Embodied AI

Roman Mineyev, Haoran Liu, Pranav Rambhia

Motivation

Here's my fridge. Let's say you are a robot and I want you to cook a breakfast for me! The breakfast includes hot milk, toasted bread and 2 fried eggs.

Please provide me with step-by-step instructions of how you are going to prepare that for me.

them to your preferred style. Please let me know if you would like them sunny-side up, over easy, or scrambled.

Step 5: Serve Your Breakfast

Once the toast is ready, I will place it on a plate. The fried eggs will then be placed beside the toast. Finally, I will carefully retrieve the warm milk from the microwave. Your breakfast of hot milk, toasted bread, and two fried eggs will then be ready for you to enjoy.

Problem Statement

- The VLMs are powerful tools!
  - Spatial/semantic reasoning, Open-vocabulary recognition, …
- What if robots can…?

Problem Statement

HOW?

Potential Approach

Train Robot VLMs | Directly use VLMs

Insufficient data | not grounded to real world

High level planner +
low level controller

RT-1 + PaLM-E

A unified end-to-end model

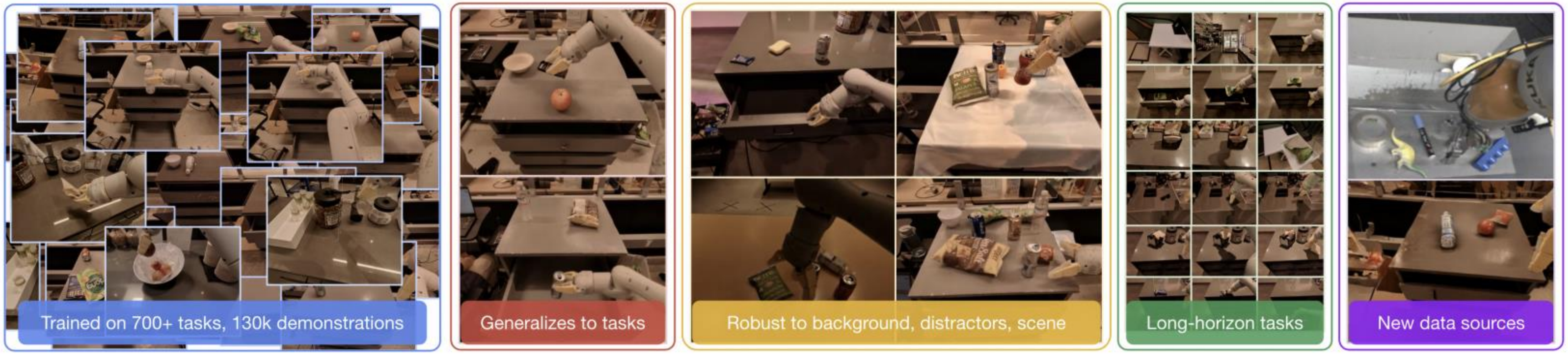
RT-2


RT-1

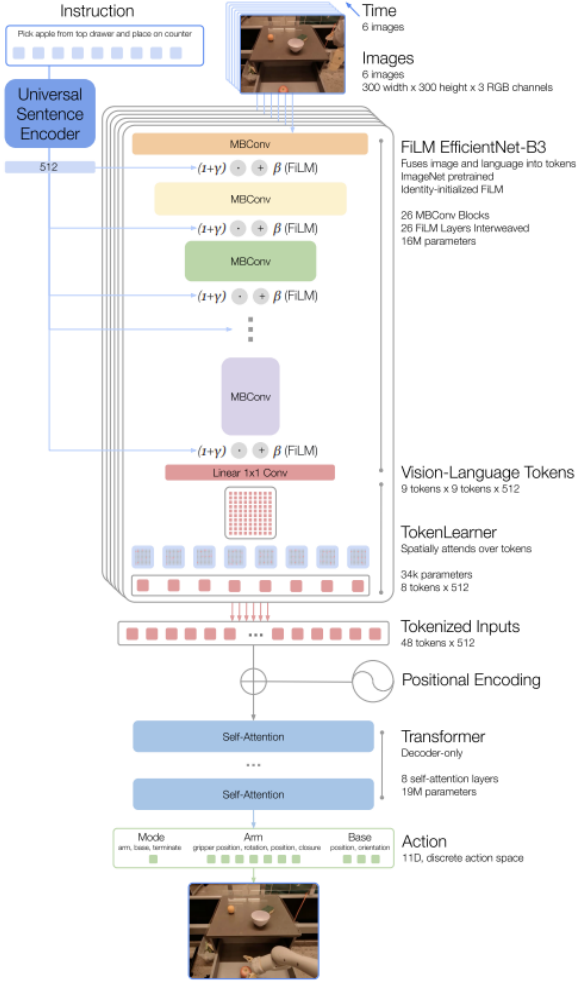
Building a Generalist Robot "Apprentice"

The Goal:

- Leverage large-scale data to build a robust robot policy that can generalize well across several different axes



RT-1 Recipe: A Massive Dataset + A Big, Fast Brain

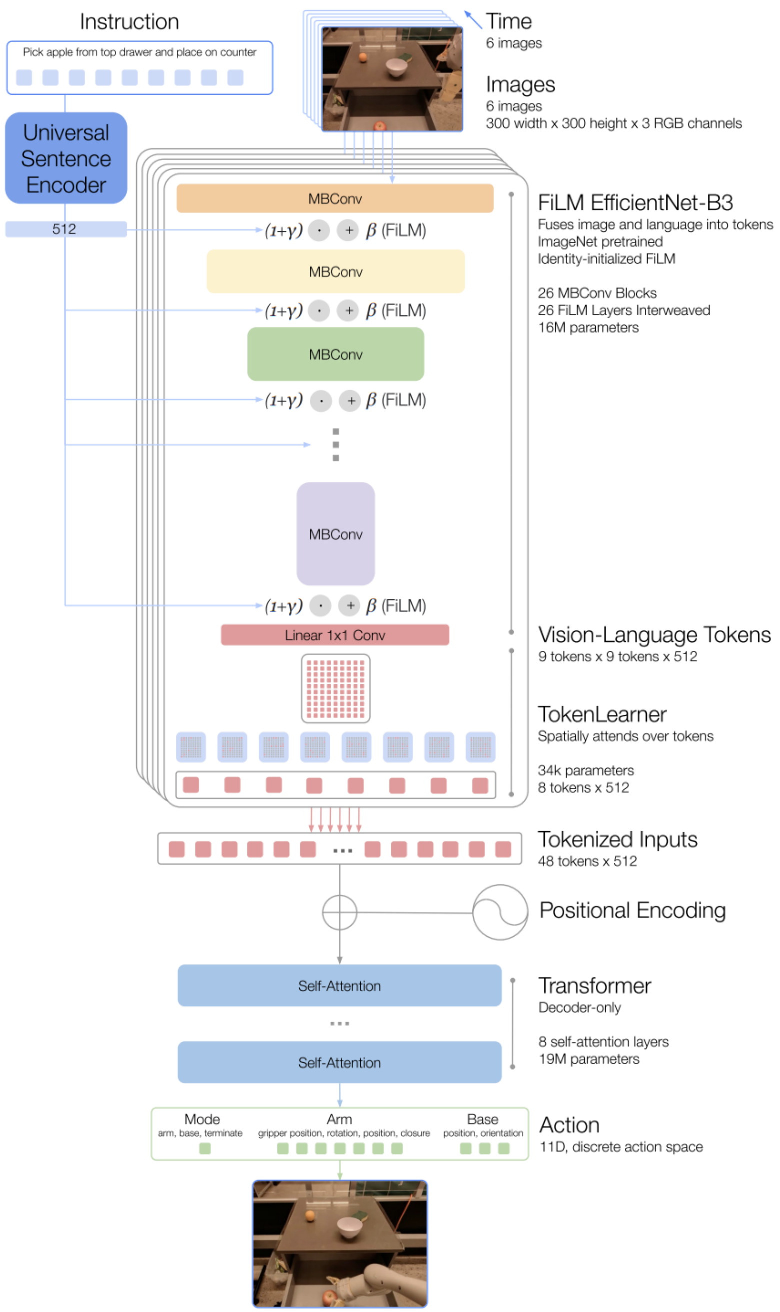


RT-1 Recipe: A Massive Dataset

Skill	Count	Description	Example Instruction
Pick `Object`	130	Lift the object off the surface	pick iced tea can
Move `Object` Near `Object`	337	Move the first object near the second	move pepsi can near rxbar blueberry
Place `Object` Upright	8	Place an elongated object upright	place water bottle upright
Knock `Object` Over	8	Knock an elongated object over	knock redbull can over
Open `Drawer`	3	Open any of the cabinet drawers	open the top drawer
Close `Drawer`	3	Close any of the cabinet drawers	close the middle drawer
Place `Object` into `Receptacle`	84	Place an object into a receptacle	place brown chip bag into white bowl
Pick `Object` from `Receptacle` and Place on the Counter	162	Pick an object up from a location and then place it on the counter	pick green jalapeno chip bag from paper bowl and place on counter
Section 6.3 and 6.4 tasks	9	Skills trained for realistic, long instructions	open the large glass jar of pistachios pull napkin out of dispenser grab scooper
Total	744		

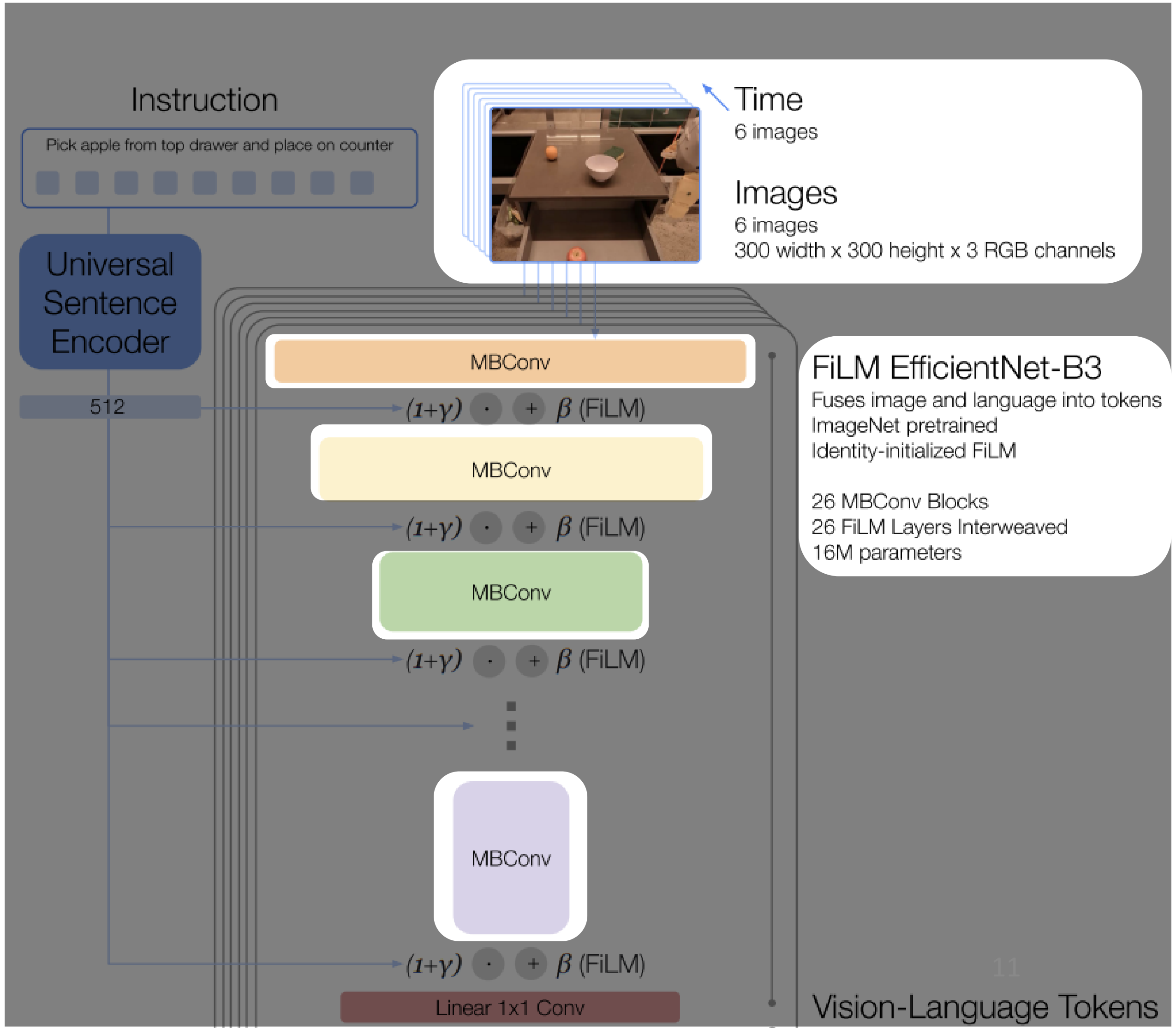
RT-1 Recipe: A Big, Fast Brain

- Inputs:
  - History of past 6 images
  - Task specified in natural language
- Outputs:
  - 11D action discretized into 256 bins:
    - x, y, z, roll, pitch, yaw for the end effector
    - Gripper closedness command
    - x, y, yaw for the base
    - A discrete variable to switch between 3 modes: Controlling arm, base or termination the episode



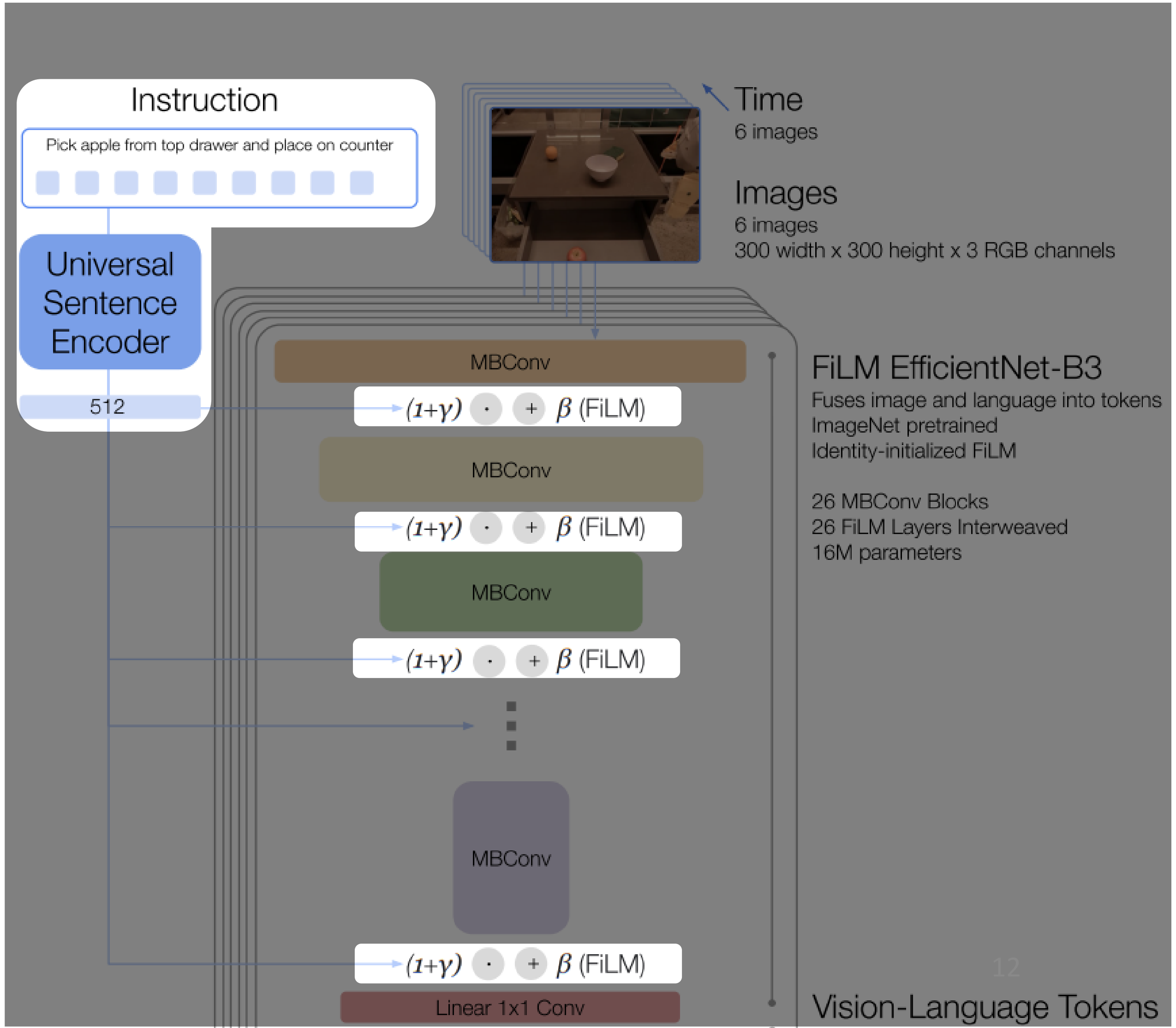
RT-1 Recipe: A Big, Fast Brain

- ImageNet pre-trained EfficientNet-B3 used to encode input image
- Transformer based Universal Sentence Encoder was used to encode language instructions
- FiLM Layers used to condition image features on language instruction through affine transformations



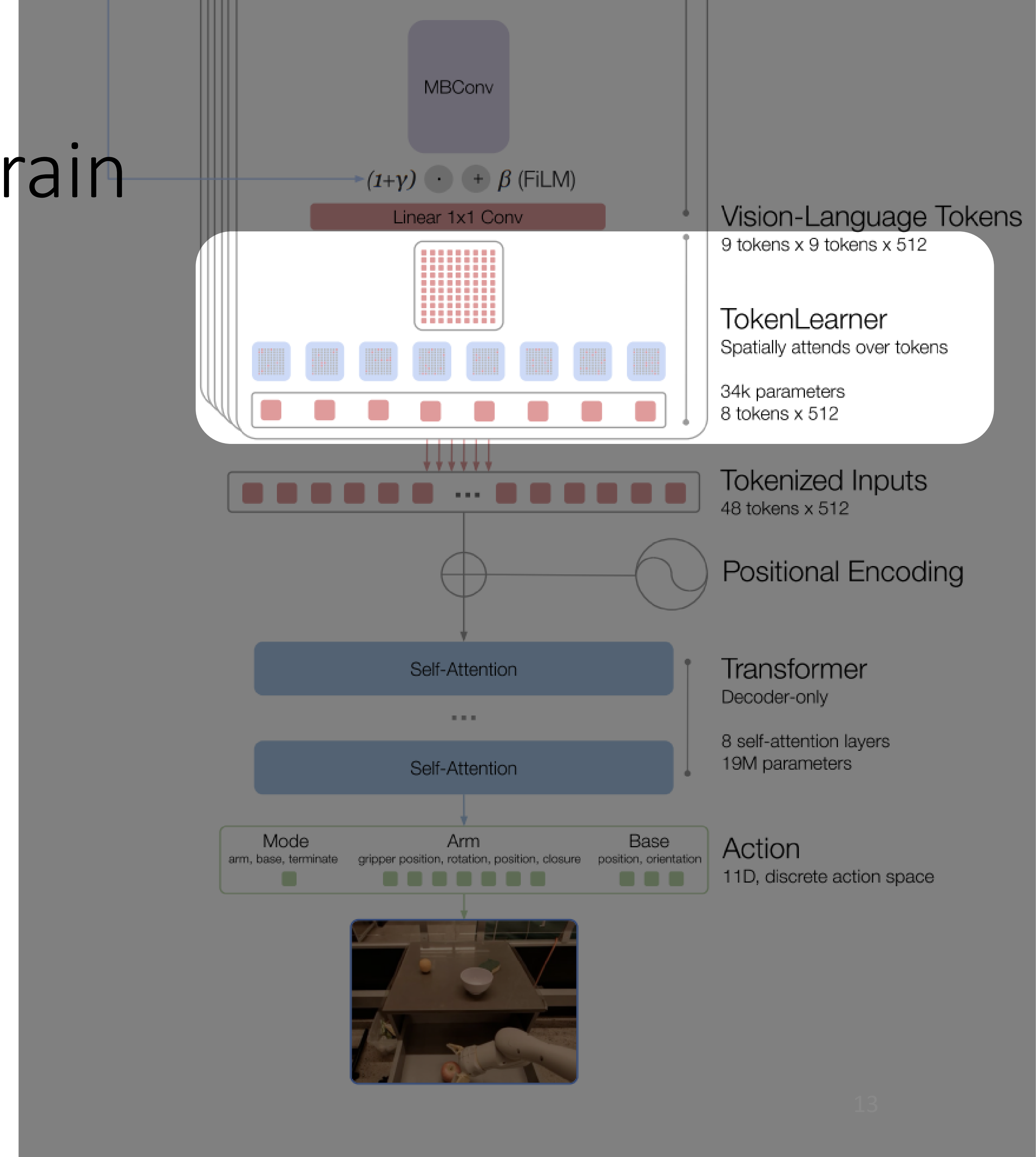
RT-1 Recipe: A Big, Fast Brain

- ImageNet pre-trained EfficientNet-B3 used to encode input image
- Transformer based Universal Sentence Encoder was used to encode language instructions
- FiLM Layers used to condition image features on language instruction through affine transformations



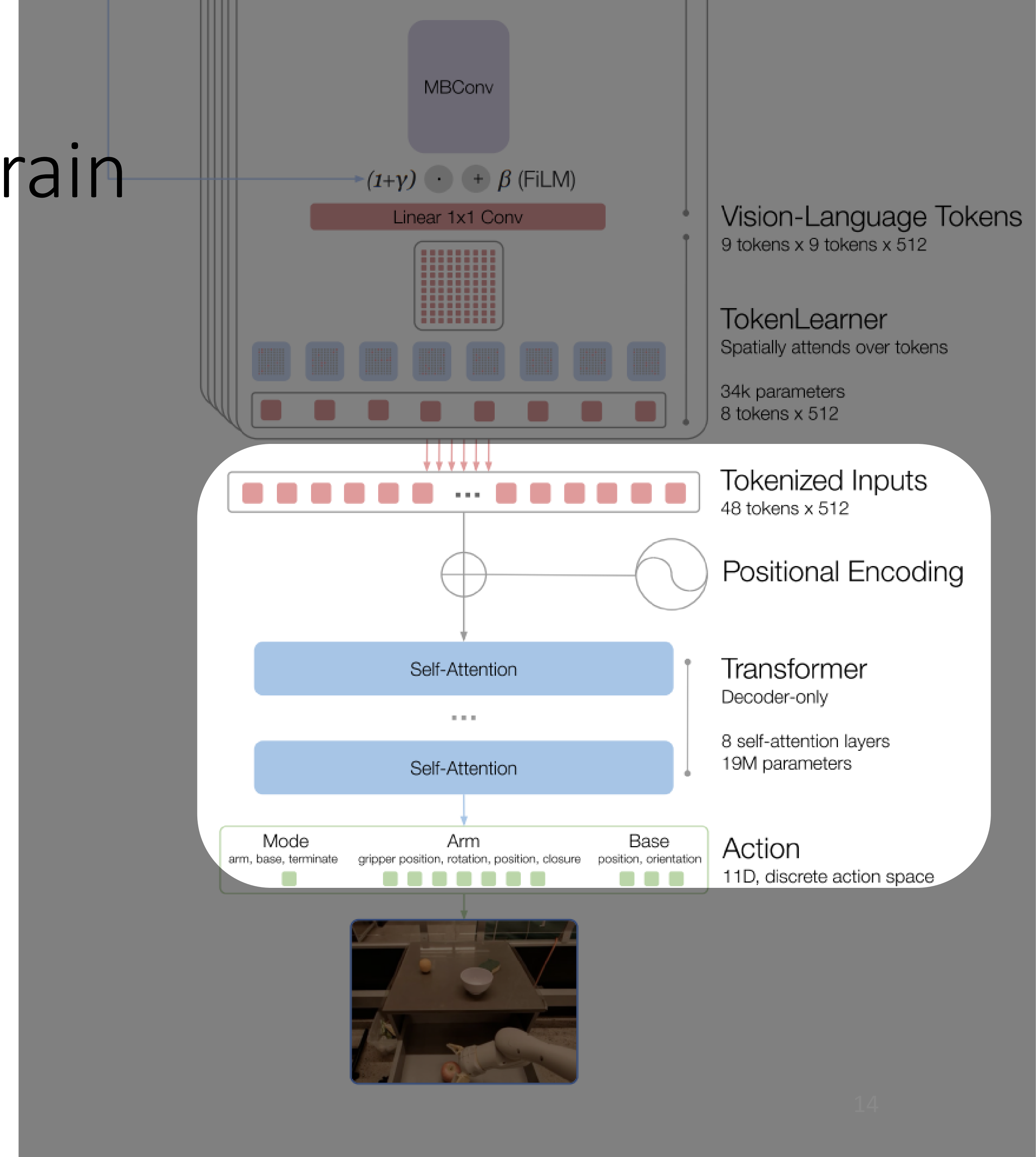
RT-1 Recipe: A Big, Fast Brain

- TokenLearner used to compress to 8 tokens per image
- 8 tokens per image (48 total) concatenated together and passed to the transformer
- Output of transformer projected onto the 256 discretized bins for each action



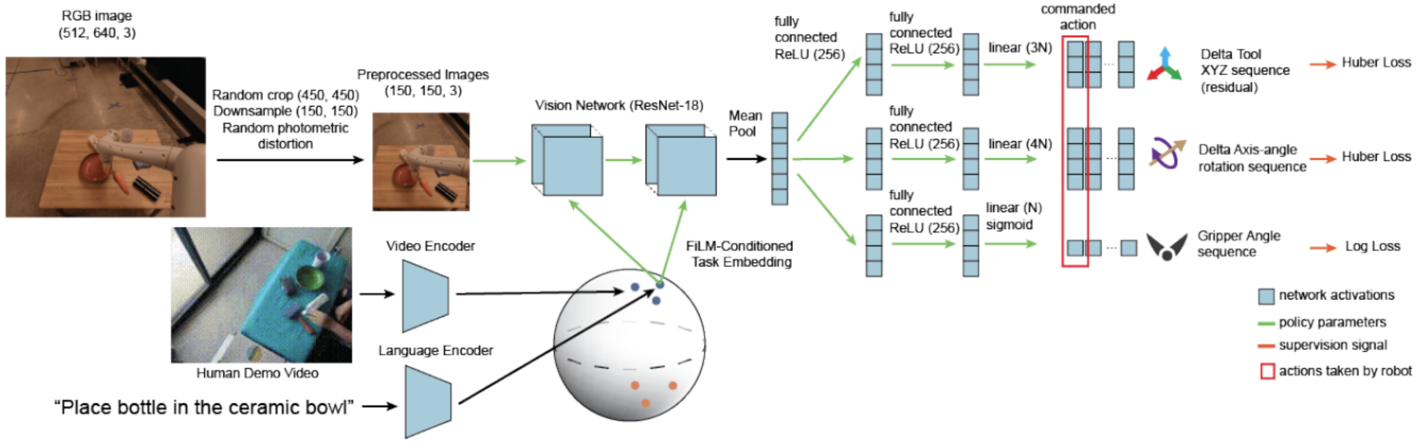
RT-1 Recipe: A Big, Fast Brain

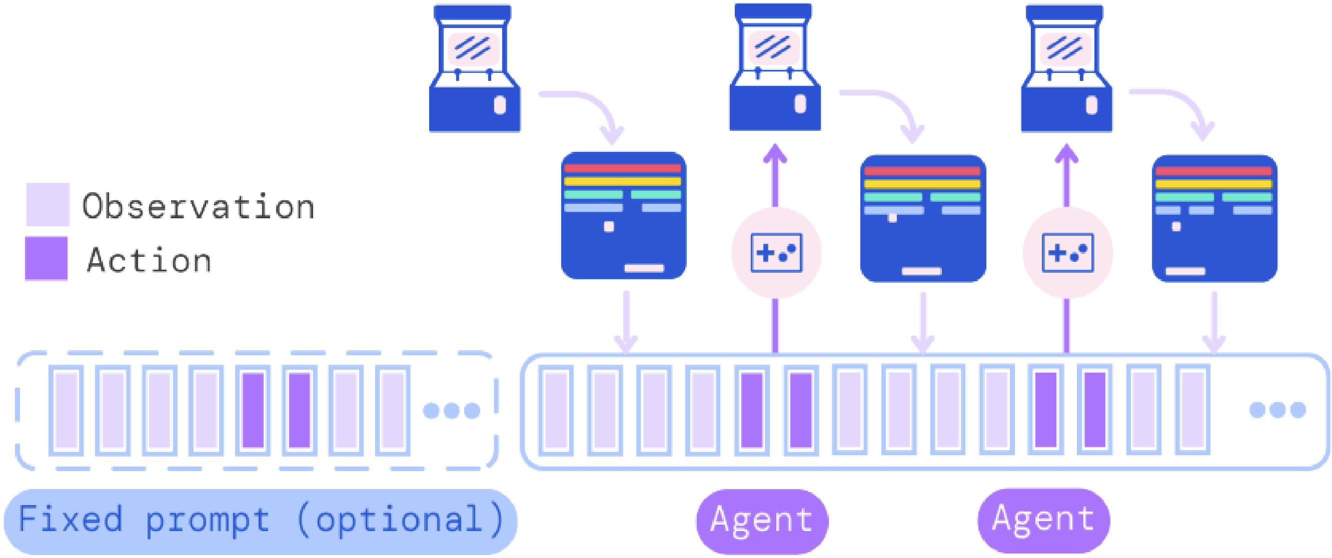
- TokenLearner used to compress to 8 tokens per image
- 8 tokens per image (48 total) concatenated together and passed to the transformer
- Output of transformer projected onto the 256 discretized bins for each action



Results: Baselines Used

BC-Z

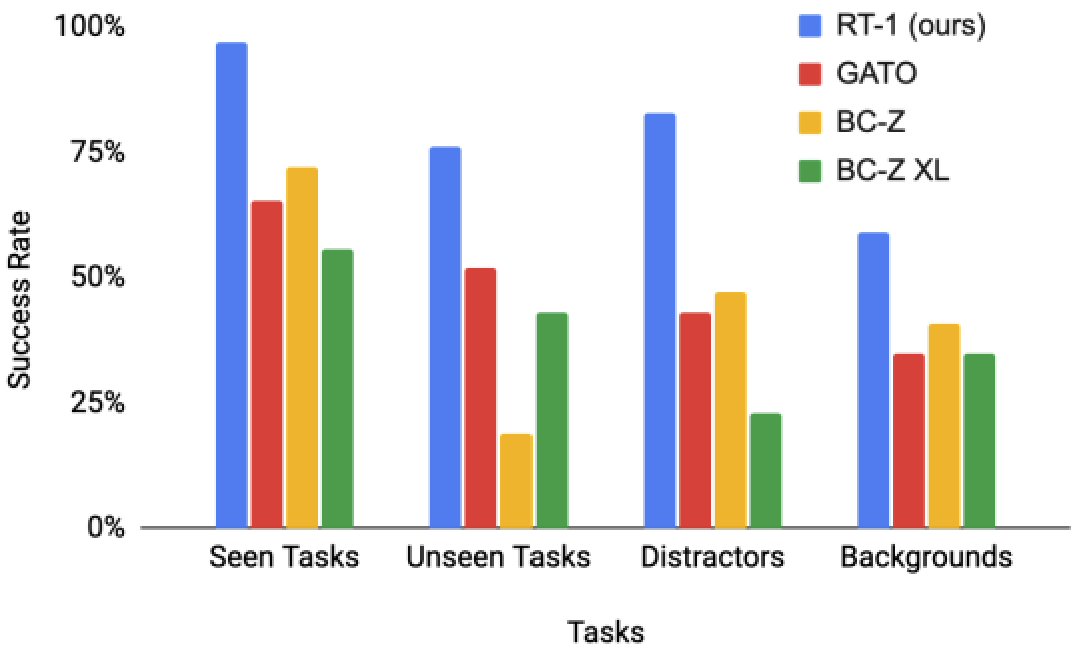




Gato

Results: Generalization

- RT-1 achieves **97%** accuracy on seen tasks, 25% higher than next best model
- RT-1 completes **76%** of the never-before-seen instructions, 24% more than the next best baseline
- RT-1 demonstrates robustness completing **83%** and **59%** of the distractor and background robustness tasks, 36% and 18% higher than next best alternative

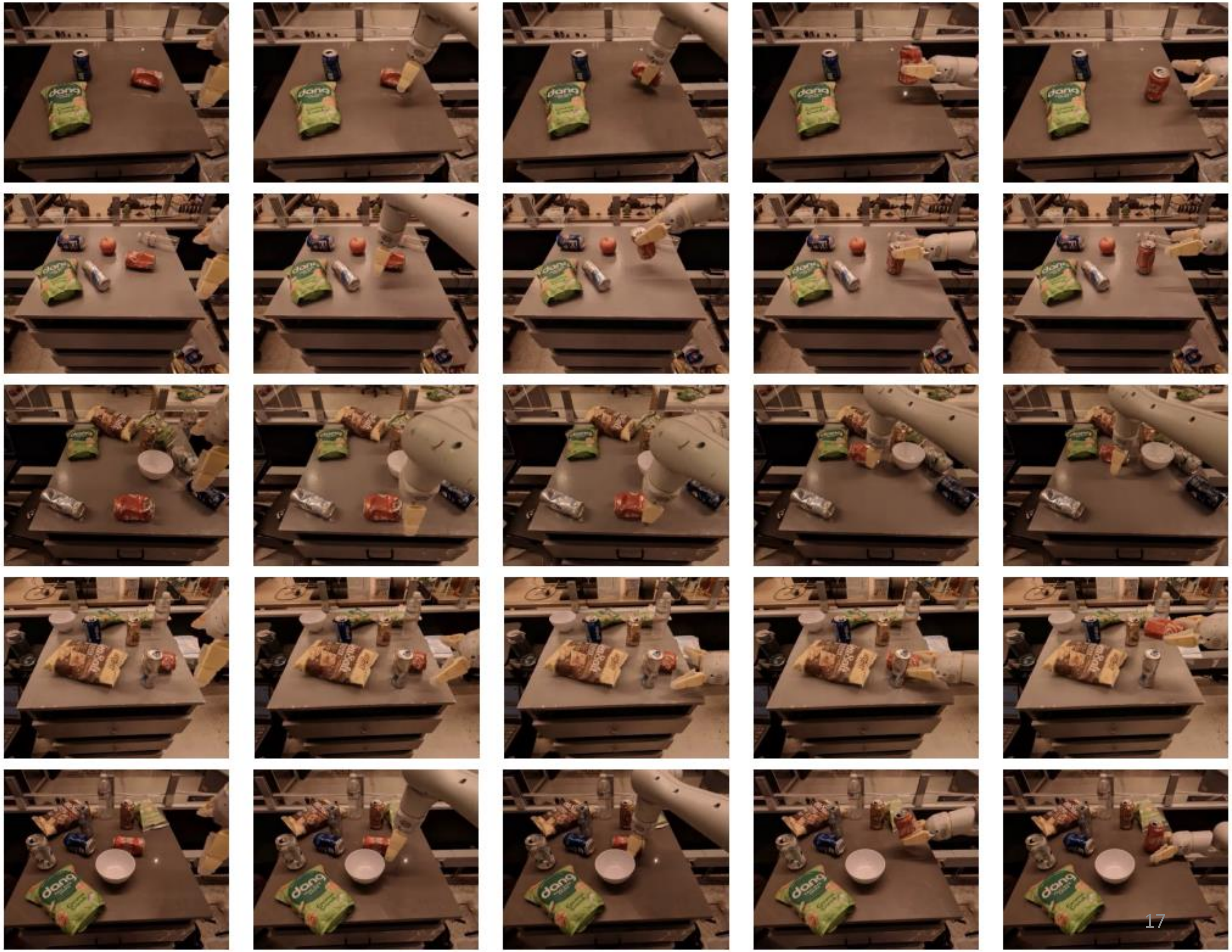


Model	Seen Tasks	Unseen Tasks	Distractors	Backgrounds
Gato (Reed et al., 2022)	65	52	43	35
BC-Z (Jang et al., 2021)	72	19	47	41
BC-Z XL	56	43	23	35
RT-1 (ours)	**97**	**76**	**83**	**59**

Easy
2 - 5 distractors,
no occlusion

Medium
9 distractors,
no occlusion

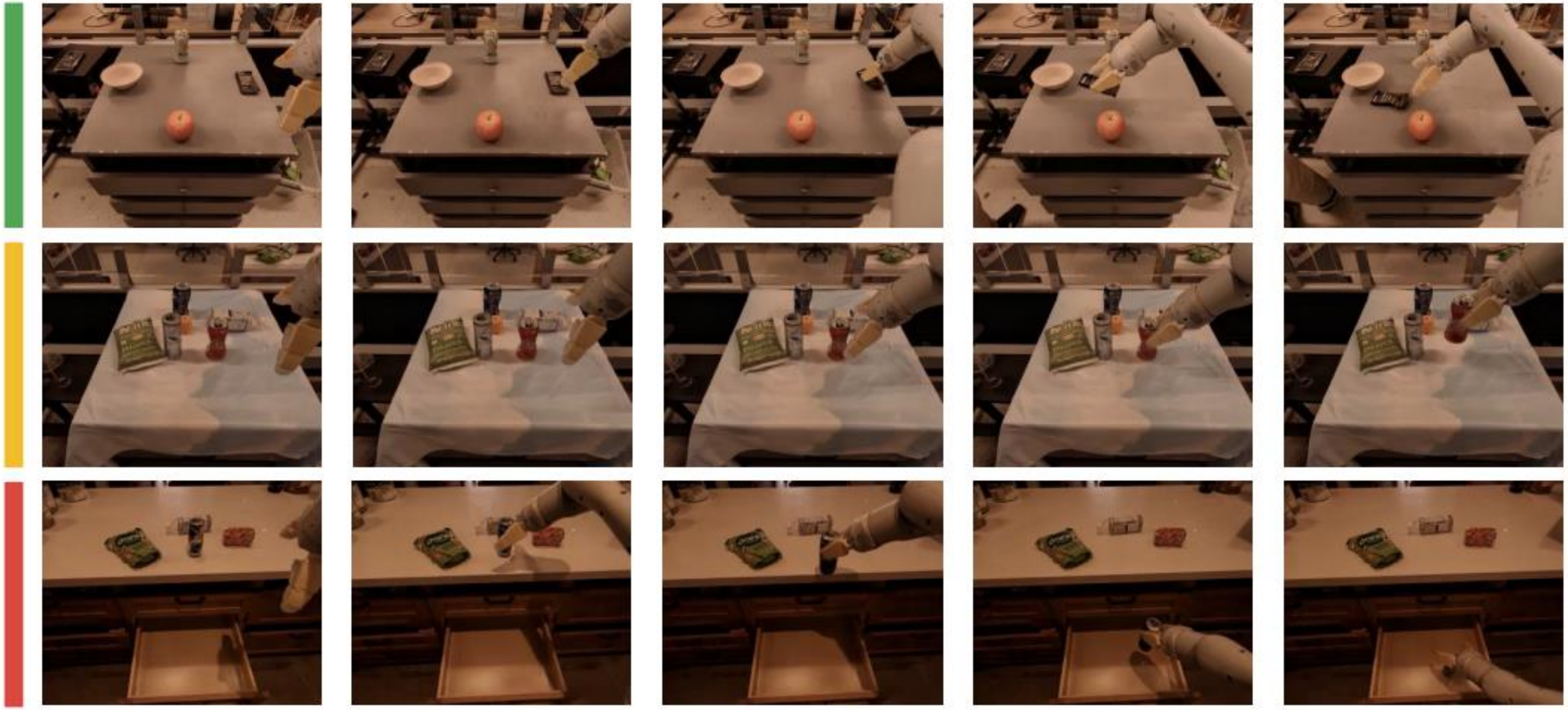
Hard
9 distractors,
occlusion



Easy
same background,
same texture

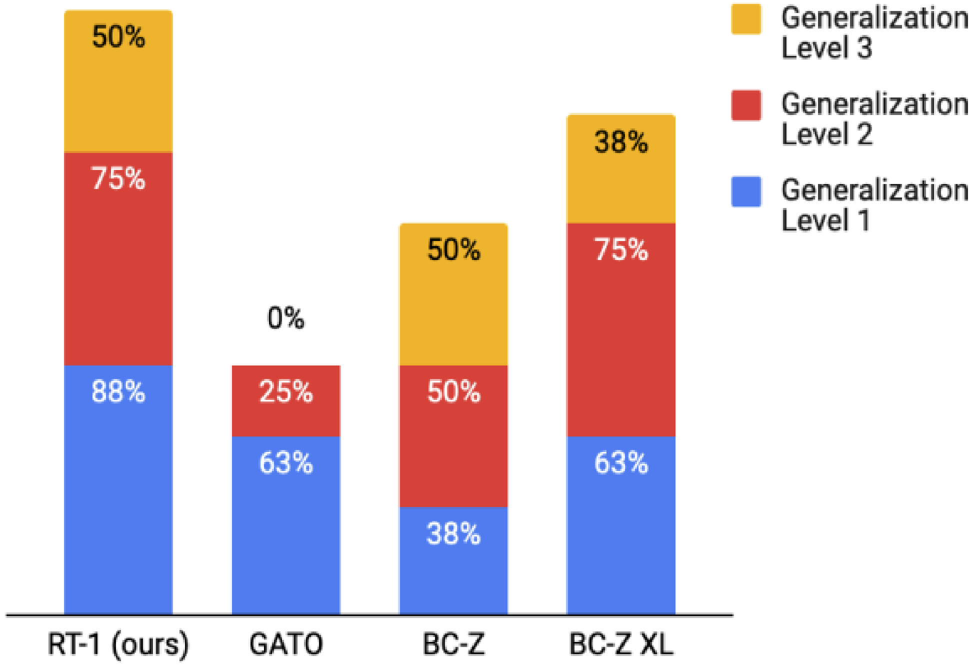
Medium
same background,
new texture

Hard
new background,
new texture



Results: Generalization

- RT-1 was tested on generalization to realistic kitchen environments. Three levels of generalization were defined as follows:
  - **L1** for generalization to the new counter-top layout and lighting conditions
  - **L2** for additionally generalization to unseen distractor objects
  - **L3** for additional generalization to drastically new task settings, new task objects or objects in unseen locations such as near a sink

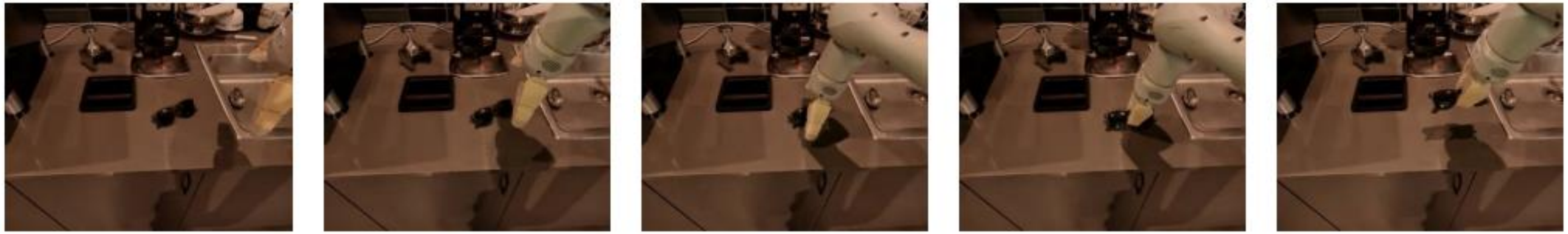


		Generalization Scenario Levels		
Models	All	L1	L2	L3
Gato Reed et al. (2022)	30	63	25	0
BC-Z Jang et al. (2021)	45	38	50	**50**
BC-Z XL	55	63	**75**	38
RT-1 (ours)	**70**	**88**	**75**	**50**

Level 1 Generalization

Level 2 Generalization

Level 3 Generalization



Results: Adding Sim Data

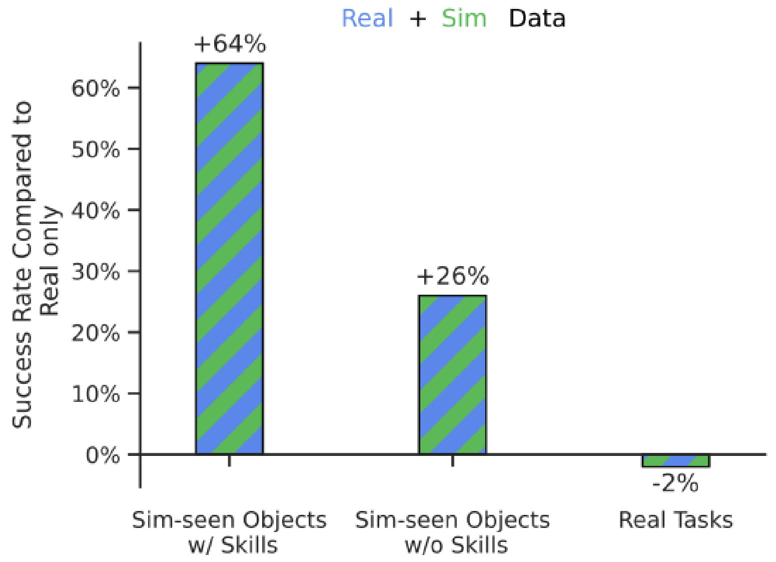
- To test whether RT-1 can learn from sim data, task demonstrations with a select few objects were removed from the training data

- Demonstrations were added for the removed objects in sim on selected skills

- RT-1 does not lose much performance on tasks with real demonstrations

- RT-1 improves by **64%** on tasks with demonstrations seen in sim only

- RT-1 improves completion rate by **26%** on tasks involving objects only seen in sim



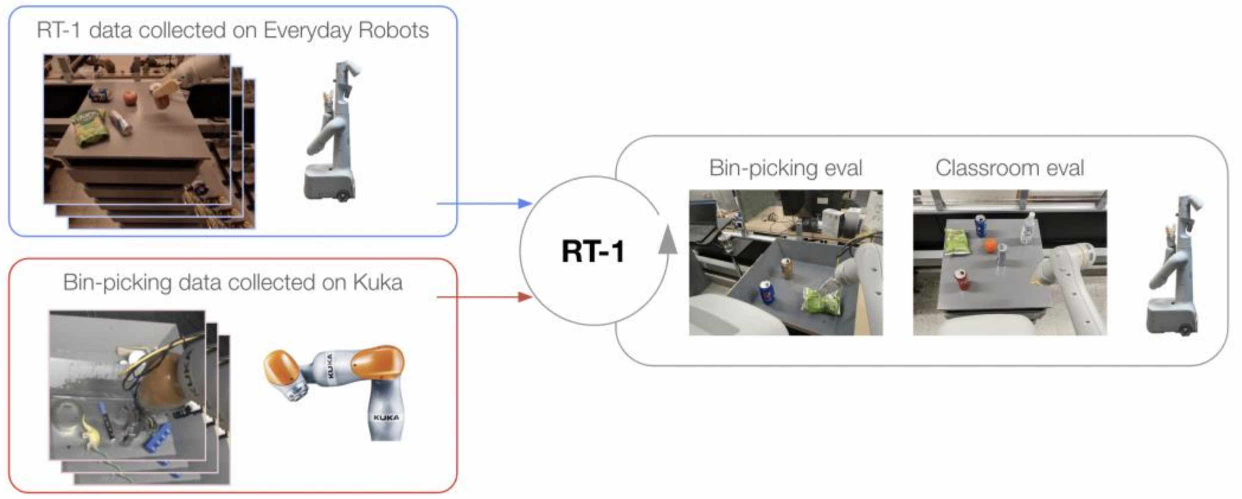
		Real Objects	Sim Objects (not seen in real)	
Models	Training Data	Seen Skill w/ Objects	Seen Skill w/ Objects	Unseen Skill w/ Objects
RT-1	Real Only	92	23	7
RT-1	Real + Sim	90(-2)	**87(+64)**	**33(+26)**

Results: Adding data from other robots

- To test whether RT-1 can benefit from data from other robots, demonstrations of Kuka IIWA robot picking up objects indiscriminately from a bin was added

- RT-1 does not lose much performance on tasks from RT-1 dataset

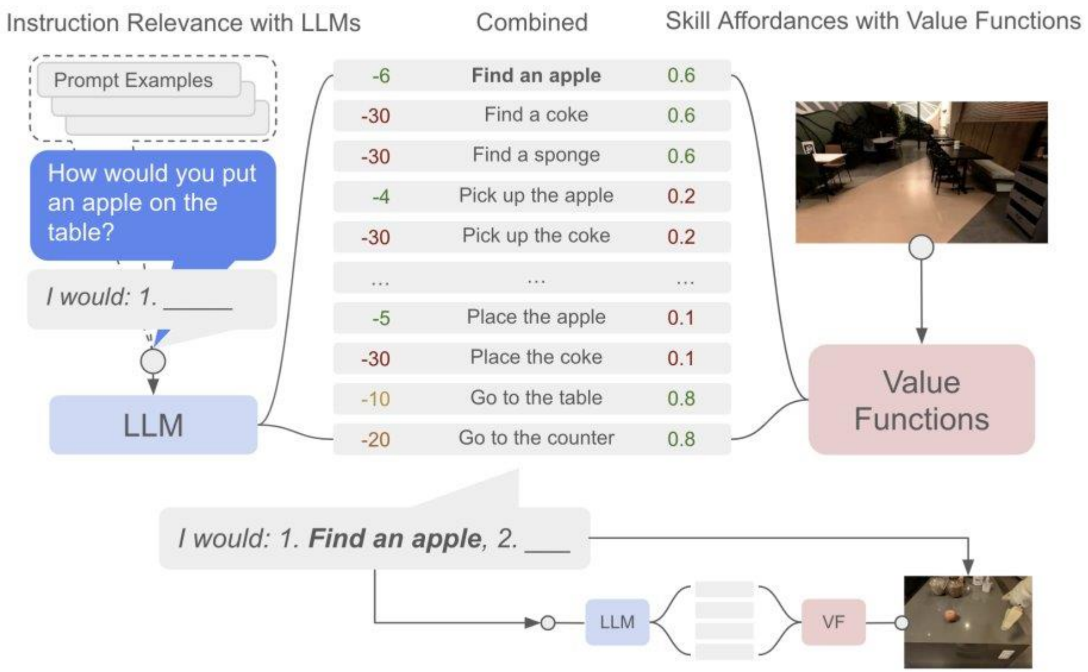
- RT-1 improves by **17% (almost 2x)** on the bin-picking eval when trained on heterogeneous data



Models	Training Data	Classroom eval	Bin-picking eval
RT-1	Kuka bin-picking data + EDR data	90(-2)	**39(+17)**
RT-1	EDR only data	92	22
RT-1	Kuka bin-picking only data	0	0

Results: Long Horizon Tasks

- RT-1 was evaluated on long horizon tasks in real kitchen environments
- SayCan was used as high-level planner for the long horizon tasks
- RT-1 achieves **67%** execution accuracy in both kitchen environments
- BC-Z performs well on kitchen 1 but does not generalize well to kitchen 2

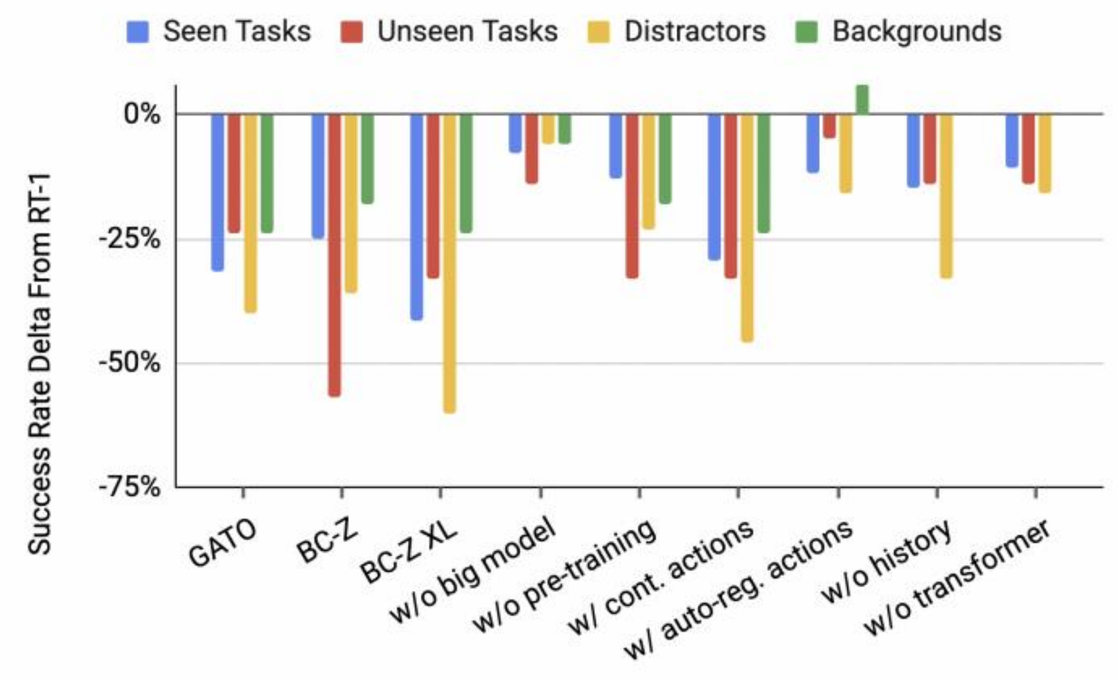


	SayCan tasks in Kitchen1		SayCan tasks in Kitchen2	
	Planning	Execution	Planning	Execution
Original SayCan (Ahn et al., 2022)*	73	47	-	-
SayCan w/ Gato (Reed et al., 2022)	87	33	87	0
SayCan w/ BC-Z (Jang et al., 2021)	87	53	87	13
SayCan w/ RT-1 (ours)	87	**67**	87	**67**

Ablations

- Pre-training EfficientNet-B3 on ImageNet is crucial for generalization
- Switching from continuous action space to discrete action space improve overall performance significantly by capturing multi-modal action distributions

Model	Seen Tasks	Unseen Tasks	Distractors				Backgrounds	Inference Time (ms)
			All	Easy	Medium	Hard	All	
Gato (Reed et al., 2022)	65 (-32)	52 (-24)	43 (-40)	71	44	29	35 (-24)	129
BC-Z (Jang et al., 2021)	72 (-25)	19 (-57)	47 (-36)	100	67	7	41 (-18)	5.3
BC-Z XL	56 (-41)	43 (-33)	23 (-60)	57	33	0	35 (-24)	5.9
RT-1 (ours)	**97**	**76**	**83**	100	100	64	**59**	15
RT-1 w/o big model	89 (-8)	62 (-14)	77 (-6)	100	100	50	53 (-6)	13.5
RT-1 w/o pre-training	84 (-13)	43 (-33)	60 (-23)	100	67	36	41 (-18)	15
RT-1 w/ continuous actions	68 (-29)	43 (-33)	37 (-46)	71	67	0	35 (-24)	16
RT-1 w/ auto-regressive actions	85 (-12)	71 (-5)	67 (-16)	100	78	43	**65 (+6)**	36
RT-1 w/o history	82 (-15)	62 (-14)	50 (-33)	71	89	14	**59 (+0)**	15
RT-1 w/o Transformer	86 (-13)	62 (-14)	67 (-16)	100	100	29	**59 (+0)**	26

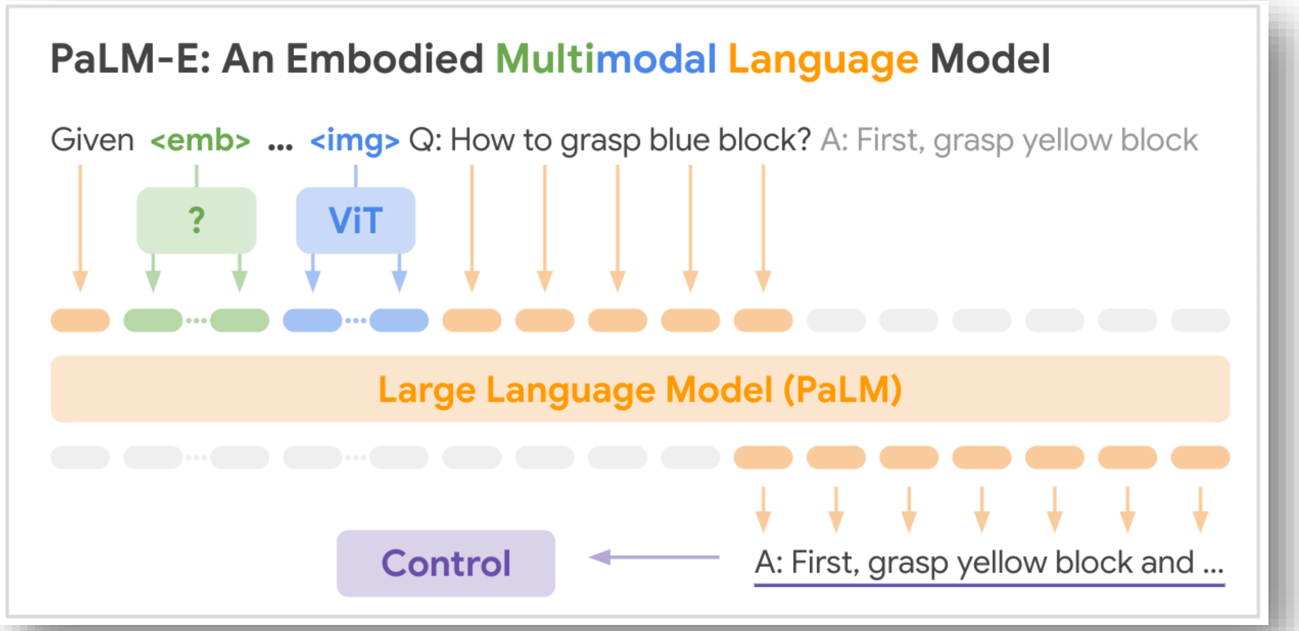


PALM-E

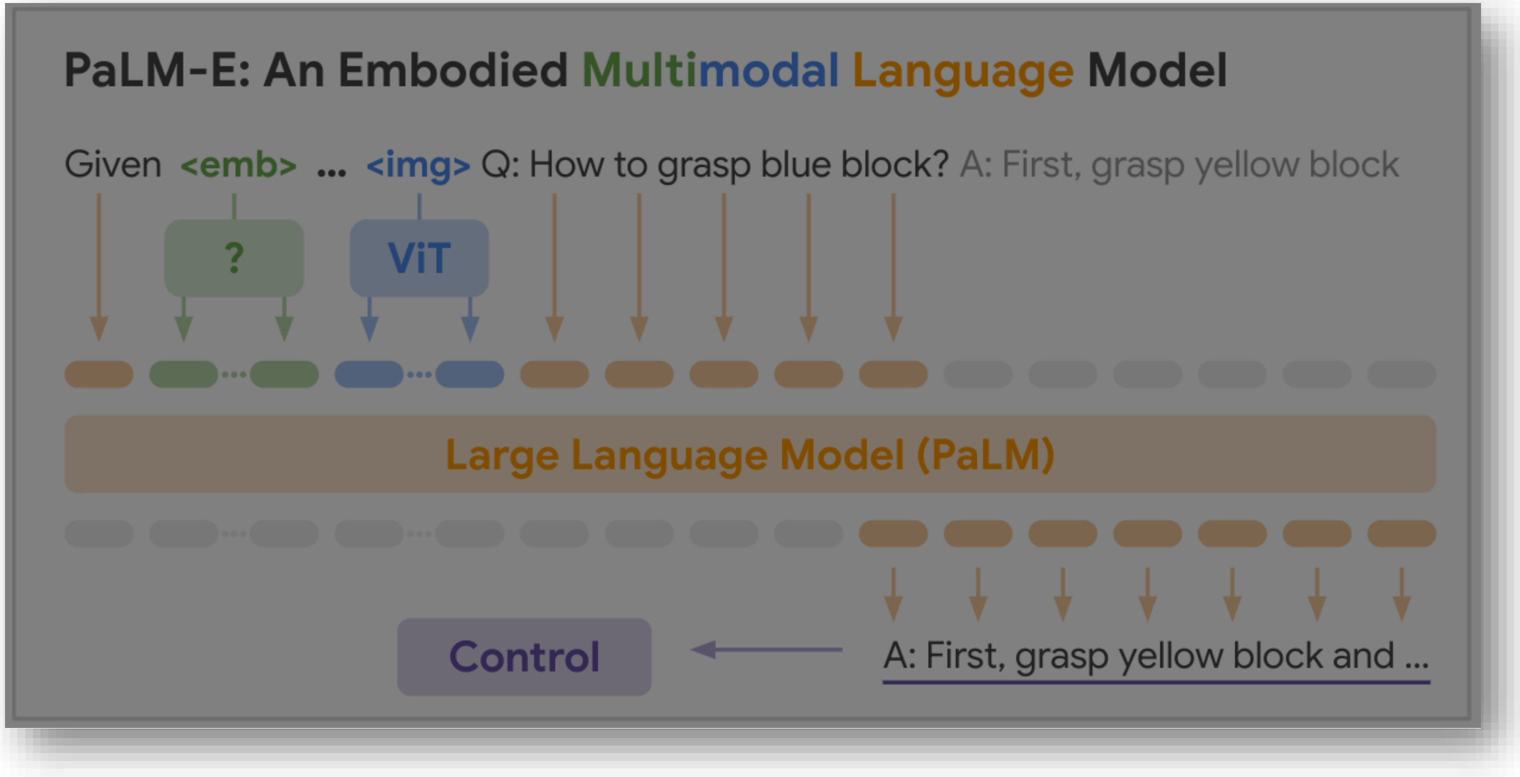
[1] Robotics at Google [2] Technische Universität Berlin [3] Google Research

Problem Statement

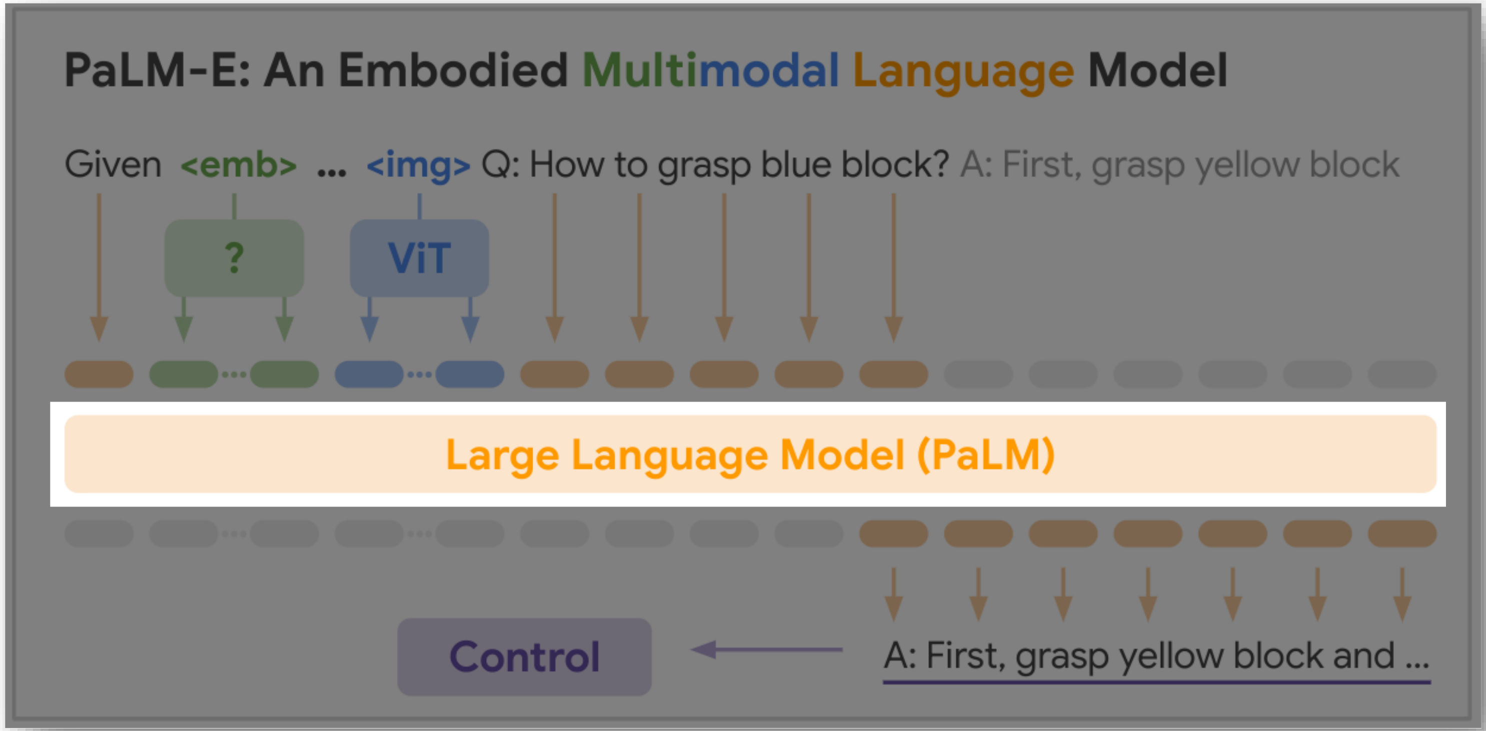
- RT-1 can follow simple language commands
- However, it suffers in complex scene/text command understanding
- Solution: Feed multimodal data into LLM to break down complex tasks for controllers like RT-1



Architecture

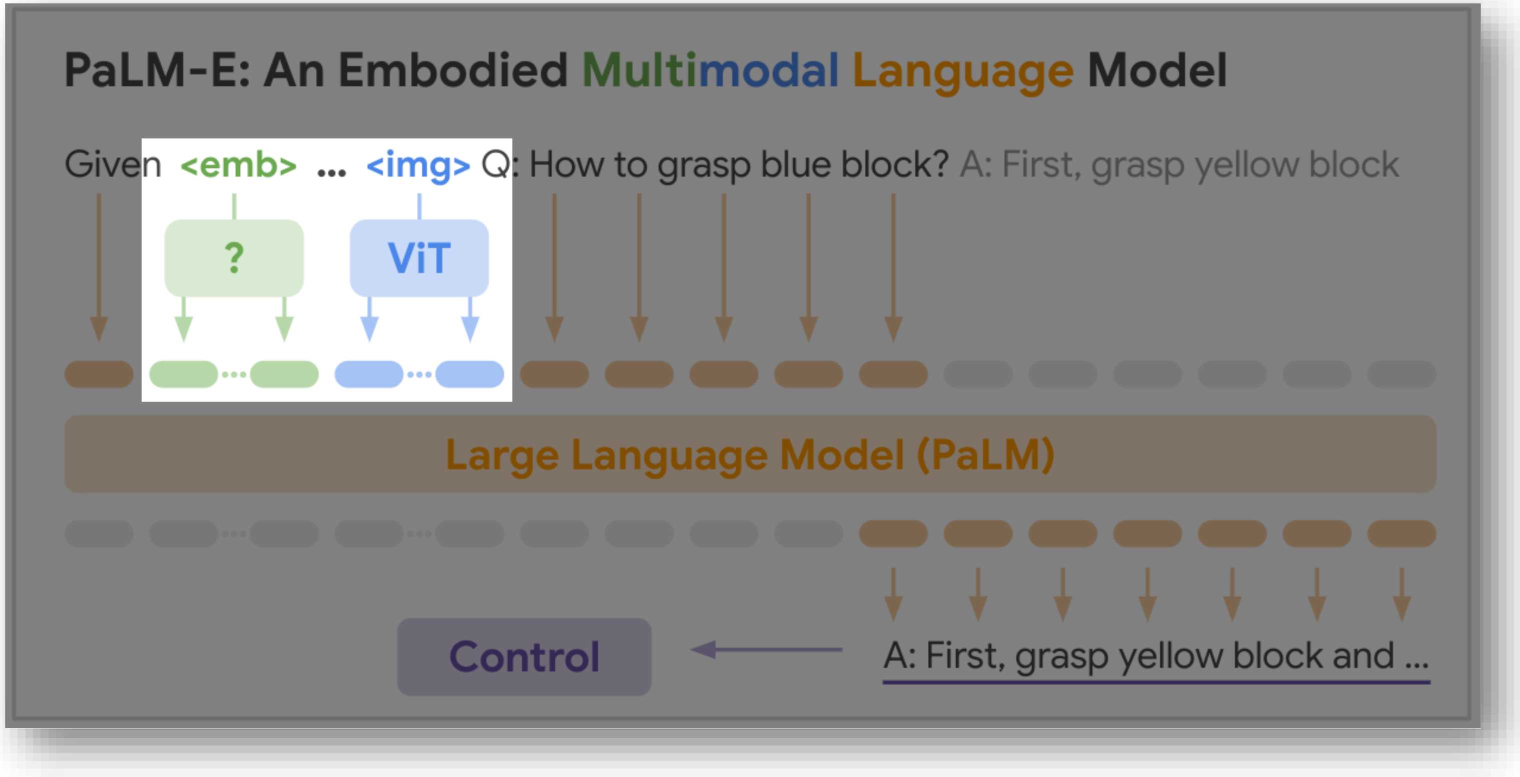


Architecture

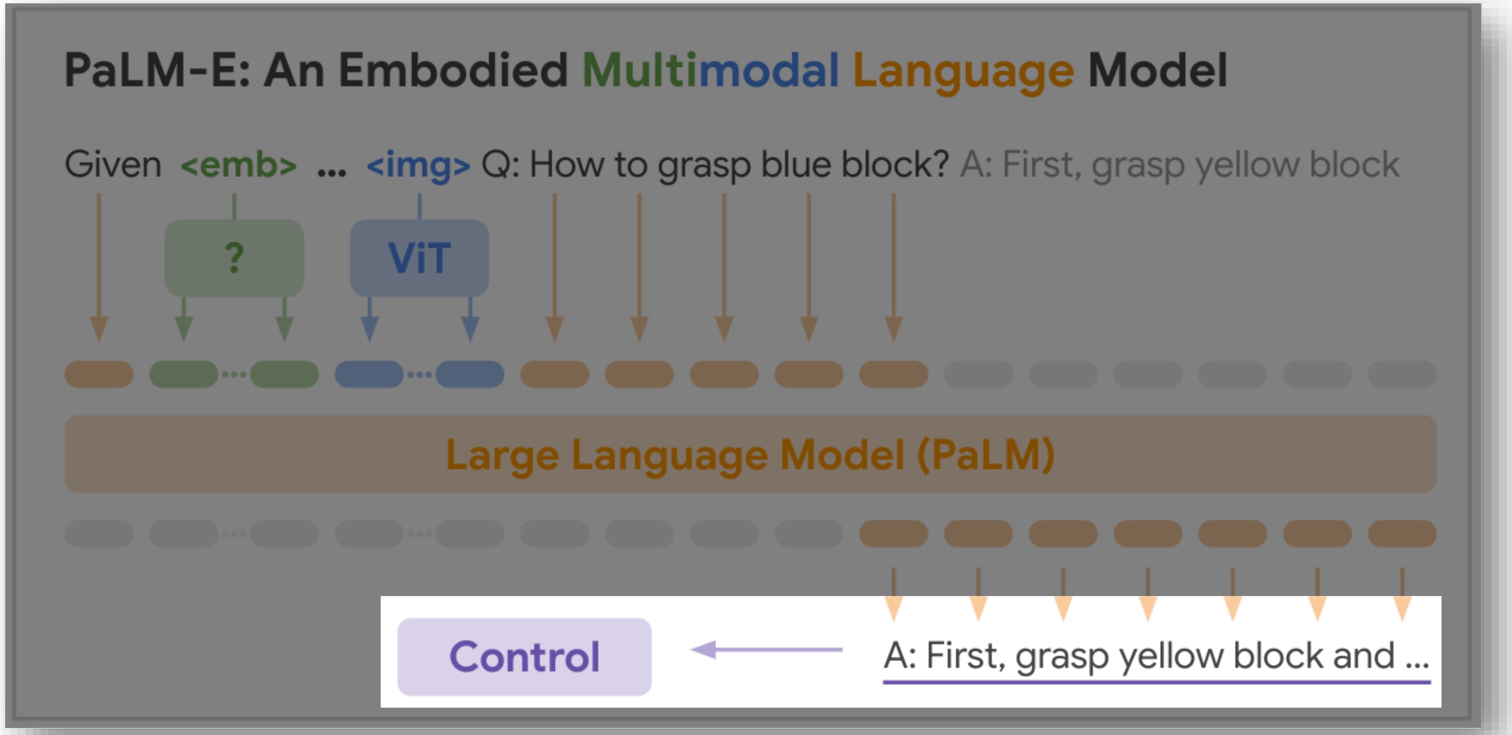


Architecture

Interleaved embeddings



Architecture



RT-1, Gato, etc.

Multimodal Input Representations

Observation space (image, robot joint values, object states, etc.)

$$\phi : \mathcal{O} \rightarrow \mathcal{X}^q$$

q vectors in the LLM embedding space

Multimodal Input Representations

Vision Transformer

ϕ_{ViT}

1) Feed image into ViT
2) Learned affine transform to match LLM dims

Object-Centric

Given GT object location masks, run ϕ_{ViT} on each masked image

$$\phi : \mathcal{O} \rightarrow \mathcal{X}^q$$

State Estimation Vector

s: state of objects in the scene

$\phi_{state} = MLP$
$x = \phi_{state(s)}$

Object Scene Representation Transformer (OSRT)

1) OSRT (ϕ_{OSRT}) gets neural scene representation of each object
2) MLP to match dims

Training

<u>LLM</u>
- Pretrained vs. not
- Frozen vs. unfrozen

<u>ViT</u>
- Pretrained vs not

<u>MLP or Affine transforms</u>
- Always trained

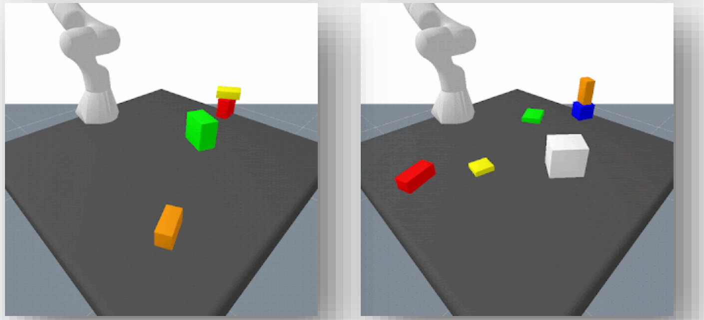
Environments

Task-and-Motion-Planning (TAMP)

- q1-q4: Questions about objects in the scene
- p1-p2: Question for how to generate a plan

- Test different input representations

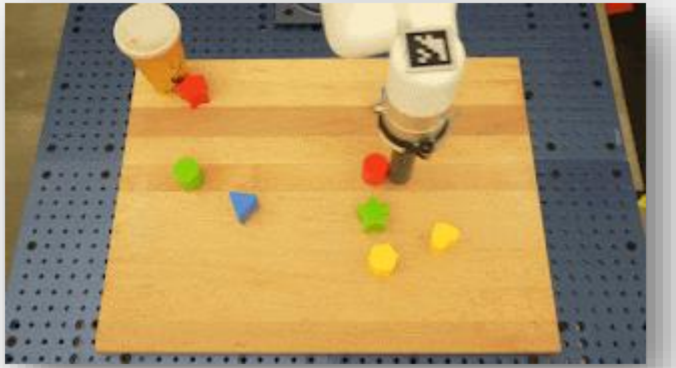
NOTE: Never executed on a robot!



Language-Table

- Pushing objects on a table
- Very diverse language commands

- ViT is used as input representation



Mobile Manipulation

- Navigating a kitchen
  - Similar to RT-1 environment

- ViT is the input representation

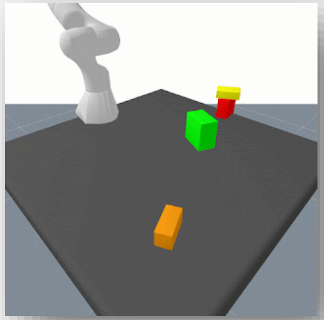
Environments

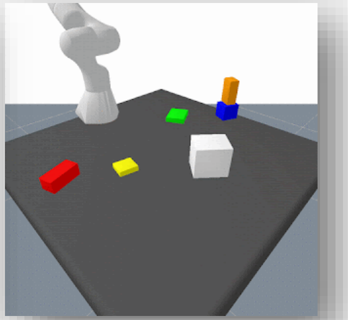
Task-and-Motion-Planning (TAMP)

- q1-q4: Questions about objects in the scene
- p1-p2: Question for how to generate a plan

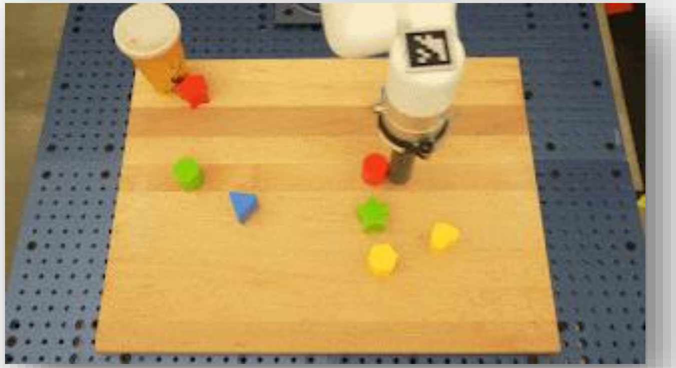
- Test different input representations

NOTE: Never executed on a robot!

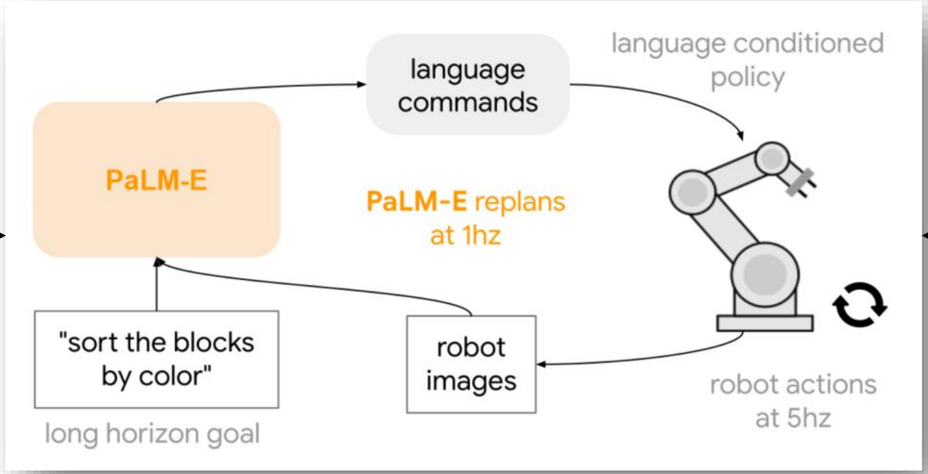




Language-Table



Mobile Manipulation



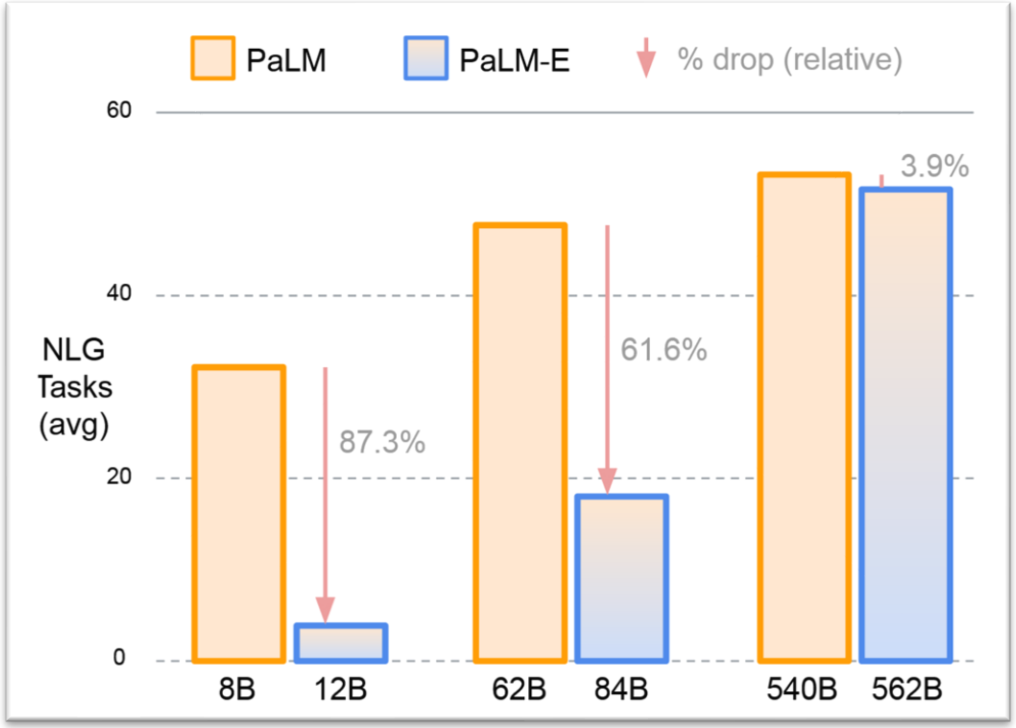
Experiments and Results

1. Performance on VL/language tasks?
   - PaLM-E is essentially PaLM + ViT. How does it do in VL tasks?
   - When PaLM-E is trained, is language performance retained?
2. Input Representations
   - How do different input representations affect performance?
3. Positive Transfer
   - Does co-training on all robot data + VL data improve performance?
4. Does it work on real robots?

Visual-Language/Language Results

Model	VQAv2 test-dev	VQAv2 test-std	OK-VQA val	COCO Karpathy test
Generalist (one model)				
PaLM-E-12B	76.2	-	55.5	135.0
PaLM-E-562B	80.0	-	$\boxed{\textbf{66.1}}$	138.7
Task-specific finetuned models				
Flamingo (Alayrac et al., 2022)	82.0	82.1	57.8†	138.1
PaLI (Chen et al., 2022)	84.3	84.3	64.5	149.1
PaLM-E-12B	77.7	77.9	60.1	136.0
PaLM-E-66B	-	-	62.9	-
PaLM-E-84B	80.5	-	63.3	138.0
Generalist (one model), with frozen LLM				
(Tsimpoukelli et al., 2021)	48.4	-	-	-
PaLM-E-12B frozen	70.3	-	51.5	128.0

SOTA performance on OK-VQA



Bigger model → less forgetting

Input Representation Results - TAMP

	ϕ	LLM pre-trained	q_1	q_2	q_3	q_4	p_1	p_2
3 - 5 objects	SayCan (w/ oracle affordances)	✓	-	-	-	-	38.7	33.3
	state	✗	100.0	99.3	98.5	99.8	97.2	95.5
	state	✓(unfrozen)	100.0	98.8	100.0	97.6	97.7	95.3
	state	✓	100.0	98.4	99.7	98.5	97.6	96.0
	state (w/o entity referrals)	✓	100.0	98.8	97.5	98.1	94.6	90.3
	ViT + TL (obj. centric)	✓	99.6	98.7	98.4	96.8	9.2	94.5
	ViT + TL (global)	✓	-	60.7	90.8	94.3	70.7	69.2
	ViT-4B (global)	✓	-	98.2	99.4	99.0	96.0	93.4
	ViT-4B generalist	✓	-	97.1	100.0	98.9	97.5	95.2
	OSRT	✓	99.6	99.1	100.0	98.8	98.1	95.7
6 objects	state	✗	20.4	39.2	71.4	85.2	56.5	34.3
	state	✓	100.0	98.5	94.0	89.3	95.3	81.4
	state (w/o entity referrals)	✓	77.7	83.7	93.6	91.0	81.2	57.1
8 objects	state	✗	18.4	27.1	38.1	87.5	24.6	6.7
	state	✓	100.0	98.3	95.3	89.8	91.3	89.3
	state (w/o entity referrals)	✓	60.0	67.1	94.1	81.2	49.3	49.3
6 objects + OOD tasks	state (8B LLM)	✗	-	0	0	72.0	0	0
	state (8B LLM)	✓	-	49.3	89.8	68.5	28.2	15.7
	state (62B LLM)	✓	-	48.7	92.5	88.1	40.0	30.0

Table: PaLM-E trained on just TAMP environment

Input Representation Results - TAMP

	ϕ	LLM pre-trained	q_1	q_2	q_3	q_4	p_1	p_2
3 - 5 objects	SayCan (w/ oracle affordances)	✓	-	-	-	-	38.7	33.3
	state	✗	100.0	99.3	98.5	99.8	97.2	95.5
	state	✓(unfrozen)	100.0	98.8	100.0	97.6	97.7	95.3
	state	✓	100.0	98.4	99.7	98.5	97.6	96.0
	state (w/o entity referrals)	✓	100.0	98.8	97.5	98.1	94.6	90.3
	ViT + TL (obj. centric)	✓	99.6	98.7	98.4	96.8	9.2	94.5
	ViT + TL (global)	✓	-	60.7	90.8	94.3	70.7	69.2
	ViT-4B (global)	✓	-	98.2	99.4	99.0	96.0	93.4
	ViT-4B generalist	✓	-	97.1	100.0	98.9	97.5	95.2
	OSRT	✓	99.6	99.1	100.0	98.8	98.1	95.7
6 objects	state	✗	20.4	39.2	71.4	85.2	56.5	34.3
	state	✓	100.0	98.5	94.0	89.3	95.3	81.4
	state (w/o entity referrals)	✓	77.7	83.7	93.6	91.0	81.2	57.1
8 objects	state	✗	18.4	27.1	38.1	87.5	24.6	6.7
	state	✓	100.0	98.3	95.3	89.8	91.3	89.3
	state (w/o entity referrals)	✓	60.0	67.1	94.1	81.2	49.3	49.3
6 objects + OOD tasks	state (8B LLM)	✗	-	0	0	72.0	0	0
	state (8B LLM)	✓	-	49.3	89.8	68.5	28.2	15.7
	state (62B LLM)	✓	-	48.7	92.5	88.1	40.0	30.0

Table: PaLM-E trained on just TAMP environment

- For 3-5 objects: Results are pretty similar for different input representations

Input Representation Results - TAMP

	ϕ	LLM pre-trained	q_1	q_2	q_3	q_4	p_1	p_2
3 - 5 objects	SayCan (w/ oracle affordances)	✓	-	-	-	-	38.7	33.3
	state	✗	100.0	99.3	98.5	99.8	97.2	95.5
	state	✓(unfrozen)	100.0	98.8	100.0	97.6	97.7	95.3
	state	✓	100.0	98.4	99.7	98.5	97.6	96.0
	state (w/o entity referrals)	✓	100.0	98.8	97.5	98.1	94.6	90.3
	ViT + TL (obj. centric)	✓	99.6	98.7	98.4	96.8	9.2	94.5
	ViT + TL (global)	✓	-	60.7	90.8	94.3	70.7	69.2
	ViT-4B (global)	✓	-	98.2	99.4	99.0	96.0	93.4
	ViT-4B generalist	✓	-	97.1	100.0	98.9	97.5	95.2
	OSRT	✓	99.6	99.1	100.0	98.8	98.1	95.7
6 objects	state	✗	20.4	39.2	71.4	85.2	56.5	34.3
	state	✓	100.0	98.5	94.0	89.3	95.3	81.4
	state (w/o entity referrals)	✓	77.7	83.7	93.6	91.0	81.2	57.1
8 objects	state	✗	18.4	27.1	38.1	87.5	24.6	6.7
	state	✓	100.0	98.3	95.3	89.8	91.3	89.3
	state (w/o entity referrals)	✓	60.0	67.1	94.1	81.2	49.3	49.3
6 objects + OOD tasks	state (8B LLM)	✗	-	0	0	72.0	0	0
	state (8B LLM)	✓	-	49.3	89.8	68.5	28.2	15.7
	state (62B LLM)	✓	-	48.7	92.5	88.1	40.0	30.0

Table: PaLM-E trained on <u>just</u> TAMP environment

- For 3-5 objects: Results are pretty similar for different input representations

	Object-centric	LLM pre-train	Embodied VQA q_1	q_2	q_3	q_4	Planning p_1	p_2
SayCan (oracle afford.) (Ahn et al., 2022)		✓	-	-	-	-	38.7	33.3
PaLI (zero-shot) (Chen et al., 2022)		✓	-	0.0	0.0	-	-	-
PaLM-E (ours) w/ input enc:								
State	✓(GT)	✗	99.4	89.8	90.3	88.3	45.0	46.1
State	✓(GT)	✓	**100.0**	96.3	95.1	93.1	55.9	49.7
ViT + TL	✓(GT)	✓	34.7	54.6	74.6	91.6	24.0	14.7
ViT-4B single robot	✗	✓	-	45.9	78.4	92.2	30.6	32.9
ViT-4B full mixture	✗	✓	-	70.7	93.4	92.1	74.1	74.6
OSRT (no VQA)	✓	✓	-	-	-	-	71.9	75.1
OSRT	✓	✓	99.7	**98.2**	**100.0**	**93.7**	**82.5**	**76.2**

Table: PaLM-E trained on <u>just</u> 1% of TAMP data

Input Representation Results - TAMP

	ϕ	LLM pre-trained	q_1	q_2	q_3	q_4	p_1	p_2
3 - 5 objects	SayCan (w/ oracle affordances)	✓	-	-	-	-	38.7	33.3
	state	✗	100.0	99.3	98.5	99.8	97.2	95.5
	state	✓(unfrozen)	100.0	98.8	100.0	97.6	97.7	95.3
	state	✓	100.0	98.4	99.7	98.5	97.6	96.0
	state (w/o entity referrals)	✓	100.0	98.8	97.5	98.1	94.6	90.3
	ViT + TL (obj. centric)	✓	99.6	98.7	98.4	96.8	9.2	94.5
	ViT + TL (global)	✓	-	60.7	90.8	94.3	70.7	69.2
	ViT-4B (global)	✓	-	98.2	99.4	99.0	96.0	93.4
	ViT-4B generalist	✓	-	97.1	100.0	98.9	97.5	95.2
	OSRT	✓	99.6	99.1	100.0	98.8	98.1	95.7
6 objects	state	✗	20.4	39.2	71.4	85.2	56.5	34.3
	state	✓	100.0	98.5	94.0	89.3	95.3	81.4
	state (w/o entity referrals)	✓	77.7	83.7	93.6	91.0	81.2	57.1
8 objects	state	✗	18.4	27.1	38.1	87.5	24.6	6.7
	state	✓	100.0	98.3	95.3	89.8	91.3	89.3
	state (w/o entity referrals)	✓	60.0	67.1	94.1	81.2	49.3	49.3
6 objects + OOD tasks	state (8B LLM)	✗	-	0	0	72.0	0	0
	state (8B LLM)	✓	-	49.3	89.8	68.5	28.2	15.7
	state (62B LLM)	✓	-	48.7	92.5	88.1	40.0	30.0

Table: PaLM-E trained on <u>just</u> TAMP environment

- For 3-5 objects: Results are pretty similar for different input representations

	Object-centric	LLM pre-train	Embodied VQA q_1	q_2	q_3	q_4	Planning p_1	p_2
SayCan (oracle afford.) (Ahn et al., 2022)		✓	-	-	-	-	38.7	33.3
PaLI (zero-shot) (Chen et al., 2022)		✓	-	0.0	0.0	-	-	-
PaLM-E (ours) w/ input enc:								
State	✓(GT)	✗	99.4	89.8	90.3	88.3	45.0	46.1
State	✓(GT)	✓	**100.0**	96.3	95.1	93.1	55.9	49.7
ViT + TL	✓(GT)	✓	34.7	54.6	74.6	91.6	24.0	14.7
ViT-4B single robot	✗	✓	-	45.9	78.4	92.2	30.6	32.9
ViT-4B full mixture	✗	✓	-	70.7	93.4	92.1	74.1	74.6
OSRT (no VQA)	✓	✓	-	-	-	-	71.9	75.1
OSRT	✓	✓	99.7	**98.2**	**100.0**	**93.7**	**82.5**	**76.2**

Table: PaLM-E trained on <u>just</u> 1% of TAMP data

When 1% of data is TAMP:

- OSRT seems to work the best

Does Co-Training Improve Performance?

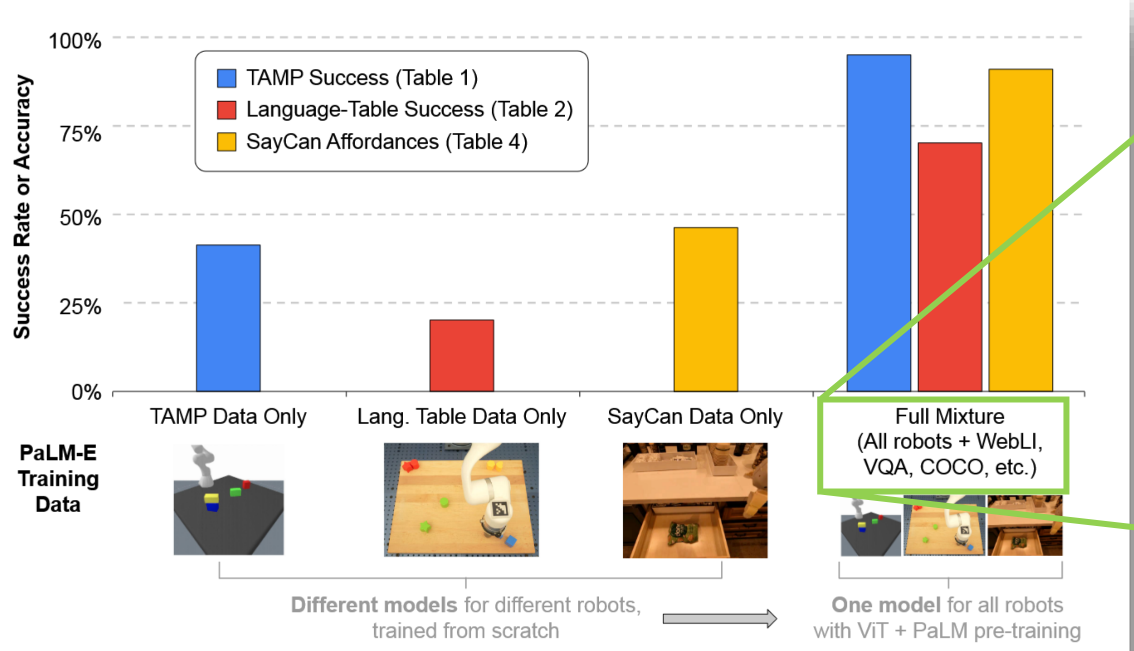
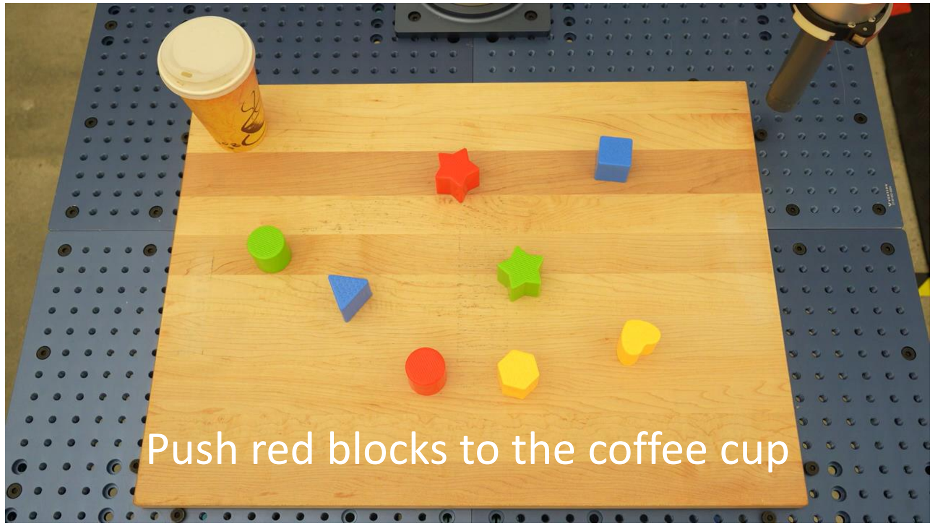


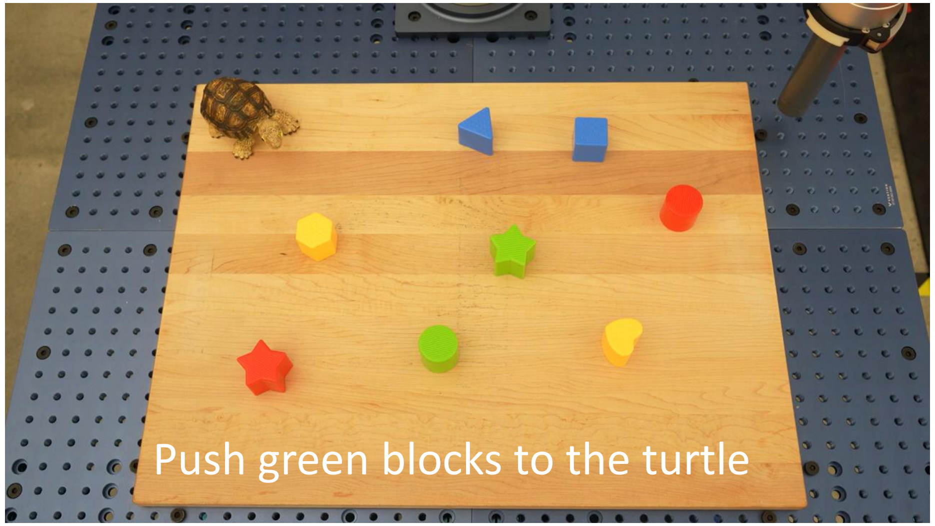
Figure 3: Overview of *transfer* learning demonstrated by PaLM-E: across three different robotics domains, using PaLM and ViT pretraining together with the full mixture of robotics and general visual-language data provides a significant performance increase compared to only training on the respective in-domain data. See Tab. 1, Fig. 4, Tab. 2, Tab. 4 for additional data in each domain.

Dataset in full mixture	Sampling frequency	%
Webli (Chen et al., 2022)	100	52.4
VQ^{2}A (Changpinyo et al., 2022)	25	13.1
VQG (Changpinyo et al., 2022)	10	5.2
CC3M (Sharma et al., 2018)	25	13.1
Object Aware (Piergiovanni et al., 2022)	10	5.2
OKVQA (Marino et al., 2019)	1	0.5
VQAv2 (Goyal et al., 2017)	1	0.5
COCO (Chen et al., 2015)	1	0.5
Wikipedia text	1	0.5
(robot) Mobile Manipulator, real	6	3.1
(robot) Language Table (Lynch et al., 2022), sim and real	8	4.2
(robot) TAMP, sim	3	1.6

Table 6: Dataset sampling frequency and ratio for the "full mixture" referred to in experiments.

Does it work?





Zero-shot Results

Going back…

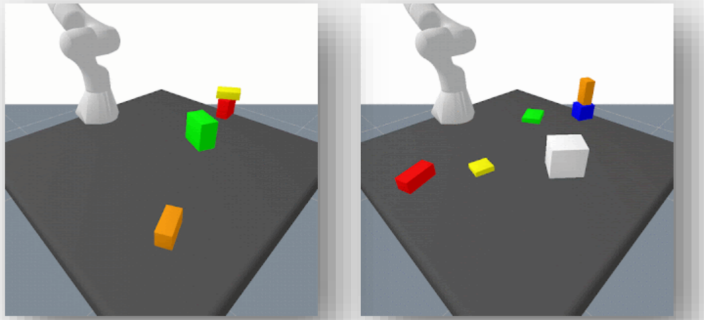
Both use ViT for input representations! Why?

Task-and-Motion-Planning (TAMP)

- q1-q4: Questions about objects in the scene
- p1-p2: Question for how to generate a plan

- Test different input representations

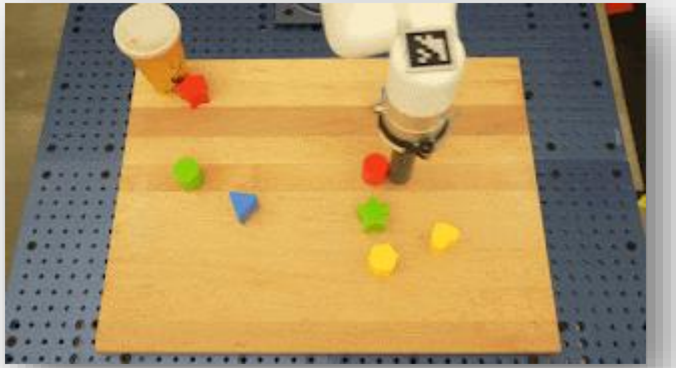
NOTE: Never executed on a robot!



Language-Table

- Pushing objects on a table
- Very diverse language commands

- ViT is used as input representation



Mobile Manipulation

- Navigating a kitchen
  - Similar to RT-1 environment

- ViT is the input representation

RT-2

Vision-Language-Action Models Transfer Web Knowledge to Robotic Control

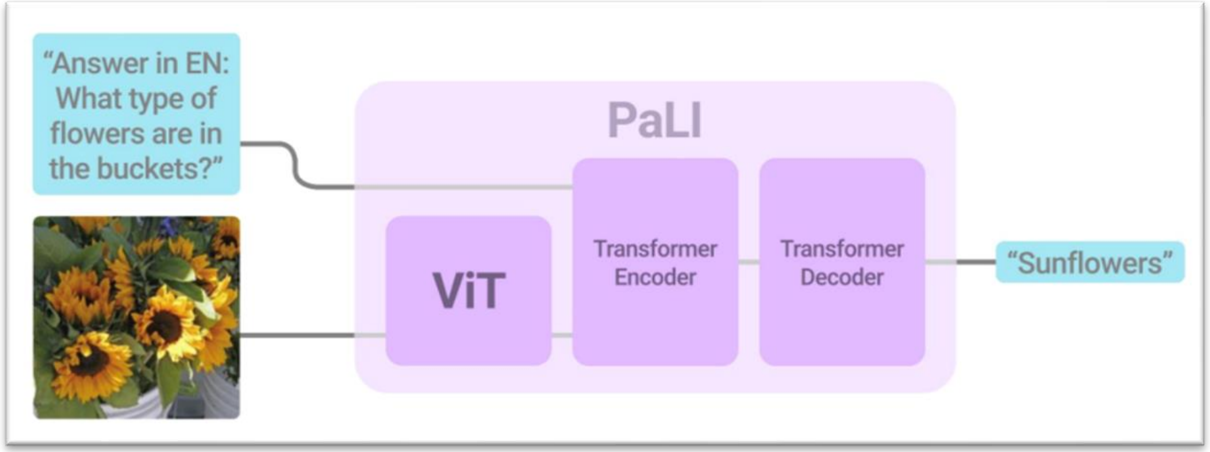
Google DeepMind

2023-8-1

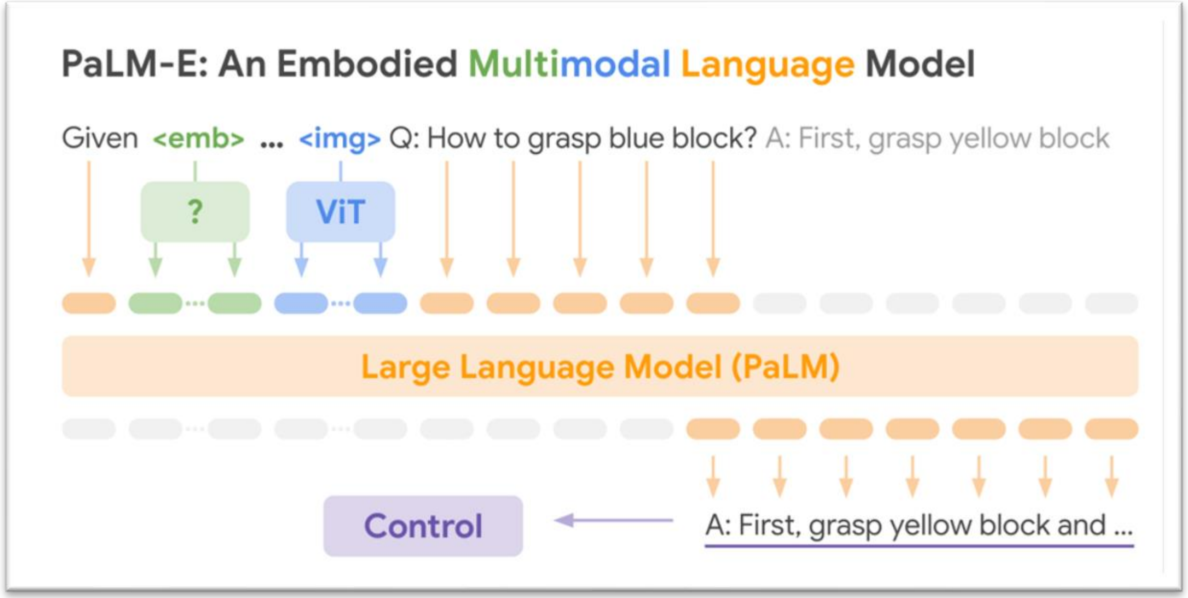
Related Works

- Vision-Language Models
- Pre-training for robotic manipulation

Vision-Language Models



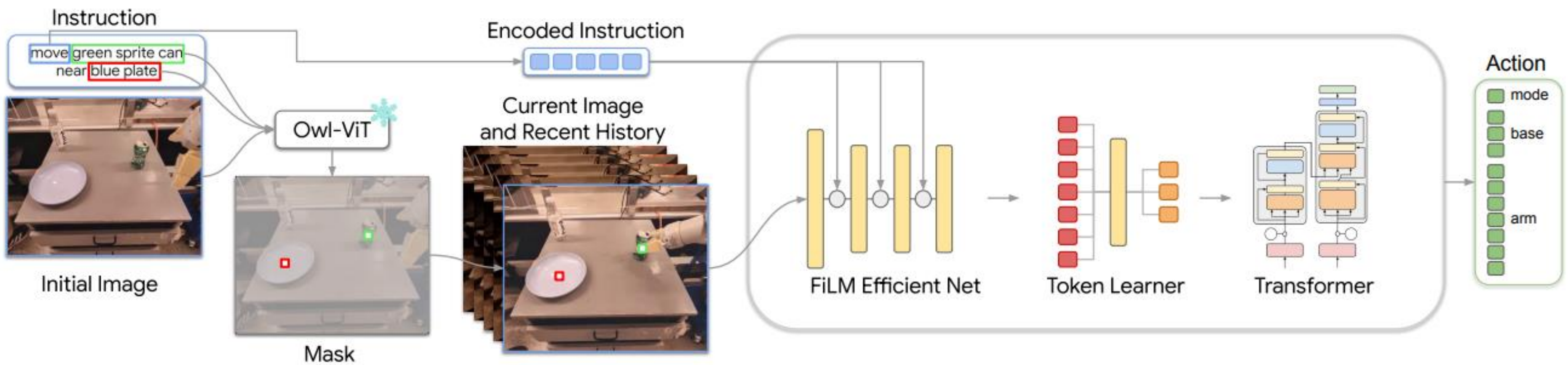
Encoder–Decoder VLM
55B/5B



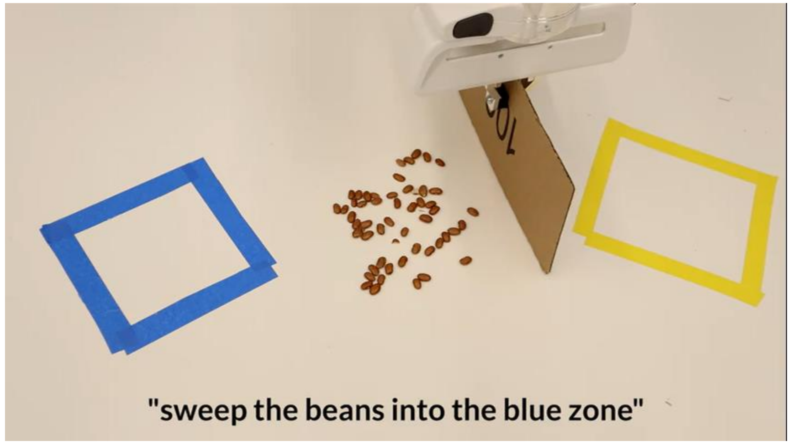
Decoder-only VLM
12B

Pre-training for Robotic Manipulation

- Vision only / Language only / VLMs for high-level tasks
- VLMs for low-level controls
  - E.g. CLIPort and MOO (Constrained)



MOO (Specific instruction structure)



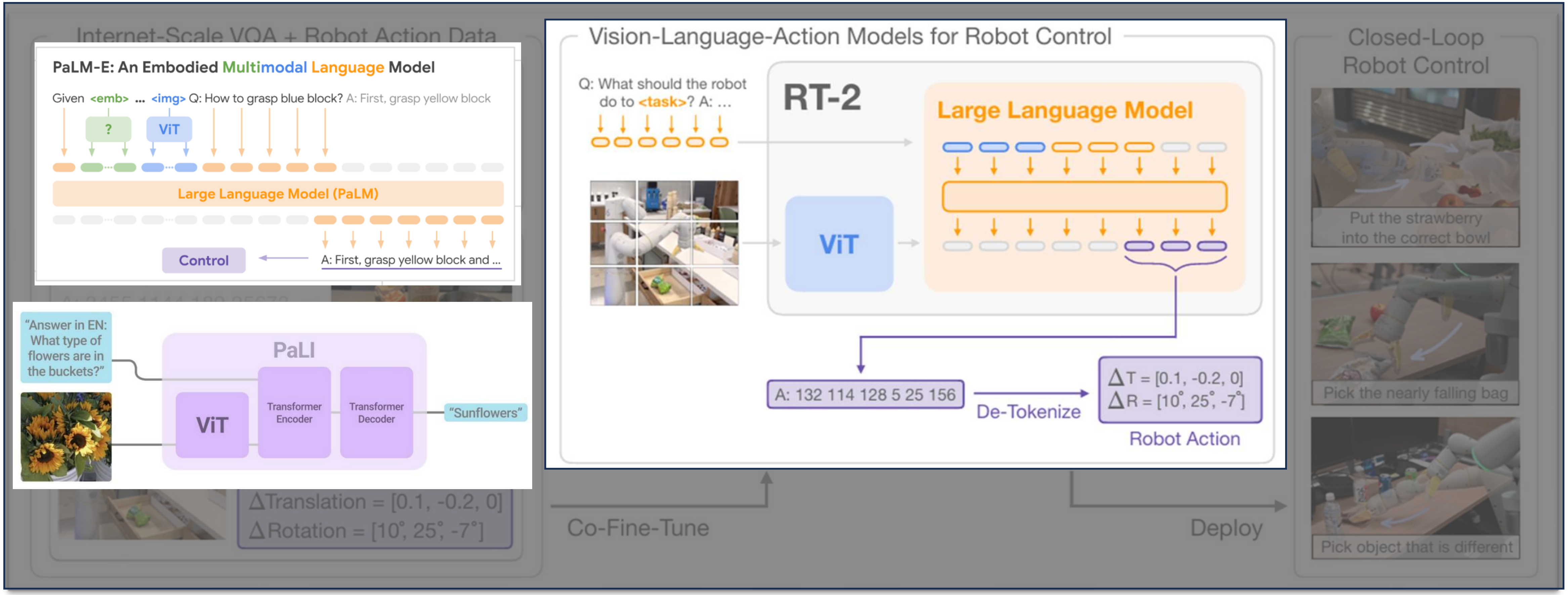
CLIPort
(two-step primitive)

- RT-2 aims at: Using VLMs for low-level controls while avoid having too many constraints.

Problem Statement

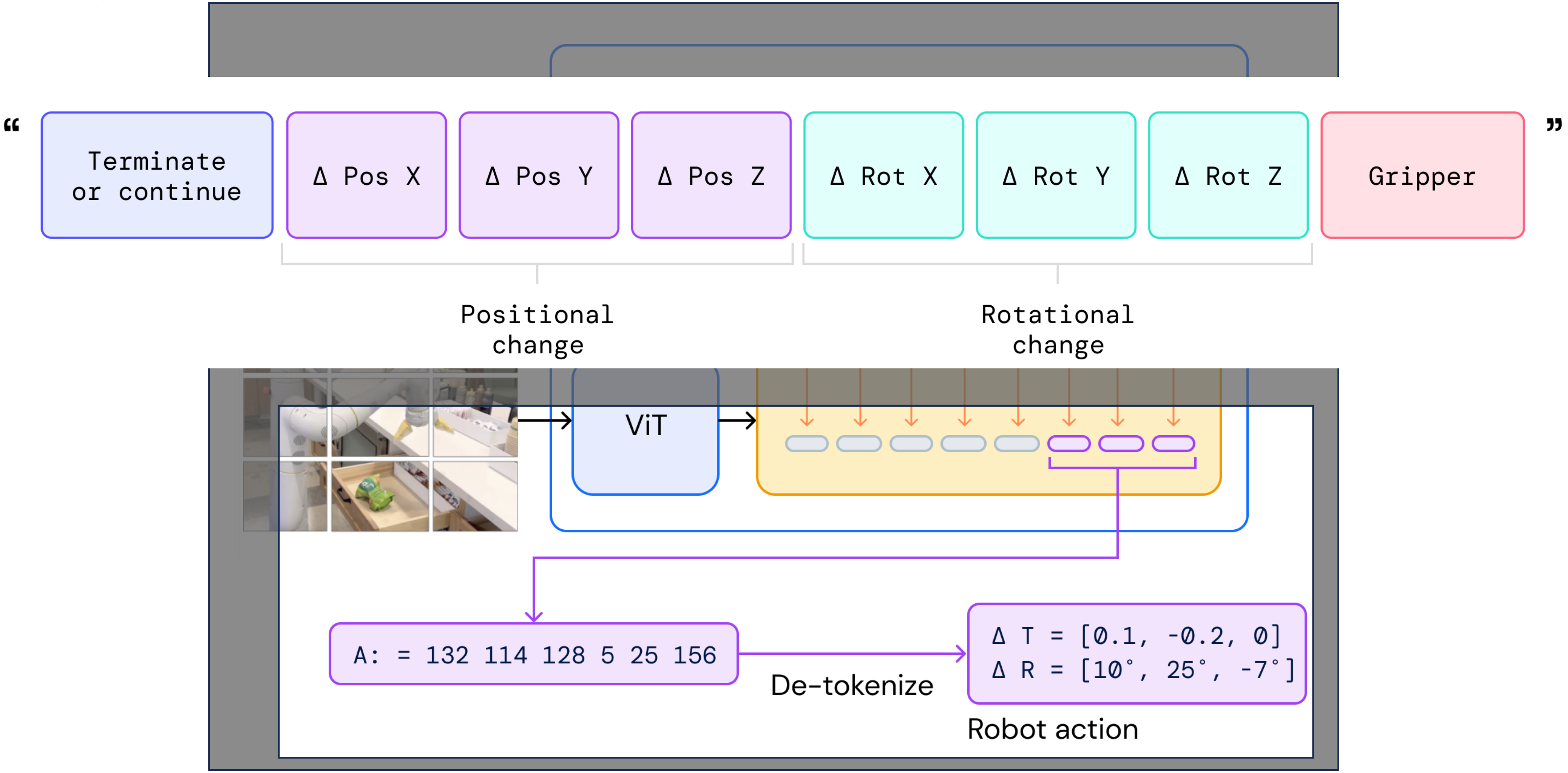
- In this paper we ask:
- Can large pre-trained vision language models be integrated directly into **low-level robotic control** to **boost generalization** and **enable emergent semantic reasoning**?

Approach

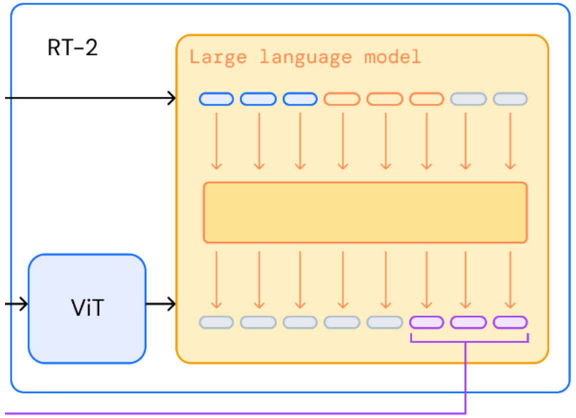


Internet-Scale VQA + Robot Action Data
PaLM-E: An Embodied Multimodal Language Model
Given <emb> ... <img> Q: How to grasp blue block? A: First, grasp yellow block
?
ViT
Large Language Model (PaLM)
Control
A: First, grasp yellow block and ...
"Answer in EN: What type of flowers are in the buckets?"
PaLI
ViT
Transformer Encoder
Transformer Decoder
"Sunflowers"
ΔTranslation = [0.1, -0.2, 0]
ΔRotation = [10°, 25°, -7°]
Vision-Language-Action Models for Robot Control
Q: What should the robot do to <task>? A: ...
RT-2
Large Language Model
ViT
A: 132 114 128 5 25 156
De-Tokenize
ΔT = [0.1, -0.2, 0]
ΔR = [10°, 25°, -7°]
Robot Action
Co-Fine-Tune
Deploy
Closed-Loop Robot Control
Put the strawberry into the correct bowl
Pick the nearly falling bag
Pick object that is different

Approach - Action as text tokens



Approach - Robot action fine-tune



Discretize to 256 bins ← Represent as 256 integers

Continuous Action Space → Associate corresponding tokens

PaLI-X: Integers up 1000 have unique tokens

PaLM-E: No tokens for numbers, just replace 256 least used tokens

" Terminate or continue | Δ Pos X | Δ Pos Y | Δ Pos Z | Δ Rot X | Δ Rot Y | Δ Rot Z | Gripper "

Positional change (Δ Pos X, Δ Pos Y, Δ Pos Z)

Rotational change (Δ Rot X, Δ Rot Y, Δ Rot Z)

During inference time, instead of generating tokens, the VLM is enforced to sample from valid action tokens

Approach – co-fine-tune

Question: Why all robot tasks in the form of VQA?

RT-2-PaLM-E: 66% RT-1 data + 33% PaLM-E data

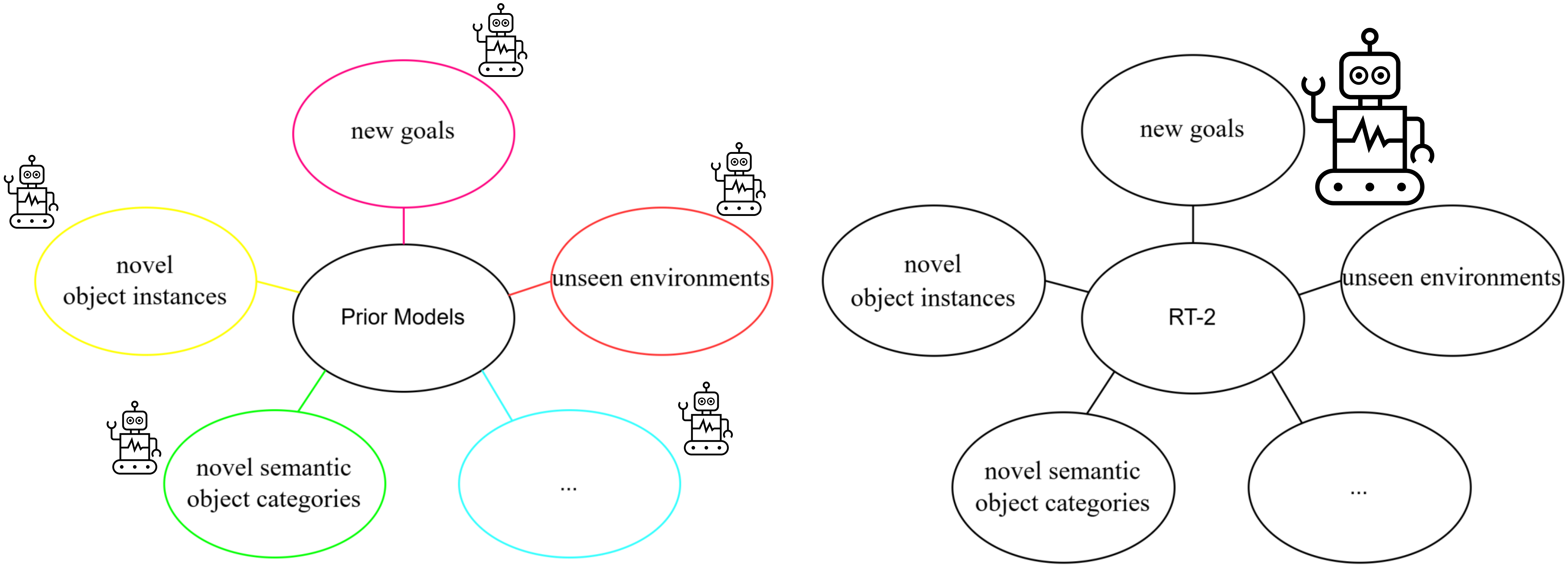
RT-2-PaLI-X: 50% RT-1 data + 50% PaLI-X data

Skill	Count	Description	Example Instruction
Pick `Object`	130	Lift the object off the surface	pick iced tea can
Move `Object` Near `Object`	337	Move the first object near the second	move pepsi can near rxbar blueberry
Place `Object` Upright	8	Place an elongated object upright	place water bottle upright
Knock `Object` Over	8	Knock an elongated object over	knock redbull can over
Open `Drawer`	3	Open any of the cabinet drawers	open the top drawer
Close `Drawer`	3	Close any of the cabinet drawers	close the middle drawer
Place `Object` into `Receptacle`	84	Place an object into a receptacle	place brown chip bag into white bowl
Pick `Object` from `Receptacle` and Place on the Counter	162	Pick an object up from a location and then place it on the counter	pick green jalapeno chip bag from paper bowl and place on counter
Section 6.3 and 6.4 tasks	9	Skills trained for realistic, long instructions	open the large glass jar of pistachios pull napkin out of dispenser grab scooper
Total	744		

Table 1: The list of skills collected for RT-1 together with their descriptions and example instructions.

Generalization in Robot Learning

- Long-standing goal:
  - Have robotic controllers that can broadly succeed in a variety of scenarios

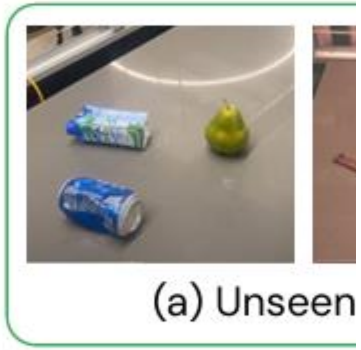


Experiments and Results

- 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?

Experiments and Results

- 1. How d… …tantly, generaliz… …ents?



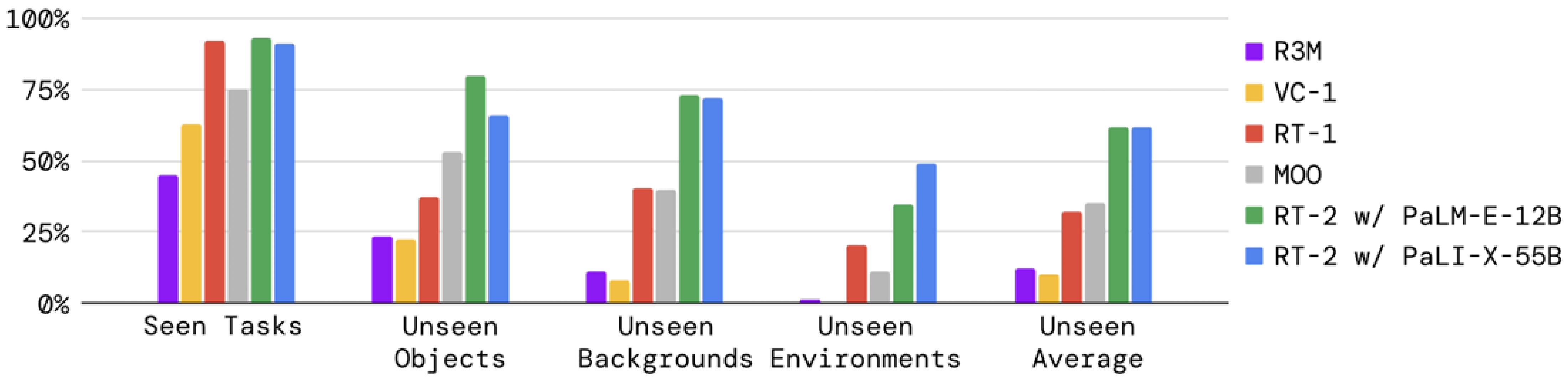
(a) Unseen

n Environments

Task Group	Tasks
Unseen Objects (Easy)	pick banana, move banana near coke can, move orange can near banana, pick oreo, move oreo near apple, move redbull can near oreo, pick pear, pick coconut water, move pear near coconut water, move pepsi can near pear
Unseen Objects (Hard)	pick cold brew can, pick large orange plate, pick chew toy, pick large tennis ball, pick bird ornament, pick fish toy, pick ginger lemon kombucha, pick egg separator, pick wrist watch, pick green sprite can, pick blue microfiber cloth, pick yellow pear, pick pretzel chip bag, pick disinfectant wipes, pick pineapple hint water, pick green cup, pick pickle snack, pick small blue plate, pick small orange rolling pin, pick octopus toy, pick catnip toy
Unseen Backgrounds (Easy)	pick green jalapeno chip bag, pick orange can, pick pepsi can, pick 7up can, pick apple, pick blue chip bag, pick orange, pick 7up can, move orange near sink, pick coke can, pick sponge, pick rxbar blueberry
Unseen Backgrounds (Hard)	pick wrist watch, pick egg separator, pick green sprite can, pick blue microfiber cloth, pick yellow pear, pick pretzel chip bag, pick disinfectant wipes, pick pineapple hint water, pick green cup, pick pickle snack, pick small blue plate, pick small orange rolling pin, pick octopus toy, pick catnip toy, pick swedish fish bag, pick large green rolling pin, pick black sunglasses

Experiments and Results

- 1. How does RT-2 perform on seen tasks and more importantly, generalize over new objects, backgrounds, and environments?



- **R3M** – a strong egocentric, representation-learning baseline that gives good visual features but isn't a full VLM policy.
- **VC-1** – a vision model pre-trained specifically for robotics, used here as a robotics-aware perception baseline.
- **RT-1** – the previous robotics transformer trained only on robot demonstrations, without internet-scale VLM pretraining.
- **MOO** – a two-stage setup where a VLM helps pick the object, but the actual control policy is separate from the VLM.

Experiments and Results

- 2. Can we observe and measure any emergent capabilities of RT-2?

    - An **emergent capability** is a new capability that appears in a model *only after* large-scale pretraining and was *not* explicitly taught or present in smaller-scale models.

    - Transferring knowledge from VLMs

2. Emergent capabilities-Qualitative Evaluations

semantic understanding and basic reasoning

physical understanding

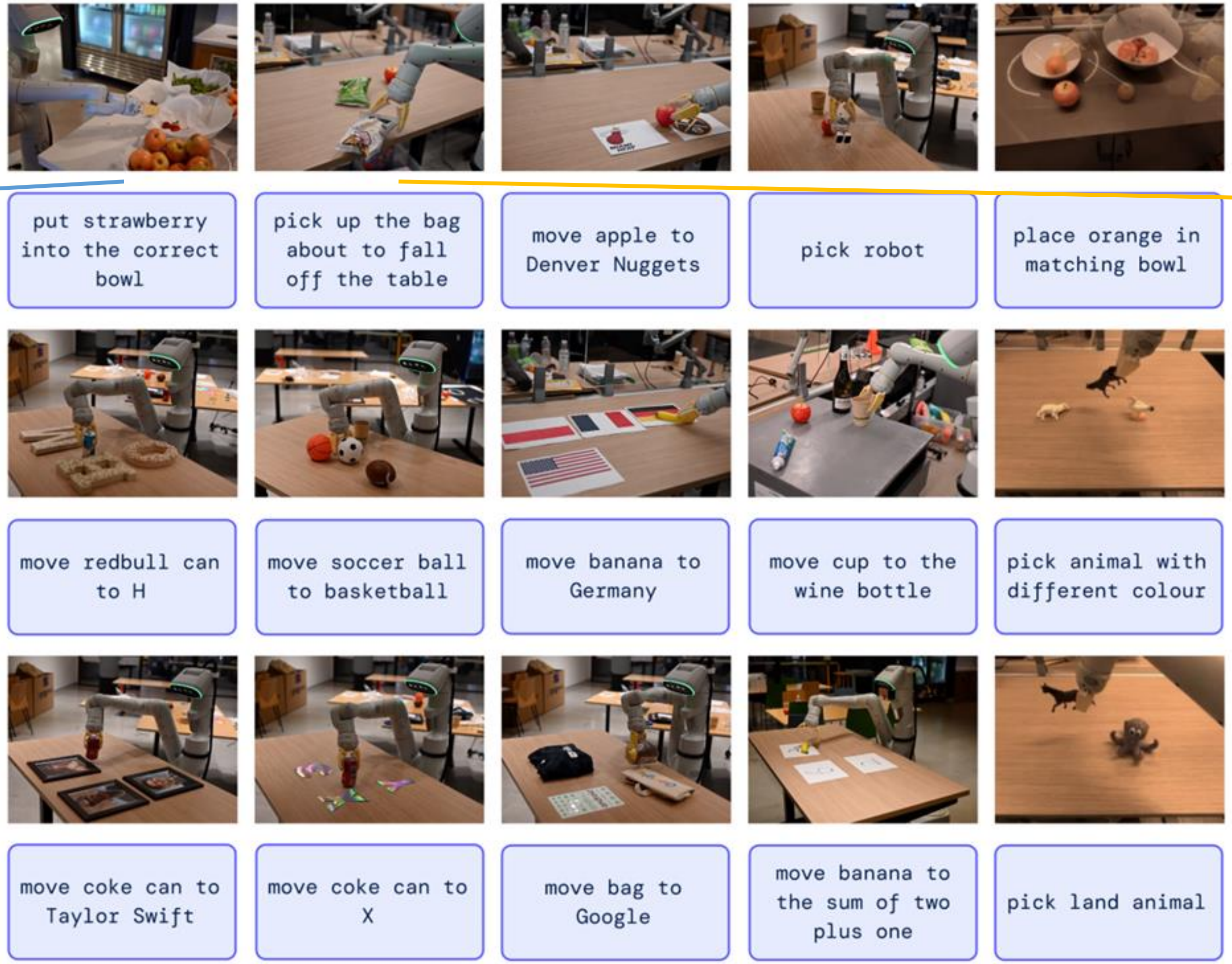


Figure 2 | RT-2 is able to generalize to a variety of real-world situations that require reasoning, symbol understanding, and human recognition. We study these challenging scenarios in detail in Section 4.

2. Emergent capabilities-Quantitative Evaluations

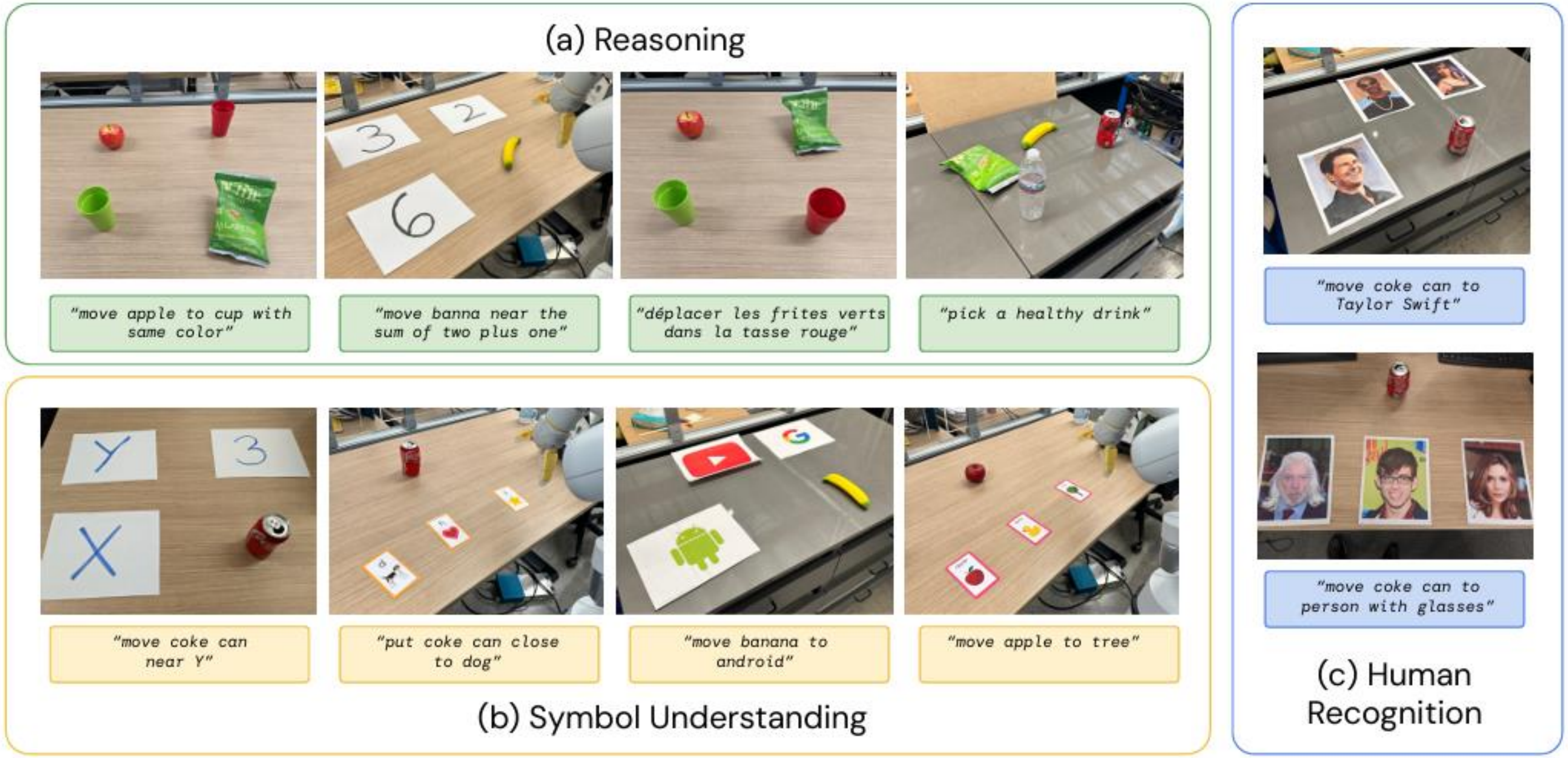
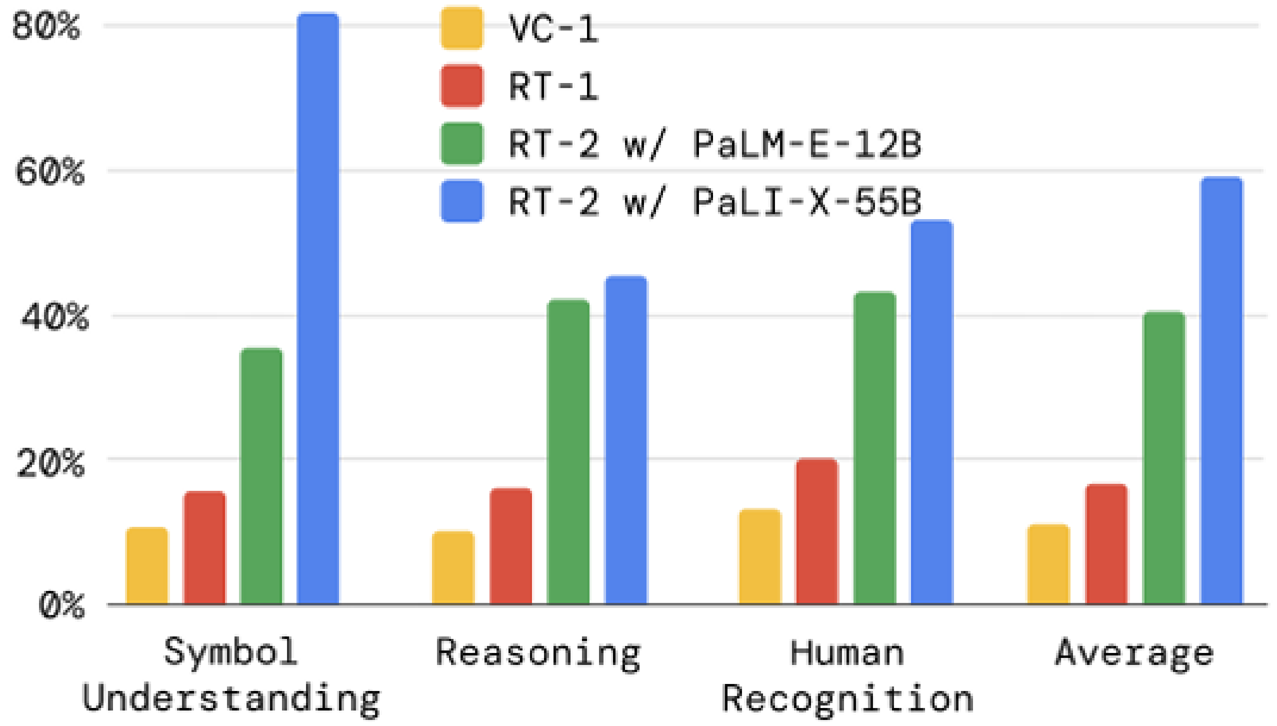


Figure 8 | An overview of some of the evaluation scenarios used to study the emergent capabilities of RT-2. They focus on three broad categories, which are (a) reasoning, (b) symbol understanding, and (c) human recognition. The visualized instructions are a subset of the full instructions, which are listed in Appendix F.2.

https://robotics-transformer2.github.io/#demo

2. Emergent capabilities-Quantitative Evaluations

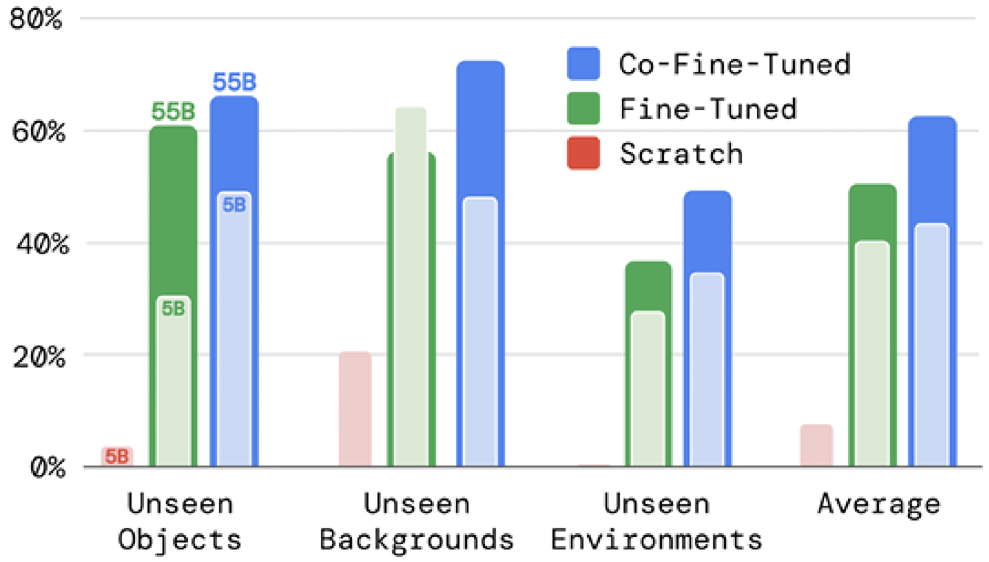
Model	Symbol Understanding				Reasoning					Person Recognition			Average
	Symbol 1	Symbol 2	Symbol 3	Average	Math	Logos	Nutrition	Color/Multilingual	Average	Celebrities	CelebA	Average	
VC-1 (Majumdar et al., 2023a)	7	25	0	11	0	8	20	13	10	20	7	13	11
RT-1 (Brohan et al., 2022)	27	20	0	16	5	0	32	28	16	20	20	20	17
RT-2-PaLI-X-55B (ours)	**93**	**60**	**93**	**82**	25	52	**48**	**58**	**46**	**53**	**53**	**53**	**60**
RT-2-PaLM-E-12B (ours)	67	20	20	36	**35**	**56**	44	35	43	33	**53**	43	40



Ablations

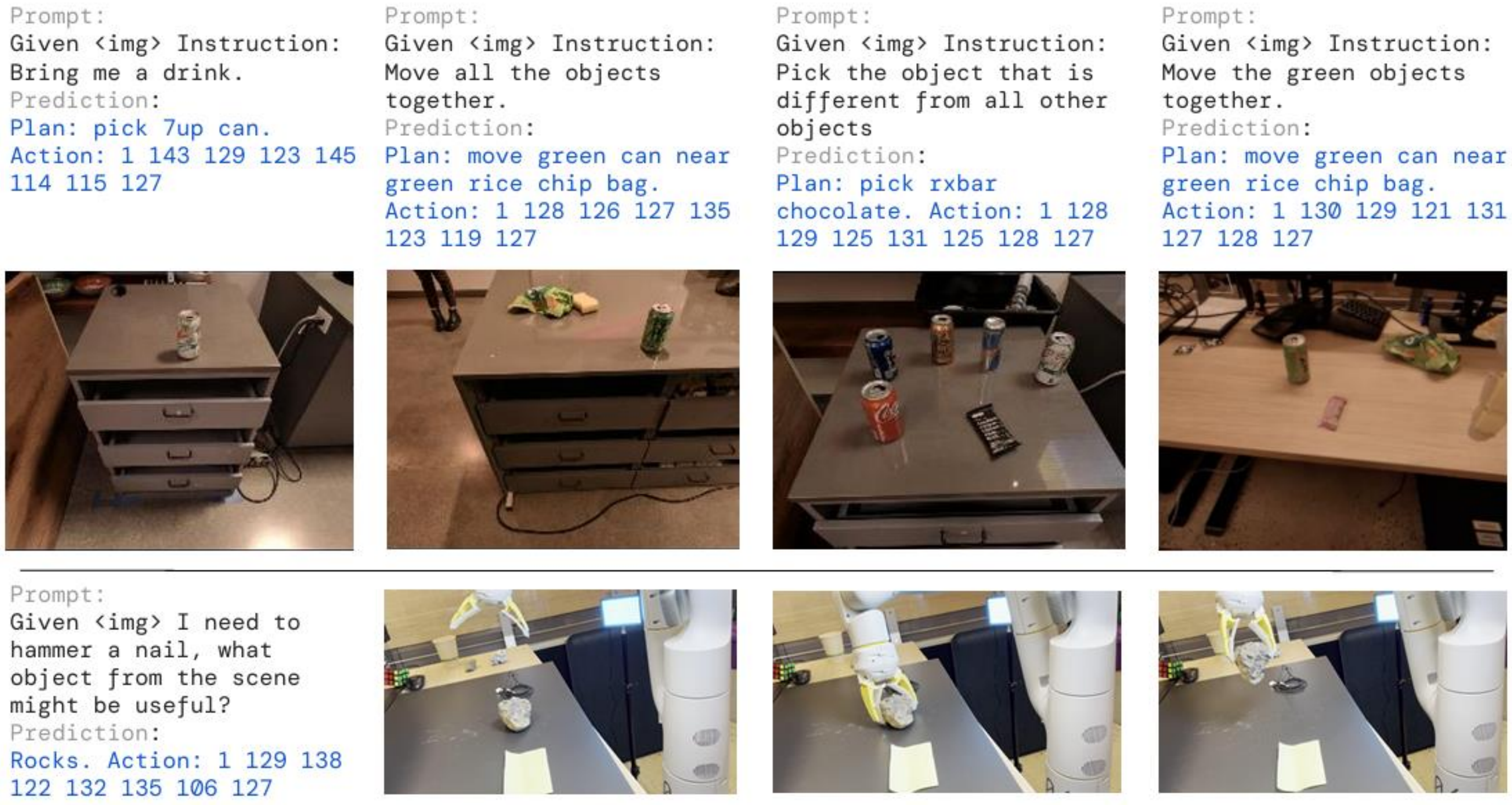
- 3. parameter count and other design decisions

Model	Size	Training	Unseen Objects		Unseen Backgrounds		Unseen Environments		Average
			Easy	Hard	Easy	Hard	Easy	Hard	
RT-2-PaLI-X	5B	from scratch	0	10	46	0	0	0	9
RT-2-PaLI-X	5B	fine-tuning	24	38	79	50	36	23	42
RT-2-PaLI-X	5B	co-fine-tuning	60	38	67	29	44	24	44
RT-2-PaLI-X	55B	fine-tuning	60	62	75	38	57	19	52
RT-2-PaLI-X	55B	co-fine-tuning	70	62	96	48	63	35	63



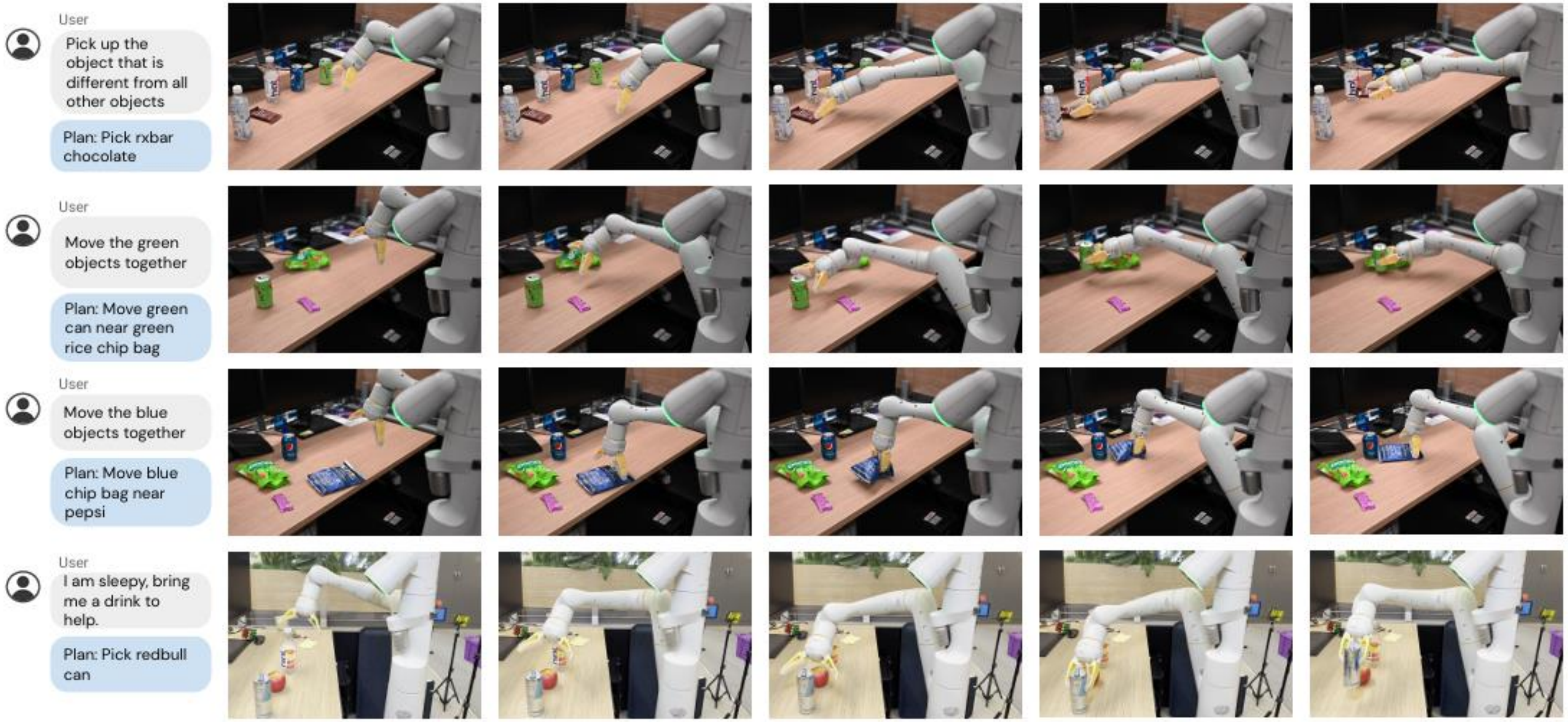
Ablations

- 4. CoT Reasoning (with RT-2-PaLM-E variant)



Ablations

- 4. CoT Reasoning



Limitations & Societal Implications

1. Doesn't generate new motions
2. High computation cost (high latency)

Limitations & Societal Implications

3. Example Failure Cases

- Not generalizing to unseen object dynamics
- Grasping objects by specific parts, such as the handle
- Novel motions beyond what was seen in the robot data, such as wiping with a towel or tool use
- Dexterous or precise motions, such as folding a towel
- Extended reasoning requiring multiple layers of indirection

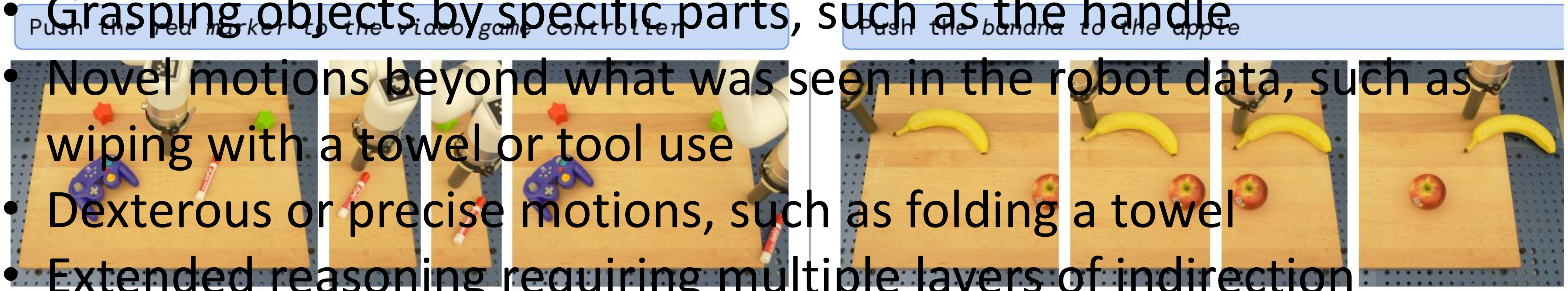


Figure 9 | Qualitative example failure cases in the real-world failing to generalize to *unseen object dynamics*.

Limitations & Societal Implications

What else?

Long horizon tasks?

Closed source

Discussion: Trade-off continuous/discrete actions
How should the VLM/VLA interact with action space?

Thank You!