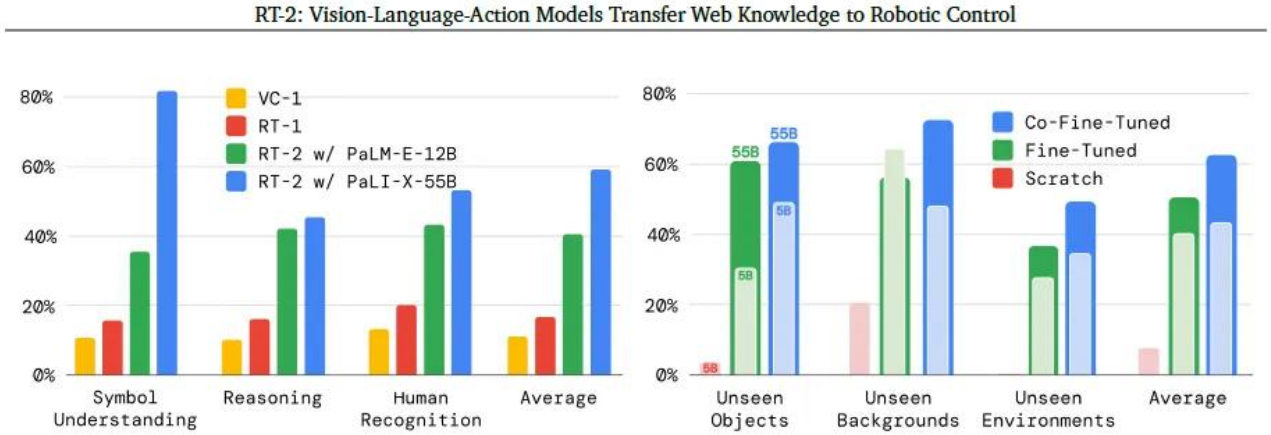




Web knowledge transfers: ~3× RT-1 on emergent skills (~60% vs 17% avg), and best on simulated Language-Table (90%).

Takeaway: RT-2 shows strong semantic transfer; ‘reasoning’ is a stronger claim than the evidence supports

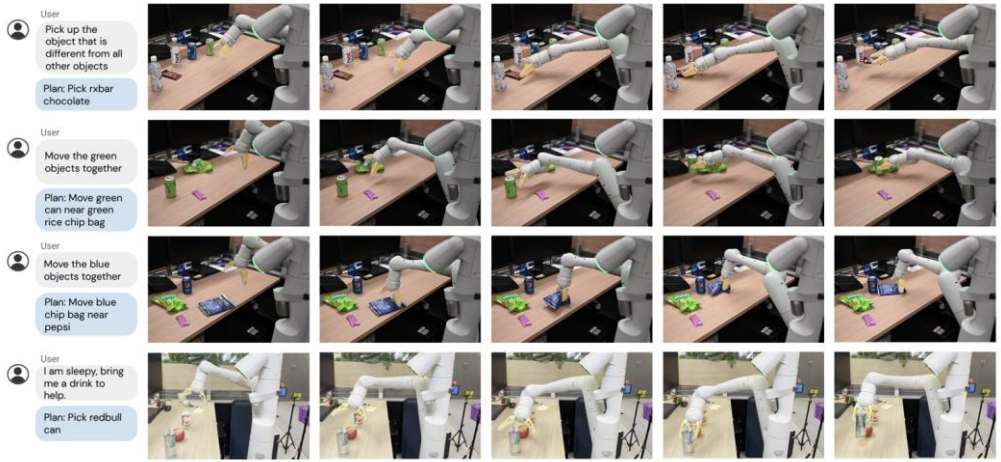
Results (RQ3- Effect of Model Size and Training Recipe)



Takeaway: **co-fine-tuned 5B scores 44% and fine-tuned 55B scores 52%, so scale buys more than the training recipe here**

Georgia Tech

Results (RQ4- Chain of Thoughts Reasoning)



~~Instruction → Action~~

Instruction → **Plan** → Action

Georgia Tech

Results (RQ4- Chain of Thoughts Reasoning)

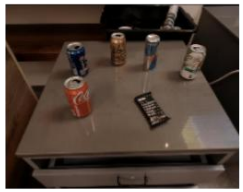
Prompt:
Given <img> Instruction: Bring me a drink.
Prediction:
Plan: pick 7up can.
Action: 1 143 129 123 145 114 115 127



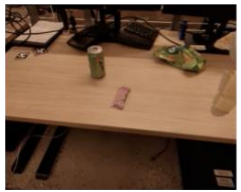
Prompt:
Given <img> Instruction: Move all the objects together.
Prediction:
Plan: move green can near green rice chip bag.
Action: 1 128 126 127 135 123 119 127



Prompt:
Given <img> Instruction: Pick the object that is different from all other objects
Prediction:
Plan: pick rxbar chocolate. Action: 1 128 129 125 131 125 128 127

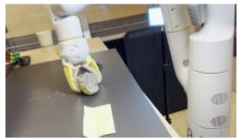


Prompt:
Given <img> Instruction: Move the green objects together.
Prediction:
Plan: move green can near green rice chip bag.
Action: 1 130 129 121 131 127 128 127



Prompt:
Given <img> I need to hammer a nail, what object from the scene might be useful?
Prediction:
Rocks. Action: 1 129 138 122 132 135 106 127







Takeaway: After a few hundred steps of fine-tuning on “Plan:” annotations, RT-2-PaLM-E writes a language plan before its action tokens

Georgia Tech

Reasoning vs Inference Cost

- The 55B model runs at 1–3 Hz and the 5B at about 5 Hz. The gain from 5B to 55B (44% → 63%) costs roughly 2–3× the latency.
- PaLM-E-12B matches PaLI-X-55B on unseen tasks, so much of that extra latency buys nothing on the generalization task.
- Chain-of-thought adds plan tokens to every prediction, so latency goes up further. The paper shows CoT only qualitatively, with no success-rate comparison

Georgia Tech

What persisted and what changed? (RT-1/RT-2)

Persisted

- **Robot data**: RT-1 dataset, 13 robots, 17 months, ~130k episodes
- **Robot & tasks**: same mobile manipulator, skills and seen-task evals
- **Action discretization**: 256 uniform bins per dimension, predicted as tokens
- **Closed-loop control**: image + instruction → next action, step by step

Changed

- **Model**: 35M robotics transformer → 5B–55B pretrained VLM
- **Actions**: separate action head → text tokens in the VLM vocabulary
- **Training**: robot data only → co-fine-tuned with web data
- **Generalization**: unseen average 32% → 62%; emergent reasoning
- **Cost**: ~3 Hz on a small model → 1–5 Hz on cloud TPUs

Limitations & Societal Implications

Limitations

- **Embodiment transfer/limited motor skills:** web data improves semantics, but physical skills stay limited to those in the robot data
- **Compute cost:** 55B model runs on cloud TPUs at 1–3 Hz, too slow for high-frequency control
- **Closed backbones:** relies on few, proprietary VLMs (PaLI-X, PaLM-E), so hard to reproduce
- **Narrow evaluation:** one robot embodiment, tabletop tasks in office kitchens

Societal implications

- **Web priors become actions:** biases and errors in internet data can turn into physical behavior
- **Safety**: robots that act on instructions never seen in training need new safeguards and evaluation
- **Access**: training and serving cost concentrates capability in a few labs, motivating open VLAs
- **Upside**: general-purpose assistive robots that follow everyday language

MolmoAct2: Action Reasoning Models for Real-world Deployment

[Fang et al.]
[preprint arXiv 2026]

[Nadira Mahamane,Josh Traynelis, Zihao Zhao]

Georgia Tech

Outline

- MolmoAct2
  - Problem Statement
  - Related Works
  - Approach – Seminal and frontier
  - Experiments & Results
  - What persisted and what changed?
  - Limitations and Implications
  - Discussion Questions

Georgia Tech

Problem Statement

- **Challenges**: Frontier Robotic policies are closed source and models currently have weak spatial grounding
  - These models are not readily deployable
  - Expensive hardware and slower reasoning (hundreds of tokens of latency per action)

- **MolmoAct2** is an action reasoning model that is effective for real-world deployment.

MolmoAct2 Contributions

1. Provides a stronger open embodied reasoning backbone, **Molmo2-ER**
2. **Open source training data sets:** BimanualYAM, DROID subsets, SO-100/101,
3. **VLA Architecture:** graft the discrete-token VLM into the flow-matching continuous-action expert via per-layer key-value (KV) conditioning
4. Introduces an open source multi-embodiment action tokenizer
5. **MolmoAct2-Think,** an adaptive-depth reasoning variant, reduced latency

Georgia Tech

Related Work

Category	Method/Paper	What it does	Drawbacks	MolmoAct2
Generalized Robot Policies	RT-1	Larger Robotic datasets	Trained on Robotic data, without web pre-training. Closed	Open model backbone, data and code
VLA Models	OpenVLA	Pretrained VLM and trained on robotic data	OpenVLA uses discrete actions	FAST tokens and continuous flow
Embodied Reasoning	ECoT	Embodied CoT performs reasoning with bounding boxes	Large token requirement, compared to other VLAs (openVLA). High Latency	Adaptive Depth Reasoning

Georgia Tech

Molmo2-ER: Embodied-Reasoning Backbone

MOTIVATION

General-purpose VLMs rarely learn metric distance, free space or scene geometry

Data 3.3M samples covering pointing, image and video spatial QA, multi-view correspondence and abstract spatial reasoning

Specialize-then-rehearse recipe

STAGE 1

Specialize

20K steps on the embodied corpus

STAGE 2

Rehearse

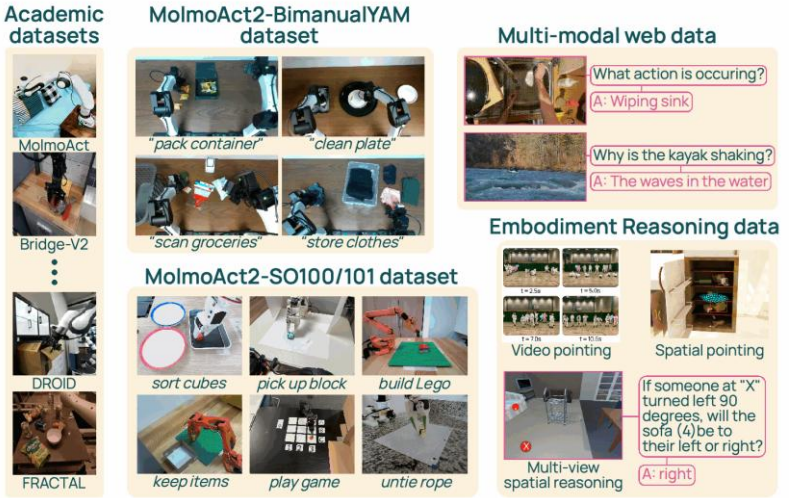
1.5K steps on a 1:1 mix with Molmo2 general data

Georgia Tech

Training Data

MOTIVATION

Open robot data is fragmented and noisy and lacks a high-quality bimanual corpus



MolmoAct2-BimanualYAM **34.5k demos over 720 h**
Teleoperated demos forming the largest open bimanual dataset

MolmoAct2-SO100/101 **38.1k episodes**
Quality-filtered from 1,222 community LeRobot datasets

MolmoAct2-DROID **74.6k episodes**
Successful Franka trajectories from DROID without idle segments

Language re-annotation **Qwen3.5-27B**
Unique instructions rise from 22% to 46%

Georgia Tech

Three-stage Training Pipeline

MOTIVATION

Learning tokens, embeddings and a flow controller at once makes optimization hard

STAGE 1

Discrete pre-training

- Next-token prediction on FAST action tokens
- Data mix of 90% robot and 10% multimodal

STAGE 2

Continuous post-training

- Adds a flow-matching action expert
- VLM insulated from the flow loss

STAGE 3

Embodiment fine-tuning

- Robot-only data for each embodiment
- Flow gradients now update the VLM

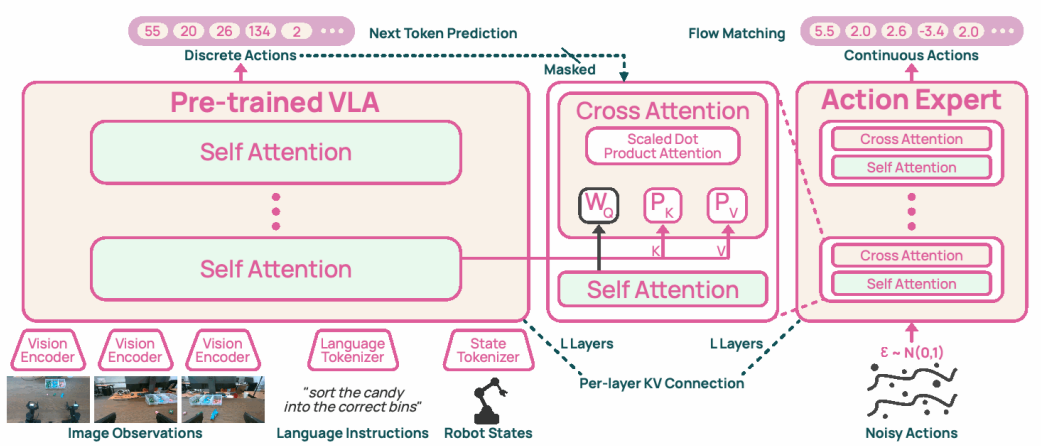
MolmoAct2-FAST Tokenizer Encodes 1 s of 32-D actions via DCT, quantization and BPE into a 2,048-token vocabulary

Georgia Tech

Architecture: Per-layer KV Conditioning

MOTIVATION

Conditioning only on final-layer features exposes just part of the backbone



Per-layer KV conditioning

Expert layer ℓ reads the keys and values of VLM layer ℓ

Masked action tokens

Target discrete actions are hidden from the expert

Multiple flow samples

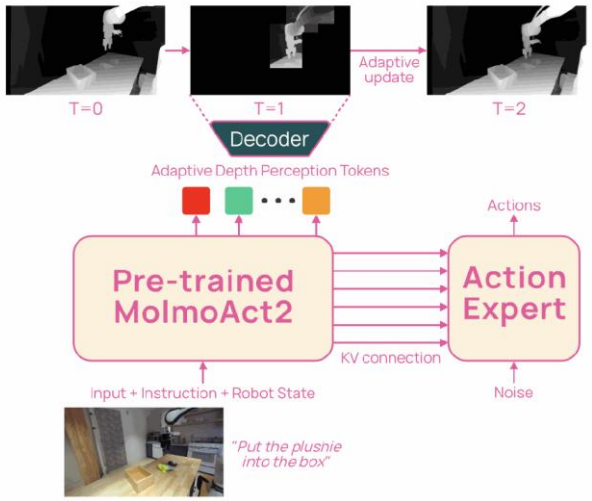
Several noise levels per chunk from one VLM pass

Georgia Tech

MolmoAct2-Think: Adaptive Depth Reasoning

MOTIVATION

Imitation supervises geometry only indirectly and dense per-step reasoning is slow



Depth perception tokens

The model predicts 100 VQ-VAE depth codes on a 10×10 grid before the action

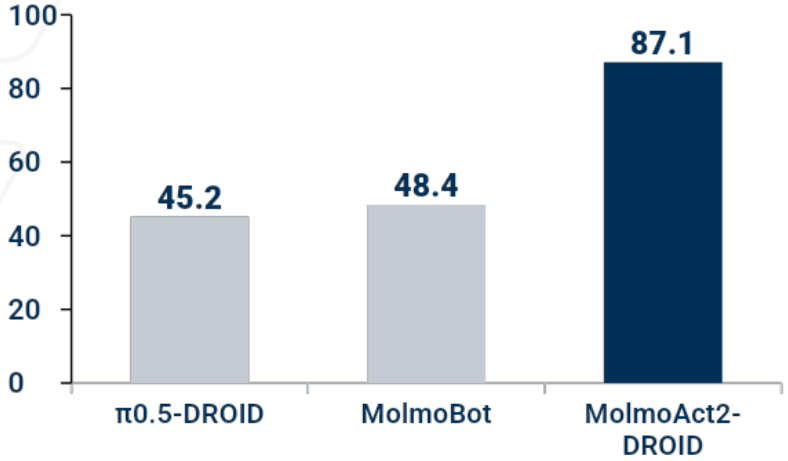
Adaptive re-generation

Only cells whose RGB patch has changed (cosine < 0.996) are regenerated. All other cells are reused from the cache.

Georgia Tech

Out-of-the-box Deployment

Zero-shot real-world DROID evaluation (%)



Benchmark	MolmoAct2	Best baseline
SO-100/101 (real)	**56.7**	45.3 π0
MolmoBot (sim)	**20.6**	10.0 π0.5
MolmoSpaces (sim)	**37.7**	34.5 π0.5
↳ Open task	**9.5**	22.7 π0.5
YAM (fine-tuned)	**≈50**	35.5 OpenVLA-OFT

Georgia Tech

LIBERO Fine-tuning Results

Suite	π0.5	MolmoAct2	Think	Δ Think
Spatial	98.8	97.8	**98.8**	**+1.0**
Object	98.2	100.0	**99.8**	**−0.2**
Goal	98.0	97.8	**98.5**	**+0.7**
Long	92.4	93.2	**95.4**	**+2.2**
Average	**96.9**	**97.2**	**98.1**	**+0.9**

+2.2 pts

MolmoAct2-Think gain on the hardest suite (Long)

+0.3 pts

Average margin of MolmoAct2 over π0.5

Georgia Tech

Depth Fine-tuning Ablation

Depth-and-action only

97.50

→

+ Mixed training

97.65 +0.15

→

+ Noise and depth gate

98.10 +0.45

Depth-token noise

Because inference conditions on predicted rather than oracle depth, 10% of input depth codes are replaced with random codes during training.

Per-layer depth gate

A learned gate initialized with bias −4 controls how strongly each expert layer uses the depth prefix.

Georgia Tech

MolmoAct2-Think: Accuracy vs. Latency

MolmoAct2

97.2%

LIBERO average success

179 ms

per 10-step action chunk (55.8 Hz amortized)

+0.9

pts accuracy

4.4×

latency

MolmoAct2-Think

98.1%

LIBERO average success

787 ms

per 10-step action chunk (12.7 Hz amortized)

Georgia Tech

What persisted and what changed?

Persisted from MolmoAct

- Open weights, training code and data
- Depth perception tokens for action reasoning
- Discrete action tokens predicted by the VLM

What changed in MolmoAct2

- **Backbone:** Molmo2-ER trained for embodied reasoning
- **Data:** three new open datasets for YAM, SO-100/101 and DROID
- **Tokenizer:** open MolmoAct2-FAST trained across five embodiments
- **Architecture:** flow-matching expert attached through per-layer KV
- **Reasoning:** depth tokens regenerated only where the scene changes

Limitations & Societal Implications

Limitations

- **Open-loop execution:** actions within each 10- to 30-step chunk run without feedback from new observations
- **Embodiment coverage:** zero-shot transfer is shown only on YAM, SO-100/101 and DROID Franka
- **Evaluation scale:** 15 trials per task and 5 per perturbation without recovery or safety metrics

Societal implications

- **Broader access:** open models on low-to-medium-cost robots let academic labs build on a deployable VLA
- **Deployment risk:** open models may reach homes before safety validation. Speed limits and an emergency stop remain essential.

Georgia Tech

Discussion Questions

1. How could we test whether depth tokens causally shape the predicted actions?
2. Do the gains come mainly from data and the backbone or from the architecture?
3. How could chunked execution become reactive without giving up throughput?
4. What evidence would demonstrate transfer to an unseen embodiment?
5. RT-2 cast actions as language tokens. Are discrete action tokens still necessary once a continuous expert executes the actions?