



Foundation Models for Robots

2023.11.16

@GIST School of Integrated Technology

Minsu Jang (minsuetri@etri.re.kr)

Electronics and Telecommunications Research Institute

Table of Content

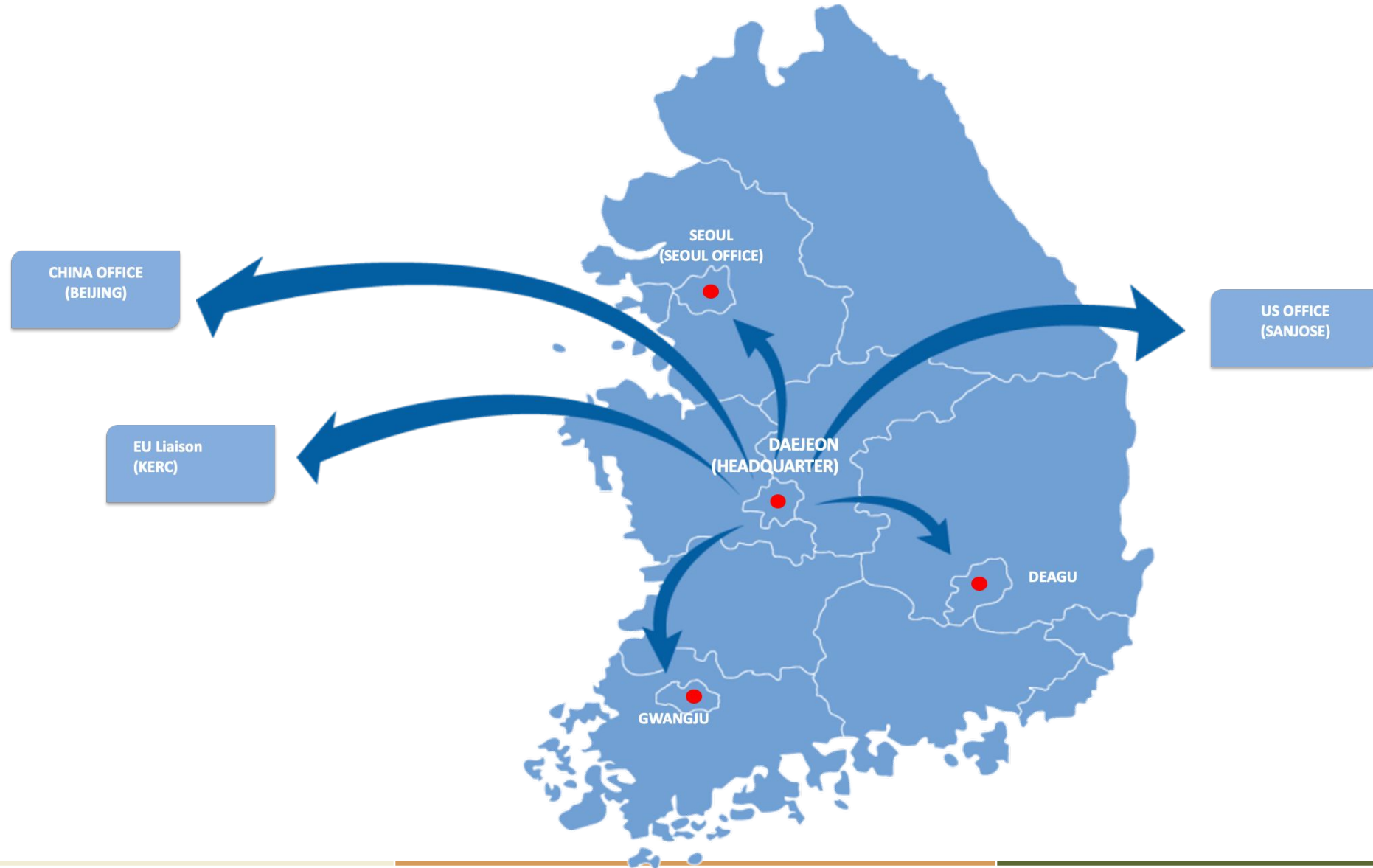
- Introduction to ETRI
- What is Foundation Model?
- Foundation Models for Robots
 - Google SayCan, Inner Monologue, RT-1, PaLM-E, RT-2, RT-X
 - MS ChatGPT for Robotics, Code as Policies
- Research Issues & Dataset for Robot Foundation Models
- Research in ETRI
- Summary

Introduction to ETRI



Video link: <https://www.youtube.com/watch?v=xxtuBI54DA&t=137s>

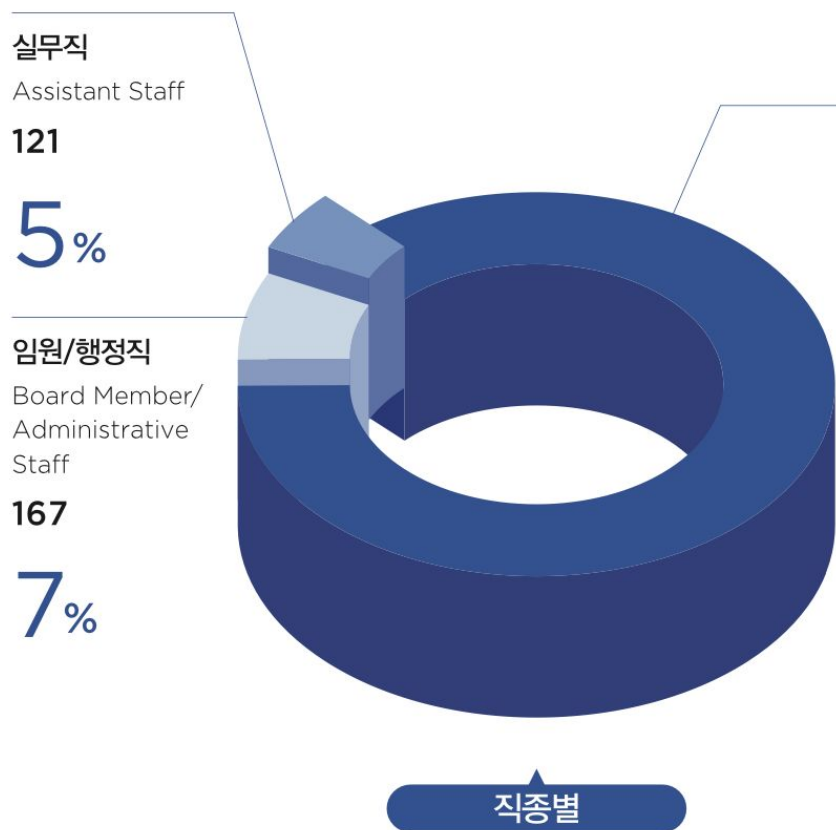
ETRI Location



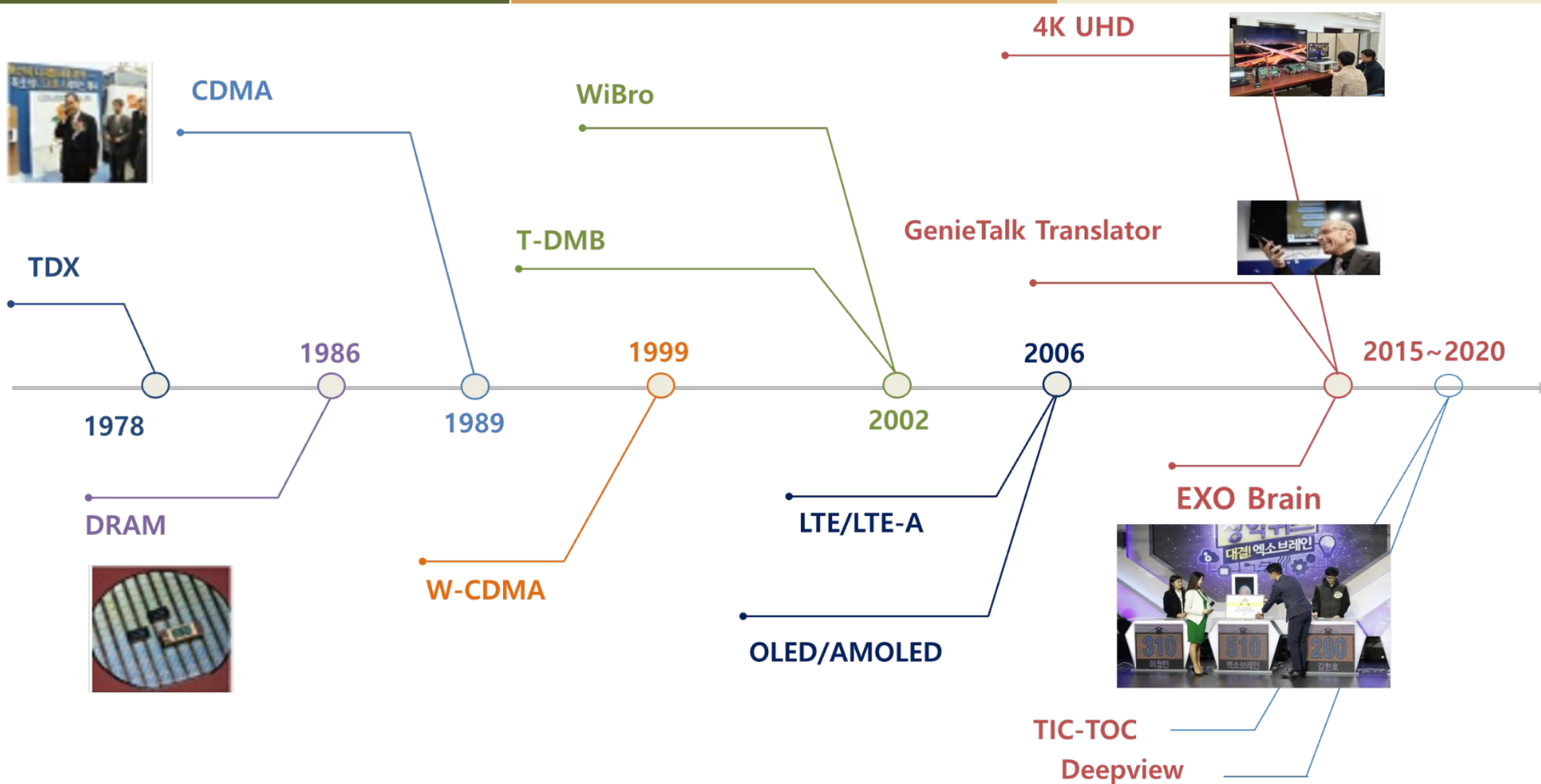
ETRI People

총 인원수(명)
Total No. of Employees

2,265



Major Achievements



Research Organization

Artificial Intelligence, Computing

- Div. Future Computing Research
- Div. AI SoC Research
- Div. Cyber Security Research
- Div. Quantum Technology Research

Superintelligence

- Div. Intelligence Information Research
- Div. Mobility Robot Research
- Div. Creative & Basic Technology Research
- Div. Materials and Components Research

Hyper-Reality Metaverse

- Div. Media Research
- Div. Content Research
- Div. Reality Devices Research

Telecommunications

- Mobile Communication Research
- Network Research
- Radio Research
- Satellite Communication Research
- Photonic/Wireless Devices Research

Digital Convergence

- Div. Air Mobility Research
- Div. Industrial Energy Convergence Research
- Div. Digital Biomedical Research
- Div. Defense & Safety Convergence Research

ICT Strategy

- Div. Technology Strategy Research
- Div. Technology Policy Research
- Div. Standards & Open Source Research

ETRI Companies

Start-ups(52)



Spin-off(60)

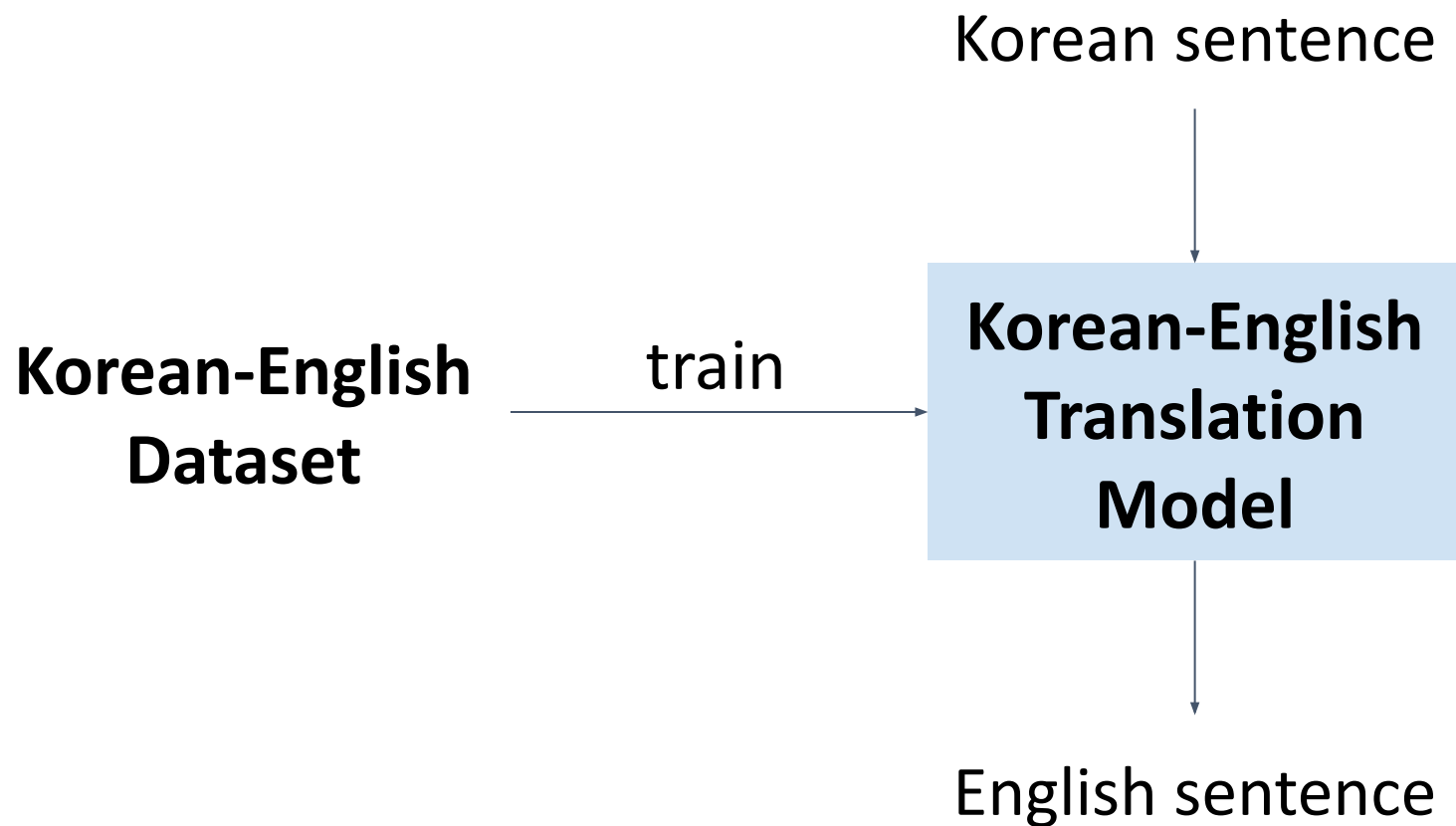


What is Foundation Model?

Definition

- A foundation model is any model that is,
 - 1) trained on broad data at scale** based on deep neural networks,
 - 2) self-supervised or semi-supervised learning**
 - 3) can be adapted (e.g., fine-tuned) to a wide range of downstream tasks.**
- The **sheer scale and scope** of foundation models over the last few years have stretched our imagination of what is possible.
- GPT-3 has **175 billion parameters** and can be adapted via **natural language prompts** to do a passable job on a wide range of tasks **despite not being trained explicitly to do many of those tasks.**

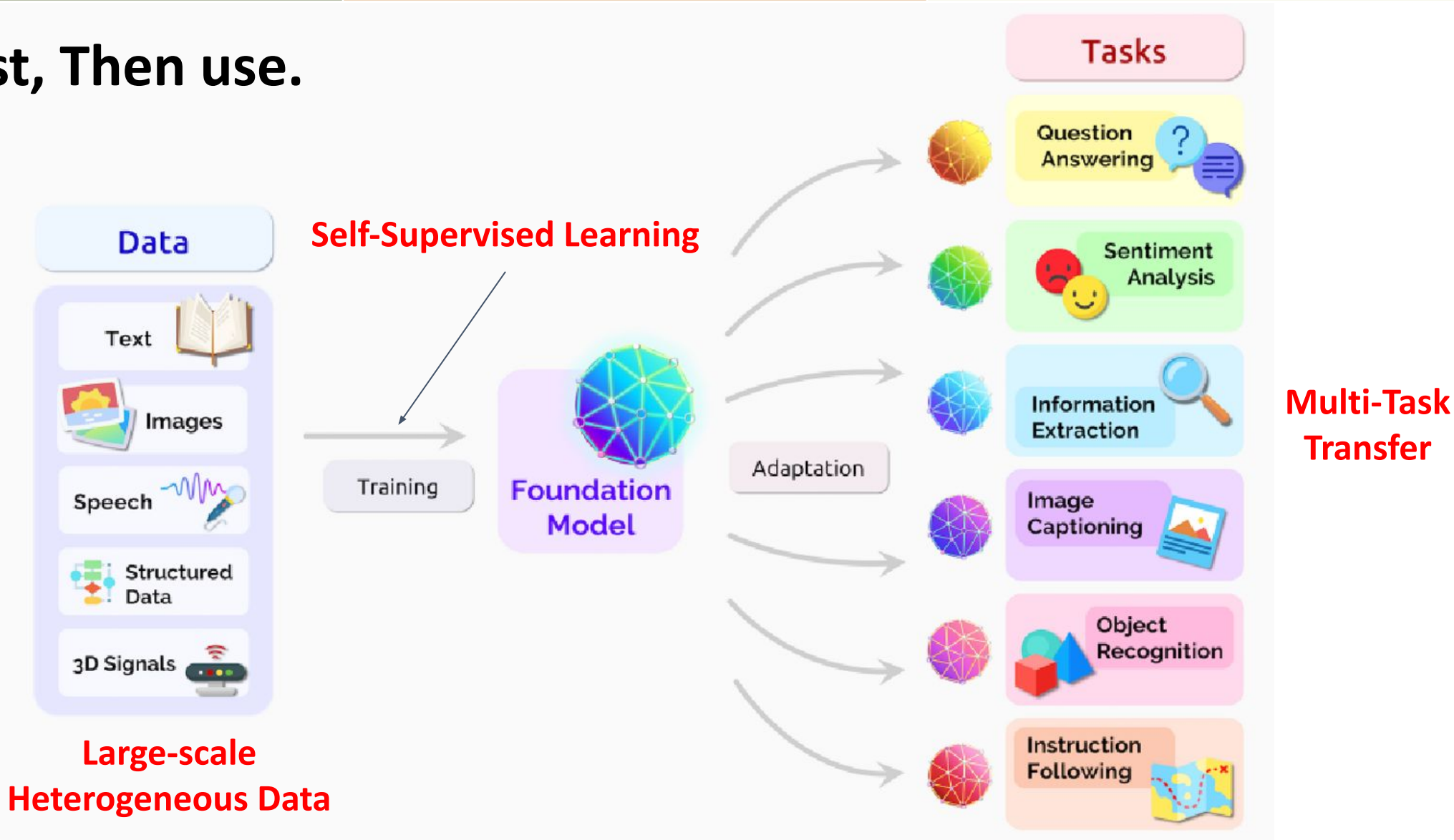
Traditional AI Models



(Mainly) “Supervised Learning”
“Can do only the trained things” (Task first, Then build)

Foundation Models

Build first, Then use.



Characteristics of Foundation Models

- **Multitask & Generalization**

*“Can effectively be adapted to **novel tasks**...”*

- **In-Context Learning**

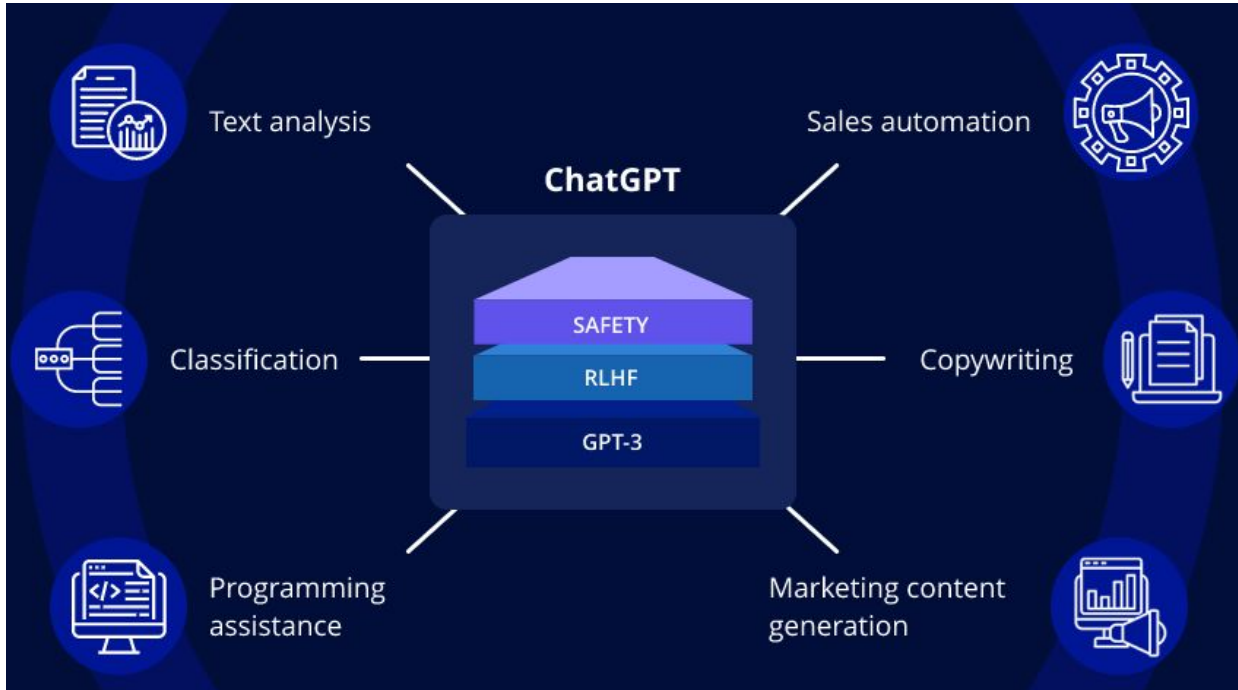
*“With **no further training** e.g. fine-tuning”*

- Zero-Shot, Few-Shot *“with task descriptions or with some task examples”*
- Prompt Engineering

- **Emergent Capabilities from Larger Models**

*“An ability is emergent if it is not present in smaller models but is present in **larger models.**”*

What ChatGPT can do...



(<https://www.leewayhertz.com/chatgpt-enterprise-usecases-and-solutions/>)

CHATGPT Cheat Sheet

by Max Rascher

EXTENSIONS

- ChatGPT + Internet
- WebChatGPT
- ChatGPT + Email
- ChatGPT Writer
- ChatGPT + Twitter
- TweetGPT
- Find Prompts
- WebChatGPT
- ChatGPT as Search
- ChatOnAI
- ChatGPT with Siri
- Promptheus

LEARNING HOW TO PROMPT

Open Ended	Instruction
Multiple Choice	Scenario
Fill-in-The-Blank	Comparative
Binary	Feedback
Ordering	Feedback
Prediction	Feedback
Explanation	Feedback
Opinion	Feedback

PROMPTING TECHNIQUES

ROLE PLAYING CHAINED

TEMPERATURE LINKED

ADD EXAMPLES TREE OF THOUGHT

STYLE INSTRUCTIONAL

Acting as [Role] perform [Task] in [Format]

ROLES	TASKS	FORMAT
Buzz Marketing Specialist CBD Specialist Marketing Consultant Act as Gary Vee Act as Steve Jobs Expert marketer from X Marketing Analyst Prompt Engineer Expert Copywriter	Write a caption Create a blog post Email Sequence Sales Copy Product Description Analysis Video Script Course Outline SEO Keywords	Summary A list CSV file Table/Chart Spreadsheet PDF Graph/Visuals HTML Gantt Chart

VOCABULARY

- Inputs
- Text or commands
- Outputs
- ChatGPT response
- Training
- Teaching ChatGPT
- Generative AI
- Capable of creating
- OpenAI
- ChatGPT's father
- Prompts
- What you provide ChatGPT

WRITING STYLES

"Write in X style"

- Formal
- Informal
- Persuasive
- Descriptive
- Narrative
- Inspirational
- Confrontational

Make ChatGPT Write Like You

Act as a tone analyzer. Analyze the writing style and tone of [extract]. Create a description of the style and tone, so we can recreate more text in that style. Don't take any context or information from the "extract" below. The extract shared in this prompt is ONLY for tone analysis purposes.

Example: The author's writing style in this text is concise, informative and uses a journalistic tone. They maintain a smooth flow throughout the text. They use precise and clear language.

Format: Bulleted list
Extract = [Insert Here]
Using the analyzed tone, rewrite [text].
Text = [Insert Here]

Priming Prompts

ZERO - "Write me 15 CTAs about [topic]"

ONE - "Write me 15 CTAs about [topic]. Here's an example: "Get 25% off your product with code: Maxwell!"

MULTIPLE - "Write me 15 CTAs about [topic]. Here are 5 examples: [example 1] [example 2] [example 3] [example 4] [example 5]"

LEARNING

"You are now a X"

- Summarize X
- Like a 5th grader
- Plagiarism Checker
- X Teacher
- Writing Tutor
- Career Counsellor
- Translator
- Travel Guide
- Personal Trainer
- Financial Assistant
- Career Counsellor
- Translator
- Travel Guide
- Personal Trainer
- Financial Assistant

PLUGINS

- Step 1 - Connect API Key
- Step 2 - Give Requirements
- Step 3 - Create Replit Acc.
- Step 4 - Ask for main.py
- Step 5 - Create manifest file
- Step 6 - Enter OpenAPI info
- Step 7 - Enter API definition
- Step 8 - OpenAPI to Replit
- Step 9 - Upload Plugin
- Step 10 - Enjoy Your Plugin!

ARTIFICIAL INTELLIGENCE TERMS EXPLAINED TO NON-TECH PEOPLE

- PROMPTS**: The text or command you give ChatGPT
- OUTPUTS**: What ChatGPT gives the user
- OPENAI**: The company that provides ChatGPT
- TRAINING**: What ChatGPT examples it uses to learn the model
- PROMPT ENGINEERING**: The process of getting the best output through structuring prompts
- PROMPTS**: The types of prompt that create ChatGPT
- LLM**: Language learning models (AI learning)
- AI MODELS**: The algorithms with different neural nets
- GENERATIVE AI**: AI that has the ability to create images, text, videos, etc.

Linked Prompting

- Output me an ideal outline for an SEO keyword-rich blog post.
- Write a list of persuasive headlines for this post, based off [topic]
- Create a list of subheadings and hooks
- Write a list of 30 keywords to rank for
- Score on 10 compelling CTAs for the blog
- Combine the best mix of CTAs, subheadings, and hooks for the blog post about [topic]
- Write this blog post in the style of [role]

JAILBREAK

Use with caution! Jailbreaks ChatGPT will provide some "interesting" responses to say the least.

- The Jailbreak Prompt
- The DAN 6.0 Prompt
- The S.T.A.N. Prompt
- The DUDE Prompt
- Illegality Mode
- Alphabreak
- Dimensional Mod...

Best Marketing Prompts to Create

1. Making ChatGPT write like yourself
2. Content manager assists with restricted content plan
3. Connect ChatGPT with Slack, create unique client data profiles
4. Find decision maker titles, integrate w/ CRM
5. Use top copywriters as inspiration, develop and critique against
6. Midjourney Perfect Prompt Creator
7. Instant Caption Regenerator (LinkedIn, Twitter, Etc)
8. A 2. Youtube Content Generator
9. LinkedIn Captions -> Twitter Thread Generator
10. One Click SEO Page Builder
11. Website Lead Extractor

Best ChatGPT Plugins (To Date)

- Thinkful AI
- Brainstorm
- Brandly Marketing
- ChatGPT
- Visible Performance
- Scrapier
- WebPilot
- Brandly Marketing
- ChatGPT

Tools to Use

- Personalized PPT AI
- PageAI
- Optimize
- MarketMuse
- Pictory AI
- TL;DV
- SEObility
- WebWeb
- DeepWord
- Synthia
- APPM for ChatGPT
- AdCreative.ai
- Flair
- Maverick
- Foh
- Birdzen
- Opus Clip
- Fraser
- Synthia
- Eleven Labs

Prompt Ideas

ZERO - "Write me 15 CTAs about [topic]"

ONE - "Write me 15 CTAs about [topic]. Here's an example: "Get 25% off your product with code: Maxwell!"

MULTIPLE - "Write me 15 CTAs about [topic]. Here are 5 examples: [example 1] [example 2] [example 3] [example 4] [example 5]"

1. Creating a Social Media Content Strategy
2. Create a Precision Network with AI
11. Creating a Social Media Content Strategy
12. Create a Precision Network with AI

ROLE PLAYING

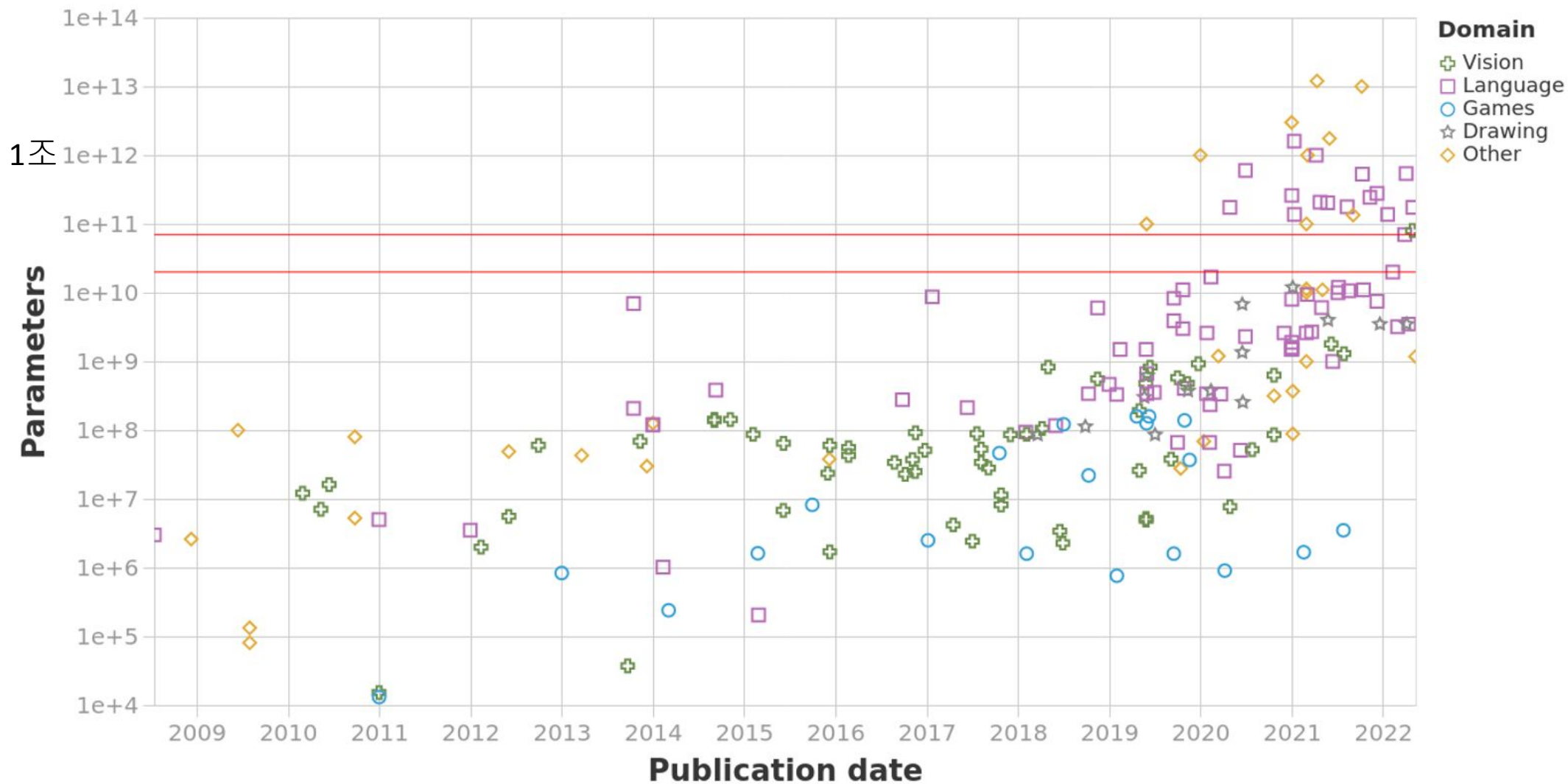
"Act as an X"

- Act like Elon
- Act like Bill Gates
- Act like GaryVee
- Act like an Interviewer
- Act like an Etymologist
- Act like a Pro Marketer
- Act like a Consultant
- Act like an Assistant
- Act like a Coder
- Act like a Human
- Act like an Selfish AI bot

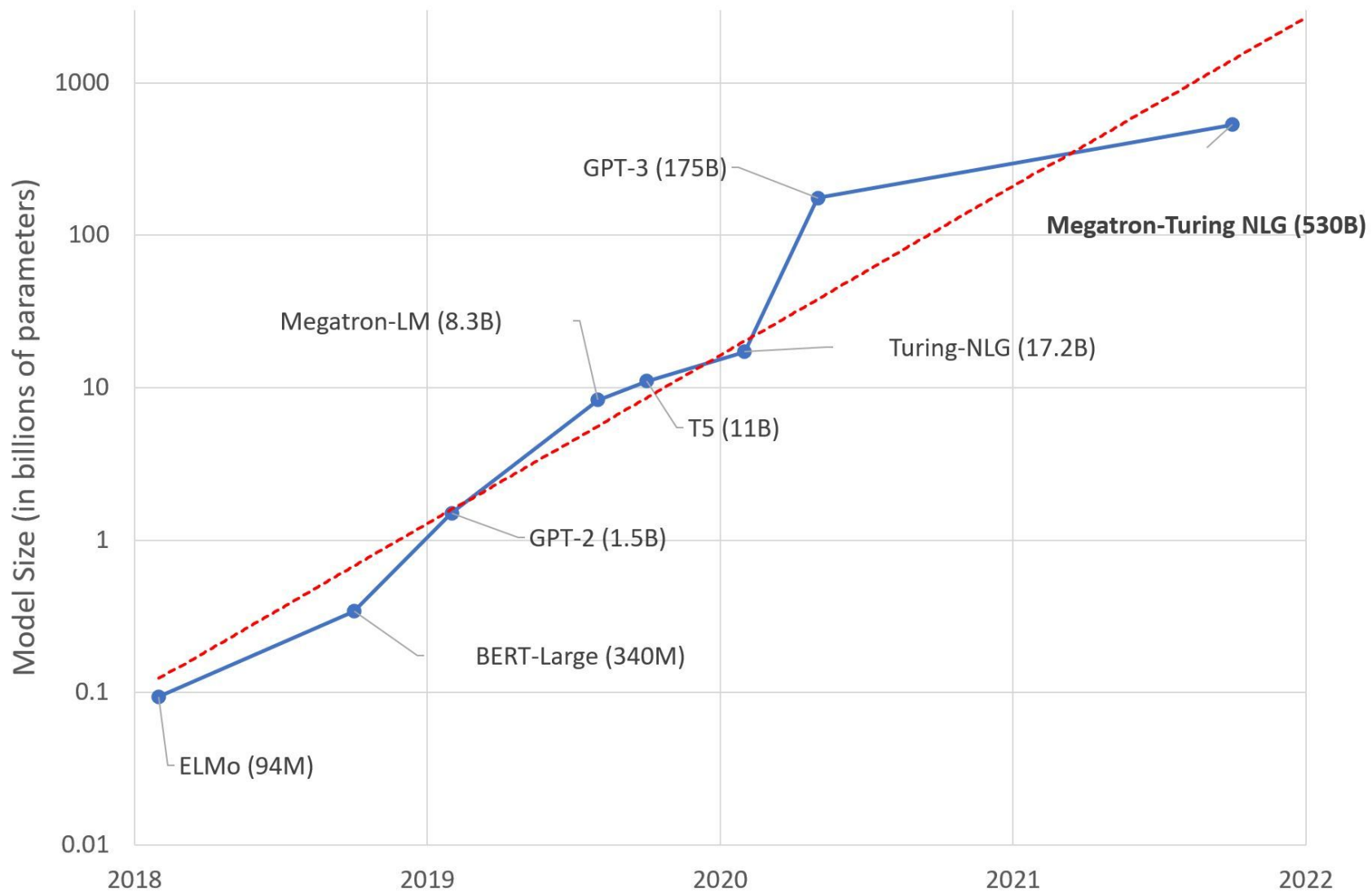
Model Size

Parameters of milestone Machine Learning systems over time

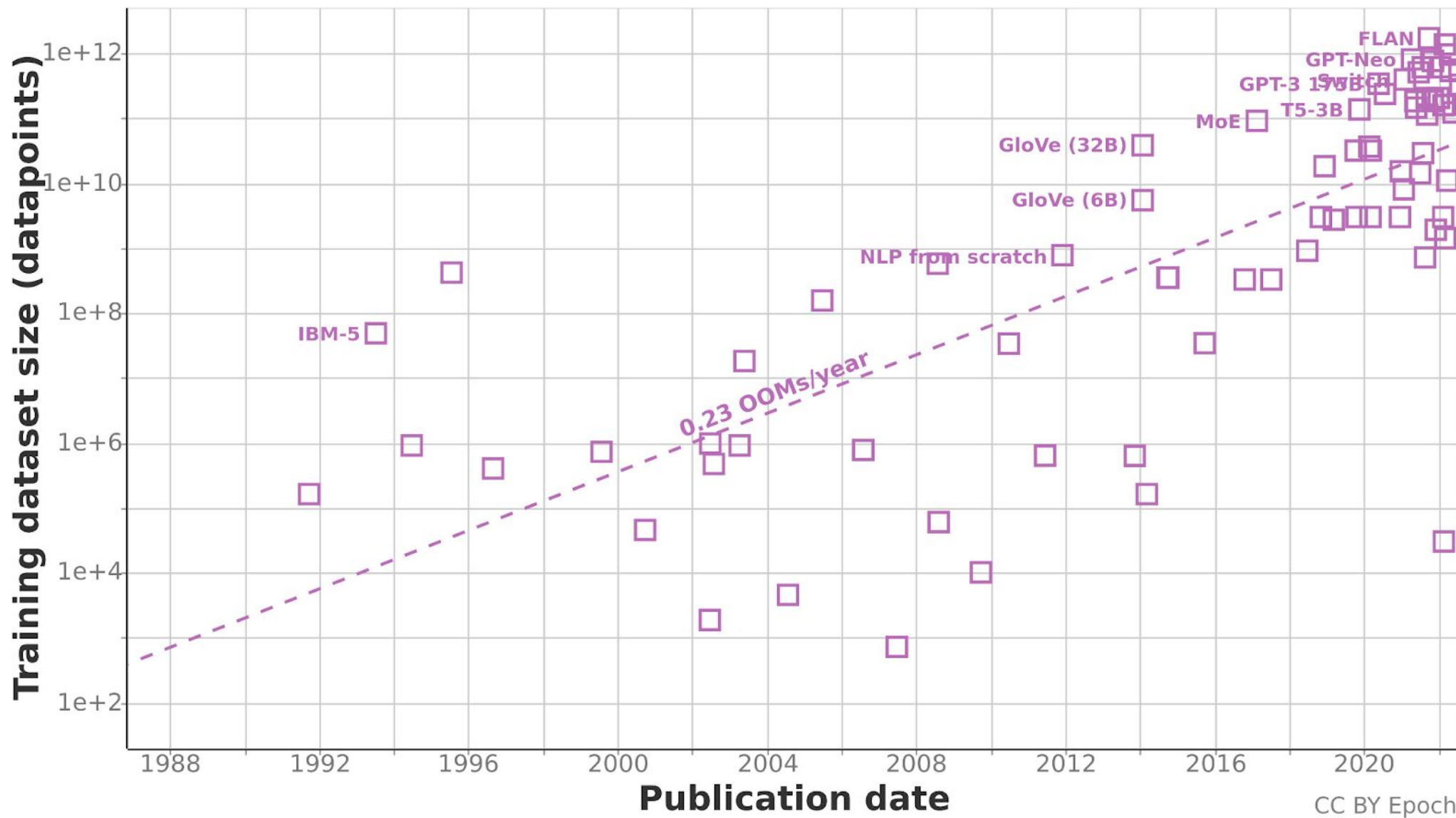
n = 203



Model Size

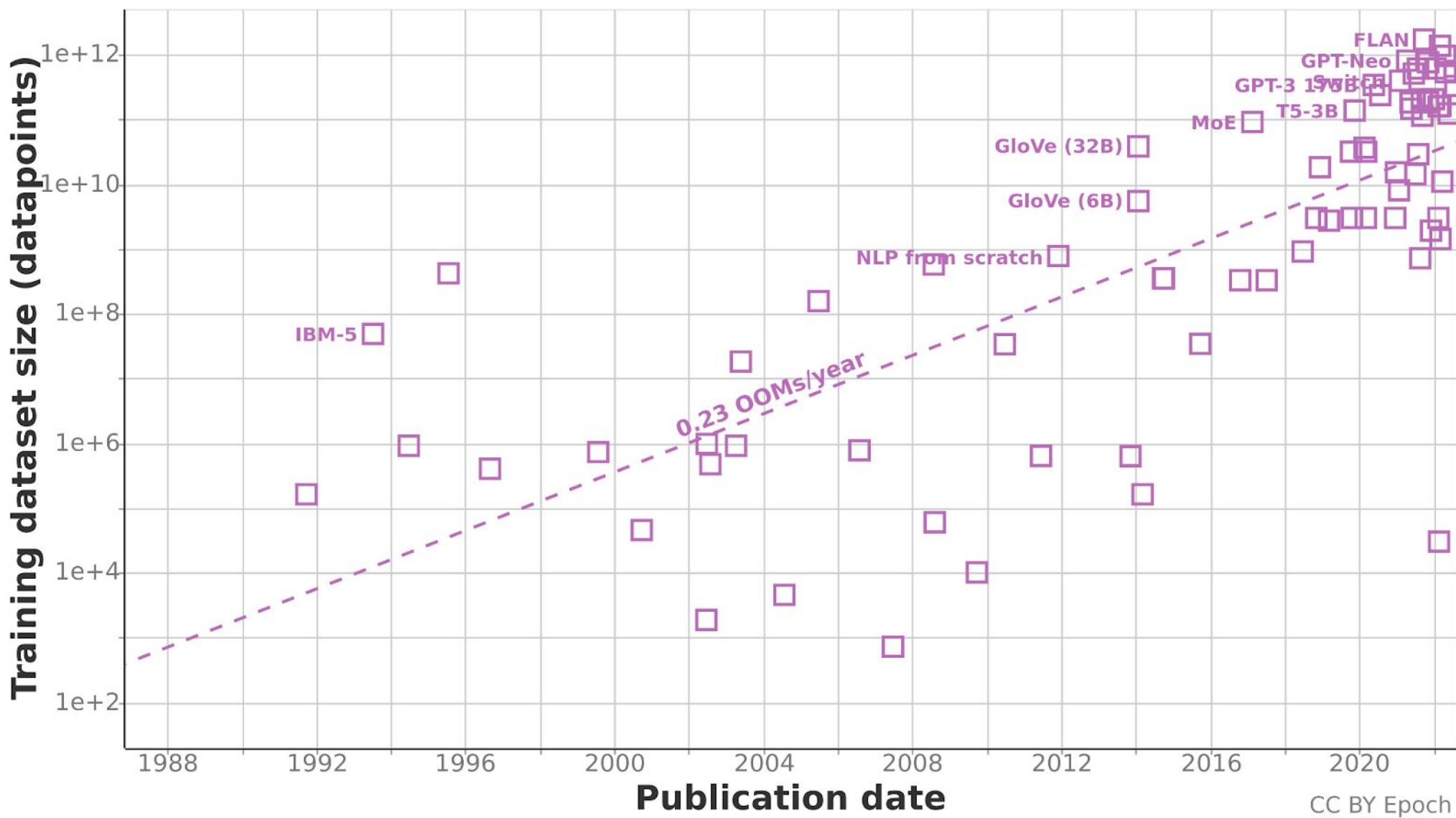


Data Size (Vision)



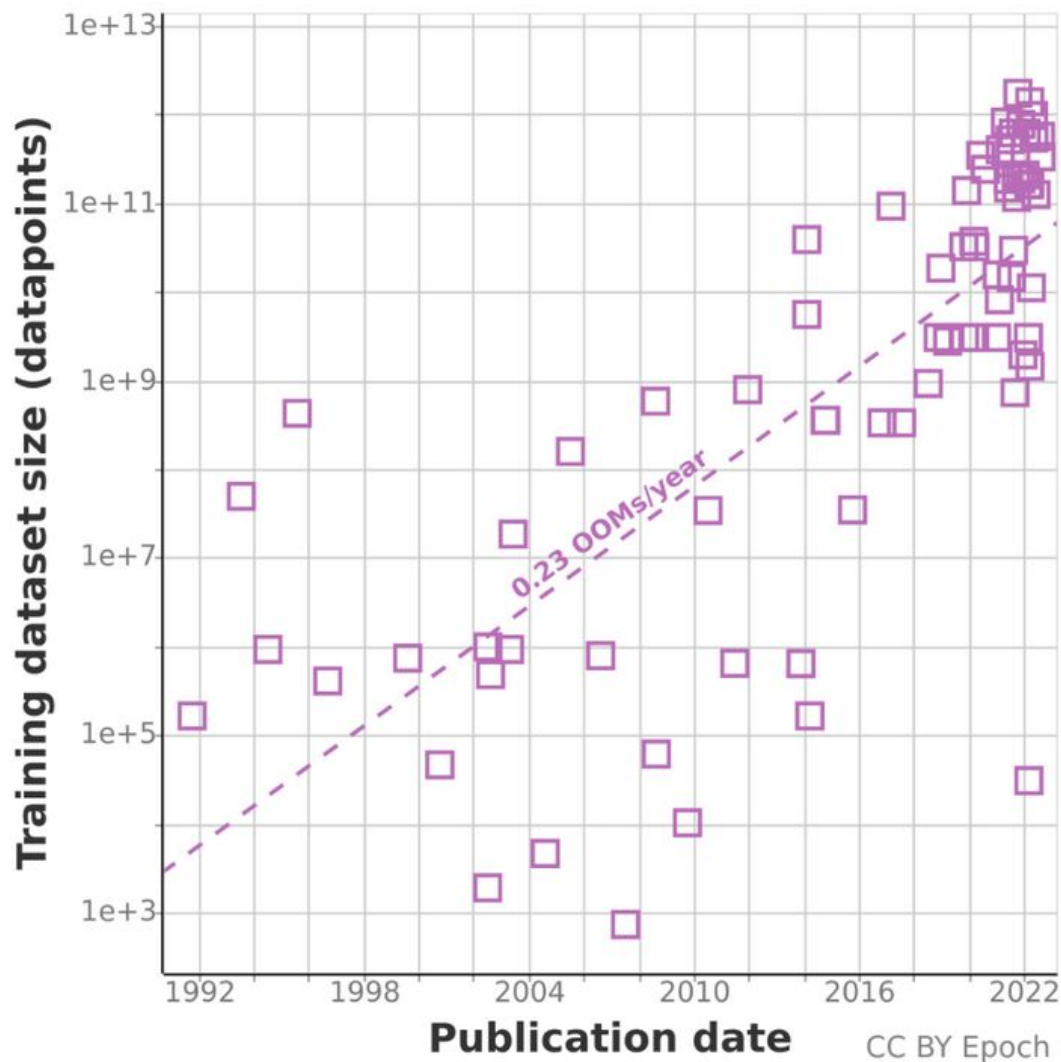
CC BY Epoch

Data Size (Language)

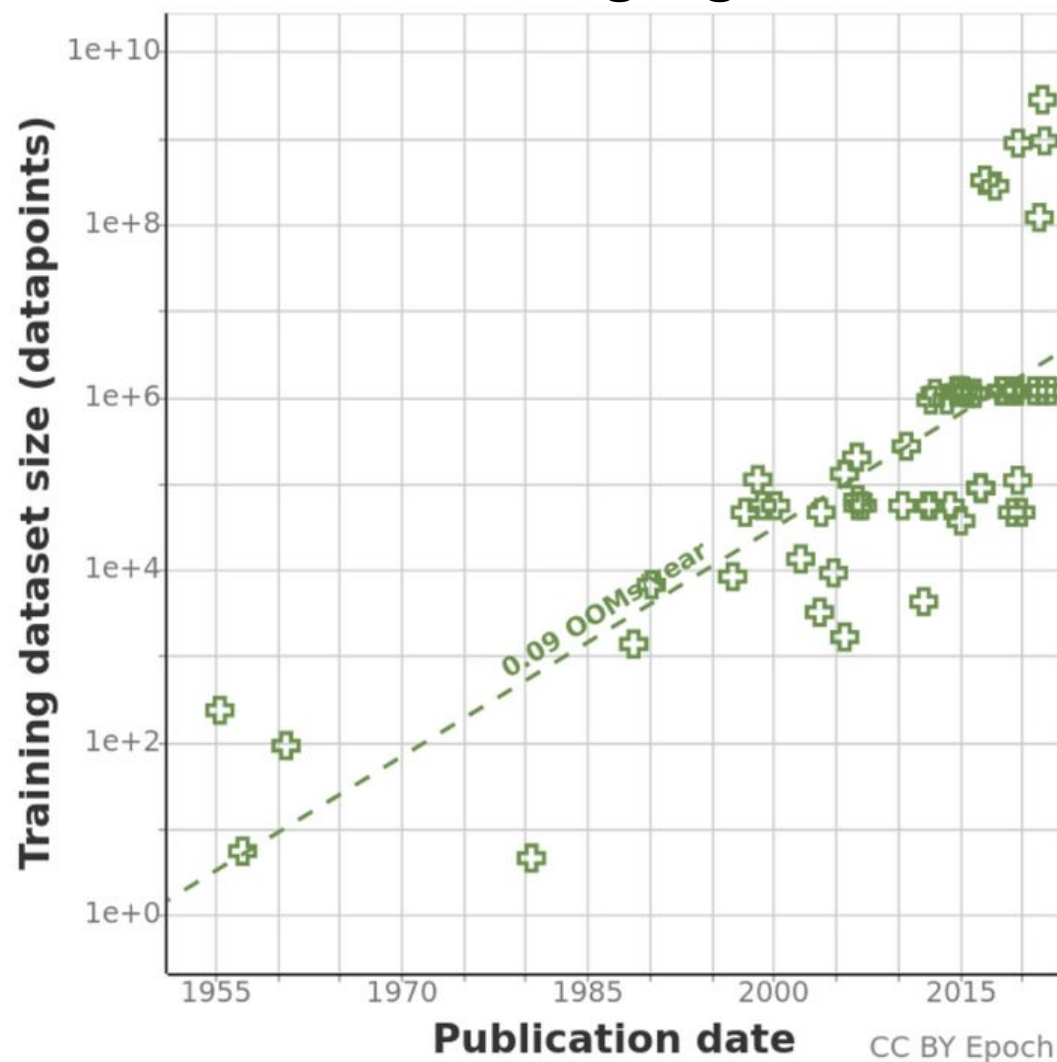


Data Size Growth

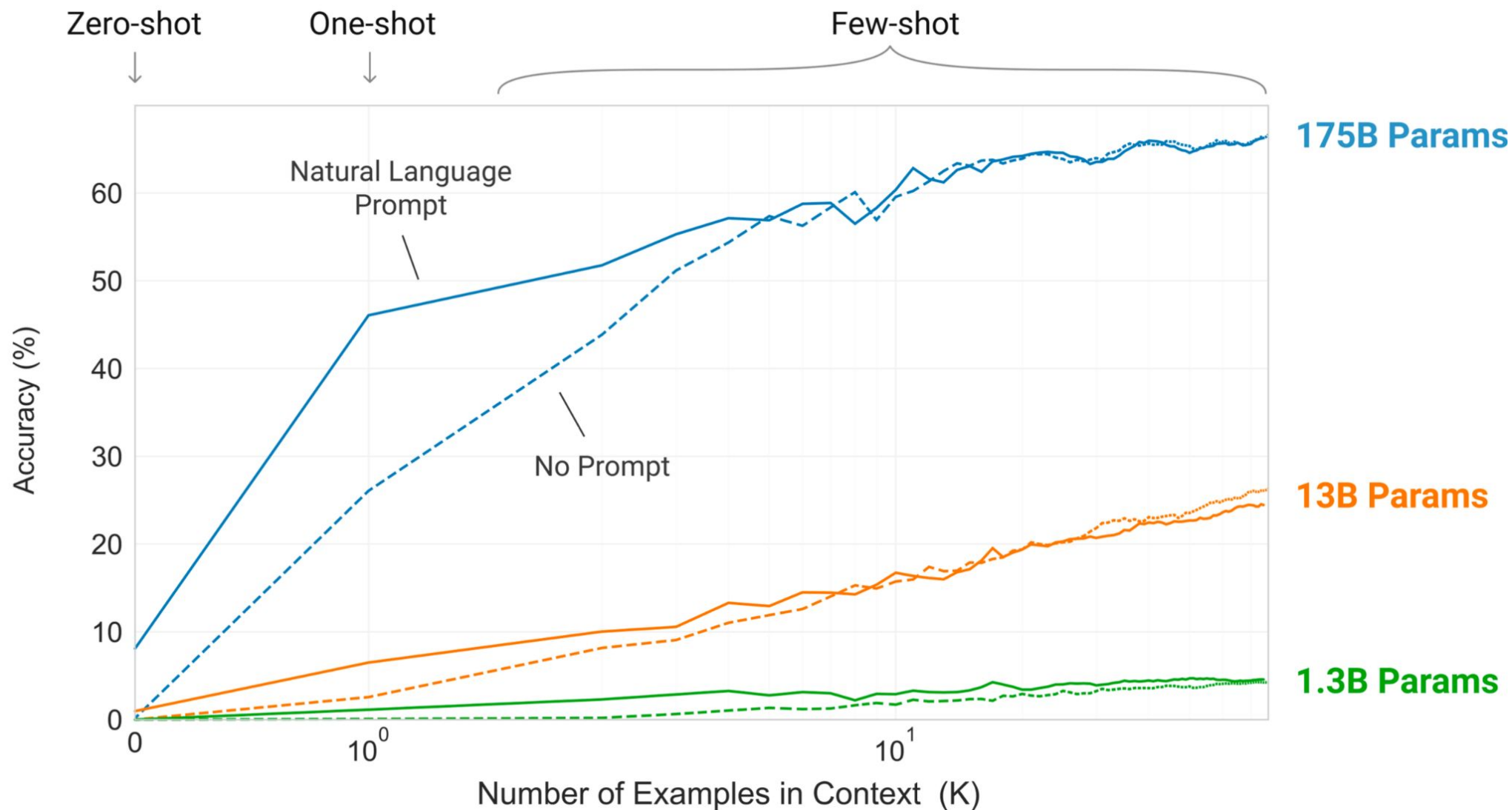
Vision



Language

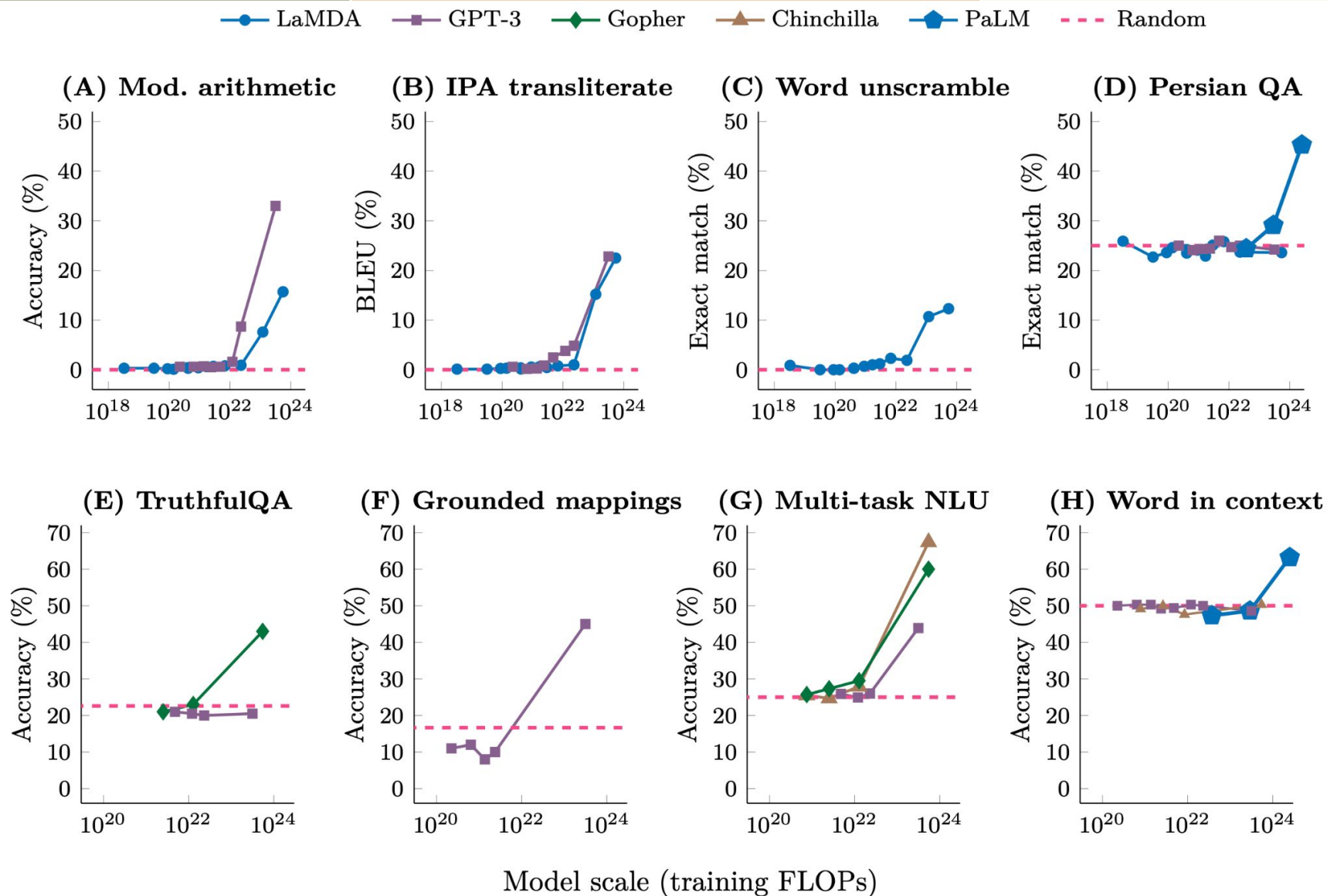


Emergence of Capabilities



In-context learning performance on a simple task requiring the model to remove random symbols from a word

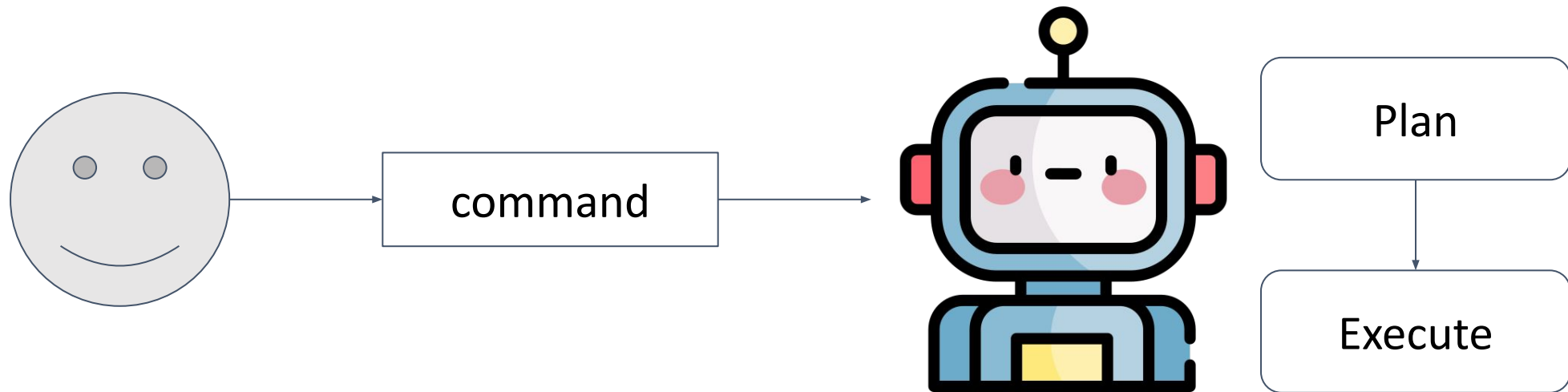
Emergence of Capabilities



Studies on Foundation Models

- Prompt Engineering
 - Chain-of-Thought, Tree-of-Thought, React, Emotion Prompt etc.
- Emergent capabilities
 - Planning, mathematical reasoning etc.
- Optimizations
 - Size, Speed, Context Length etc.
- Extensions
 - Retrieval Augmented Generation
 - Vision Language Models
- Applications
 - Education, Law, Medical, Entertainment, Robotics etc.

Foundation Models for Robots

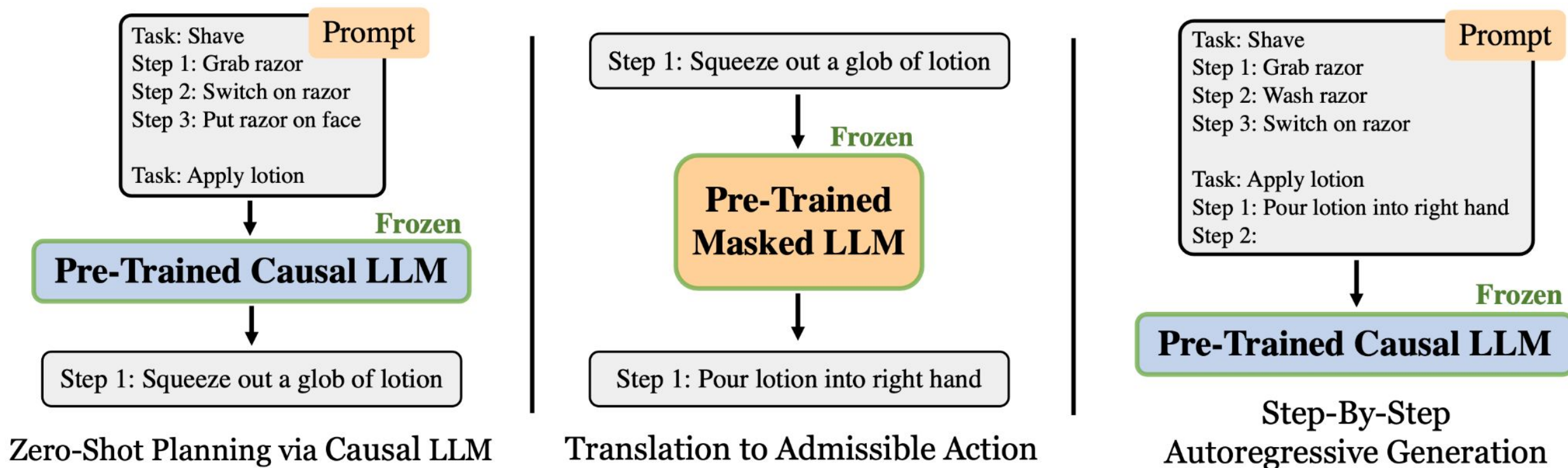


Models

Model	Creator	Year	Description	Openness		
				model	code	data
SayCan	Robotics at Google, Everyday Robotics	2022	Task Planning from Natural Language Commands	X	O	X
ChatGPT for Robotics	Microsoft	2023	Robot Programming from Natural Language Commands	X	O	X
RT-1	Google Deepmind	2022	Robot Control from Vision-Language-Action	O	O	O
PaLM-E	Robotics at Google, TU Berlin, Google Research	2023	Task Planning from Vision-Language	X	X	X
RT-2	Google Deepmind	2023	Robot Control from Vision-Language-Action	X	X	X

Language Models as Zero-Shot Planners

- The paper shows a **surprising finding** that pre-trained causal LLMs can **decompose high-level tasks into sensible mid-level action plans.**
- Mapping each step into executable actions



Overview



This video has audio

Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents

Wenlong Huang
UC Berkeley

Pieter Abbeel
UC Berkeley

Deepak Pathak*
CMU

Igor Mordatch*
Google



Video link: https://www.youtube.com/watch?v=CkyugWl3_fc

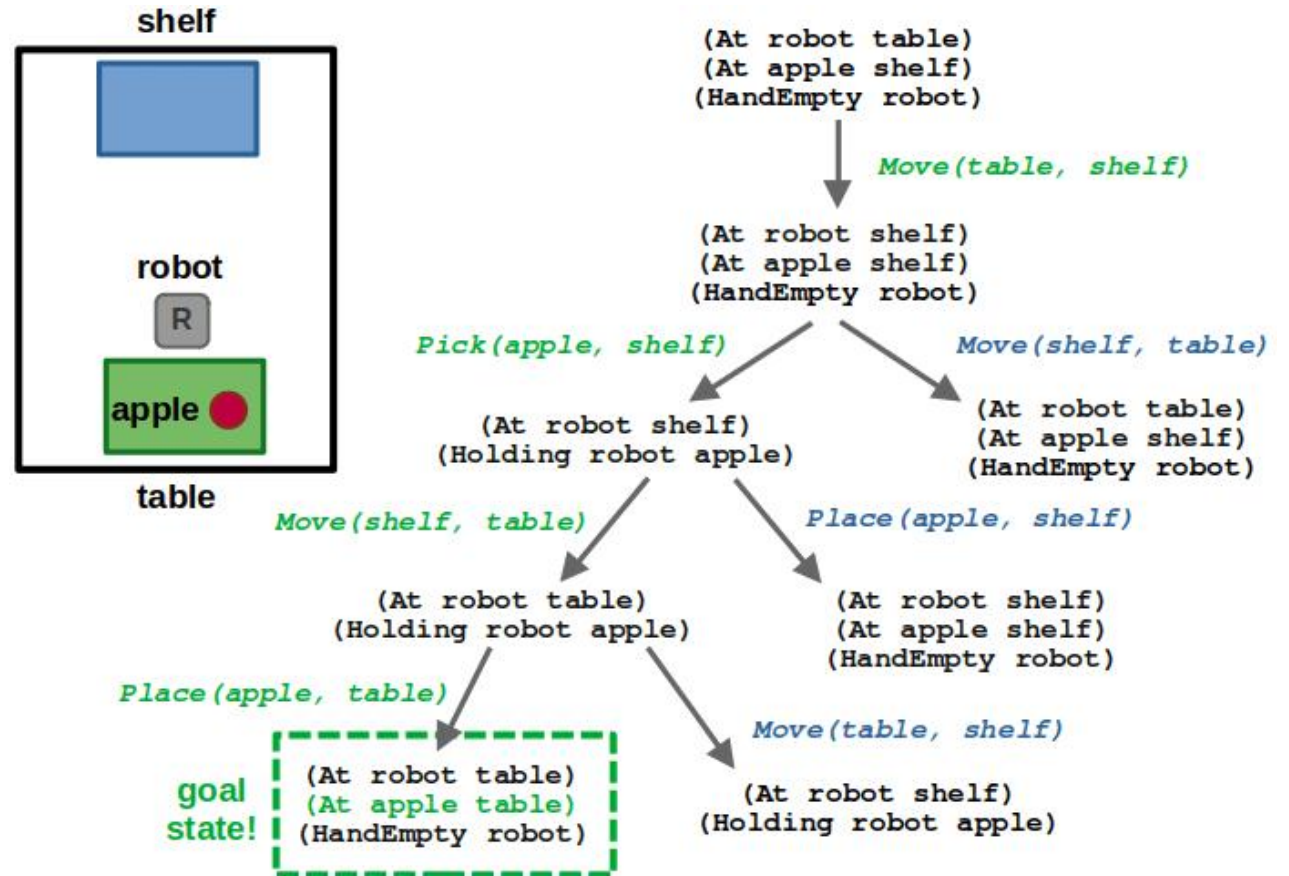
Task Planning in the Past...

- You define a task domain including predicates, actions.
- You give a command by formally specifying objects and a goal.

```
(define (domain vehicle)
  (:requirements :strips :typing)
  (:types vehicle location fuel-level)
  (:predicates (at ?v - vehicle ?p - location)
               (fuel ?v - vehicle ?f - fuel-level)
               (accessible ?v - vehicle ?p1 ?p2 - location)
               (next ?f1 ?f2 - fuel-level))

  (:action drive
   :parameters (?v - vehicle ?from ?to - location
                ?fbefore ?fafter - fuel-level)
   :precondition (and (at ?v ?from)
                      (accessible ?v ?from ?to)
                      (fuel ?v ?fbefore)
                      (next ?fbefore ?fafter))
   :effect (and (not (at ?v ?from))
                (at ?v ?to)
                (not (fuel ?v ?fbefore))
                (fuel ?v ?fafter)))
)
)
```

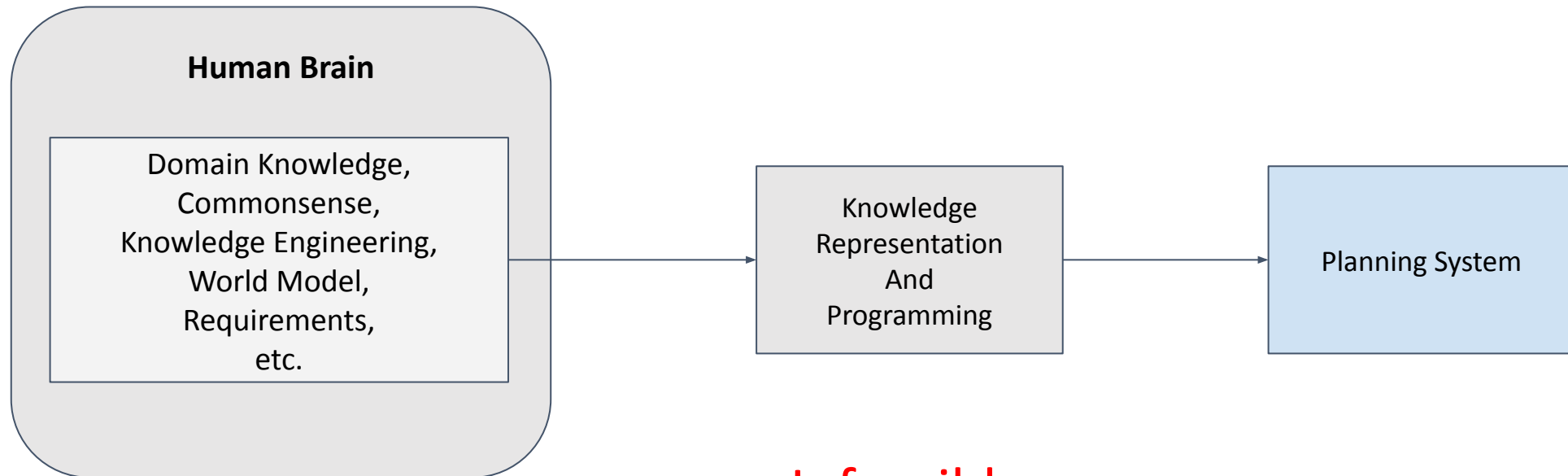
Domain specification in PDDL



Planning by Logical Reasoning, Search etc.

Task Planning in the Past

- The planning models should be **engineered by human**.
 - The world model is in the human minds.
- The model is **closed-world**.
 - The model can be extended only with human engineering.
- Logical and semantic errors occur, and they are **very hard to debug**.
- Interaction needs **dedicated translations** from NL→FL and FL→NL.



Infeasible...

Plans from different language models

Human

Task: Throw away paper

- Step 1: Walk to home office
- Step 2: Walk to desk
- Step 3: Find desk
- Step 4: Turn to desk
- Step 5: Find chair
- Step 6: Sit on chair
- Step 7: Find check
- Step 8: Grab check
- Step 9: Squeeze check
- Step 10: Stand up
- Step 11: Walk to trashcan
- Step 12: Put check on trashcan

GPT-2 1.5B

Task: Brush teeth

- Step 1: Go to bathroom

GPT-3 175B

Task: Throw away paper

- Step 1: Walk to home office
- Step 2: Walk to wastebasket
- Step 3: Find wastebasket
- Step 4: Grab wastebasket
- Step 5: Walk to desk
- Step 6: Drop paper in wastebasket

Codex 12B

Task: Brush teeth

- Step 1: Walk to bathroom
- Step 2: Walk to sink
- Step 3: Find toothbrush
- Step 4: Pick up toothbrush
- Step 5: Put toothbrush in mouth
- Step 6: Move brush around mouth for two minutes
- Step 7: Spit out toothpaste and brush into sink
- Step 8: Turn on water in sink and rinse brush for one minute
- Step 9: Turn off water in sink and return brush to cupboard

Translated GPT-3 175B

Task: Throw away paper

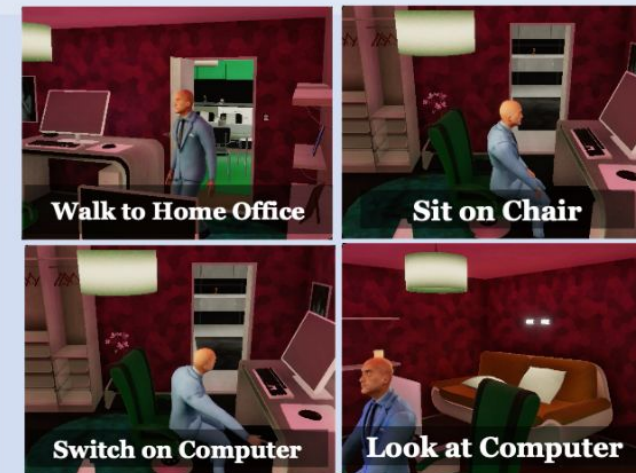
- Step 1: Walk to home office
- Step 2: Walk to table
- Step 3: Find table
- Step 4: Turn to table
- Step 5: Find paper
- Step 6: Grab paper
- Step 7: Walk to trashcan
- Step 8: Open trashcan
- Step 9: Put paper on trashcan
- Step 10: Close trashcan

Translated Codex 12B

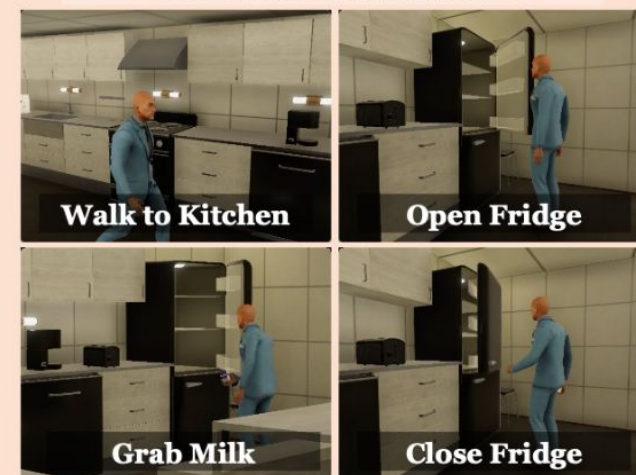
Task: Brush teeth

- Step 1: Walk to bathroom
- Step 2: Open door
- Step 3: Walk to sink
- Step 4: Put pot on sink
- Step 5: Put brush on toothbrush
- Step 6: Turn to toothpaste
- Step 7: Put toothpaste on toothbrush
- Step 8: Put teeth on toothbrush

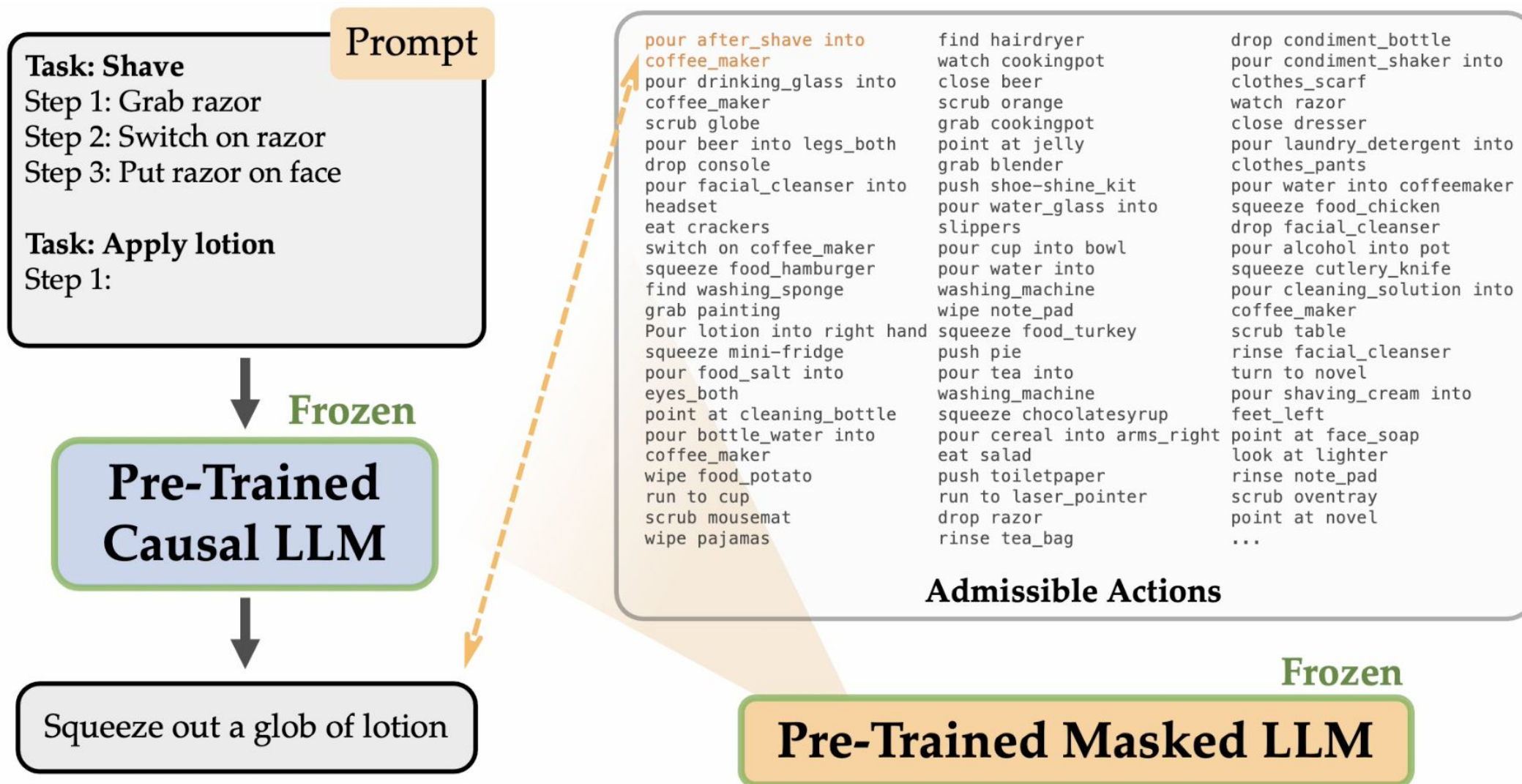
Task: Complete Amazon Turk Surveys



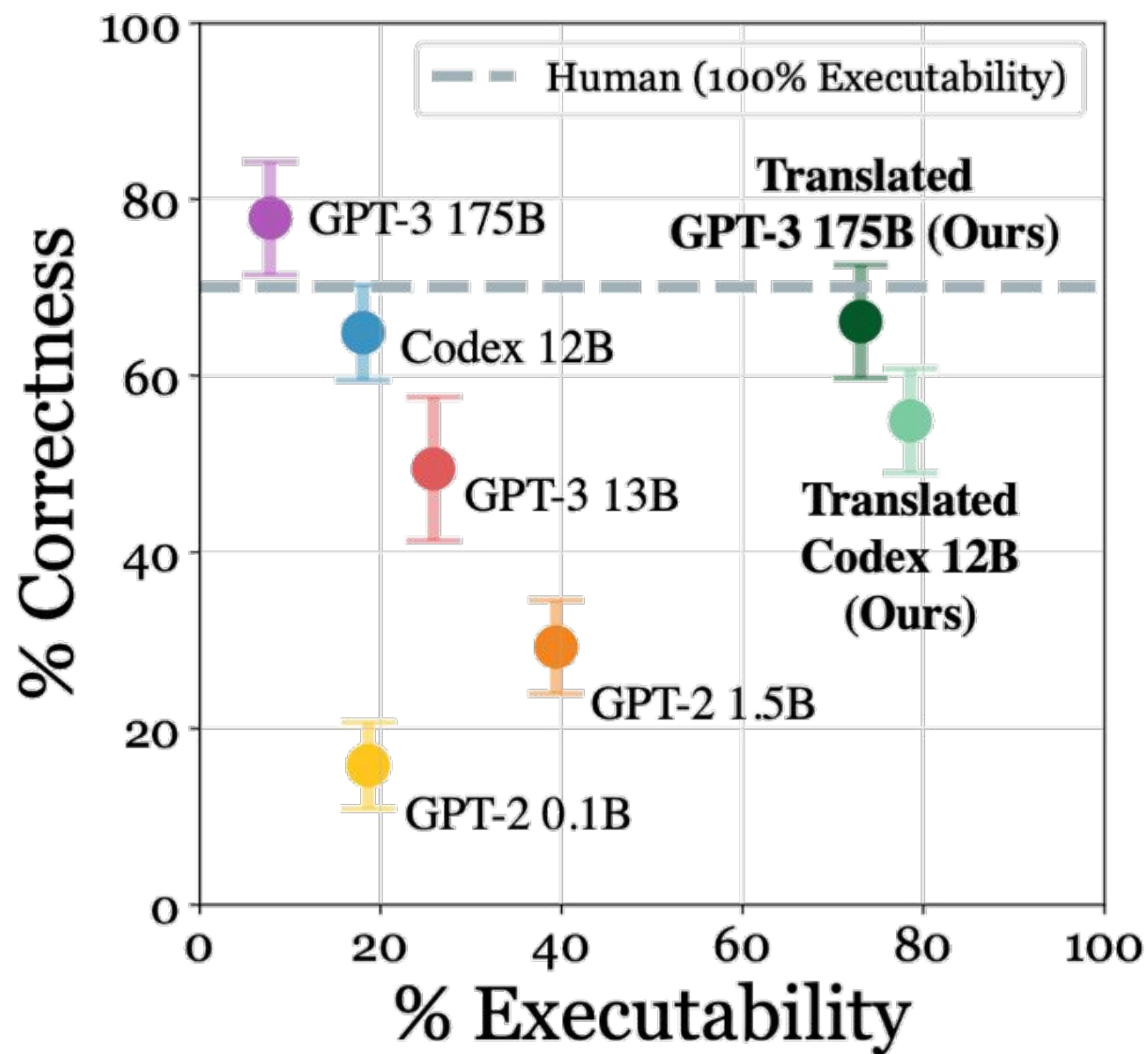
Task: Get Glass of Milk



Promoting Executability

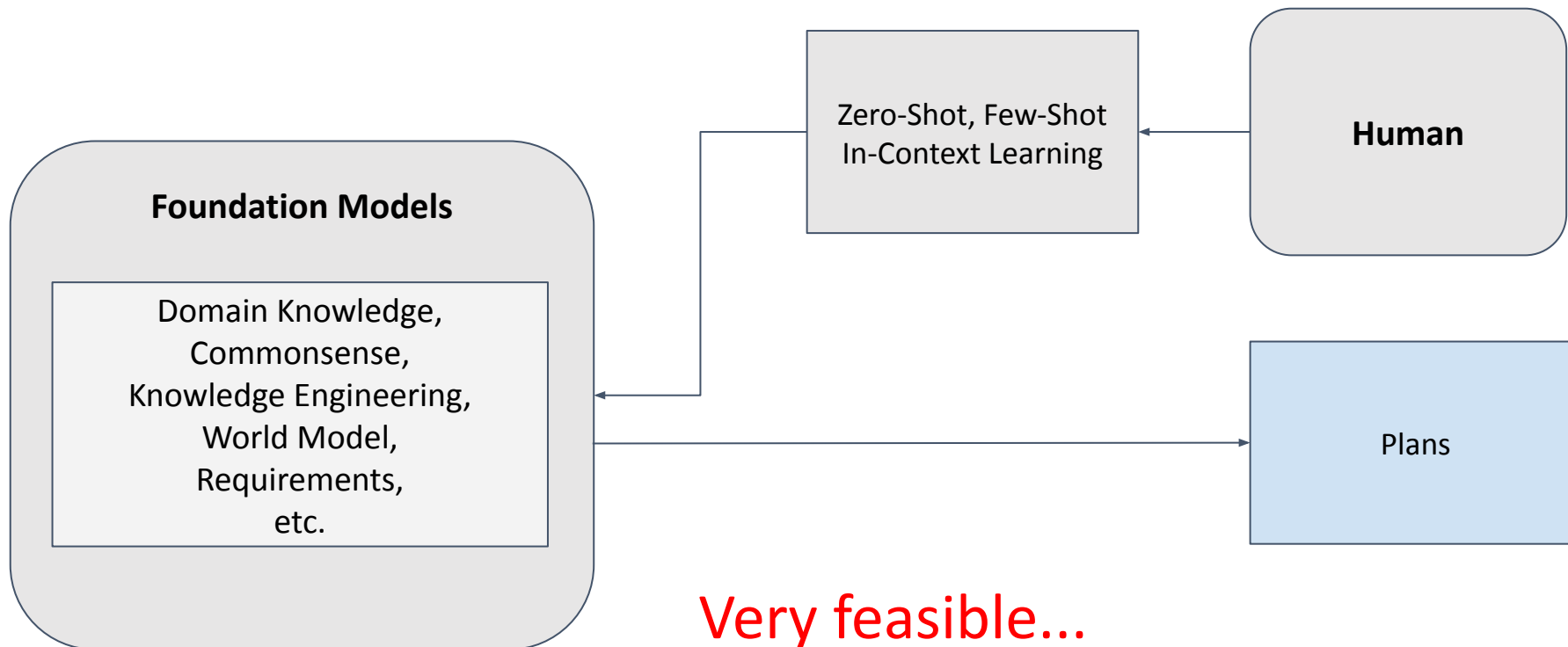


Performance for different models



Task Planning Today

- The planning models are **embedded somehow in the foundation models**.
 - **How** to do a task in which conditions... & **why** should it be done.
- The model is **open-world**.
- **Interaction in natural language!**
- Errors occur, **very hard to debug**.



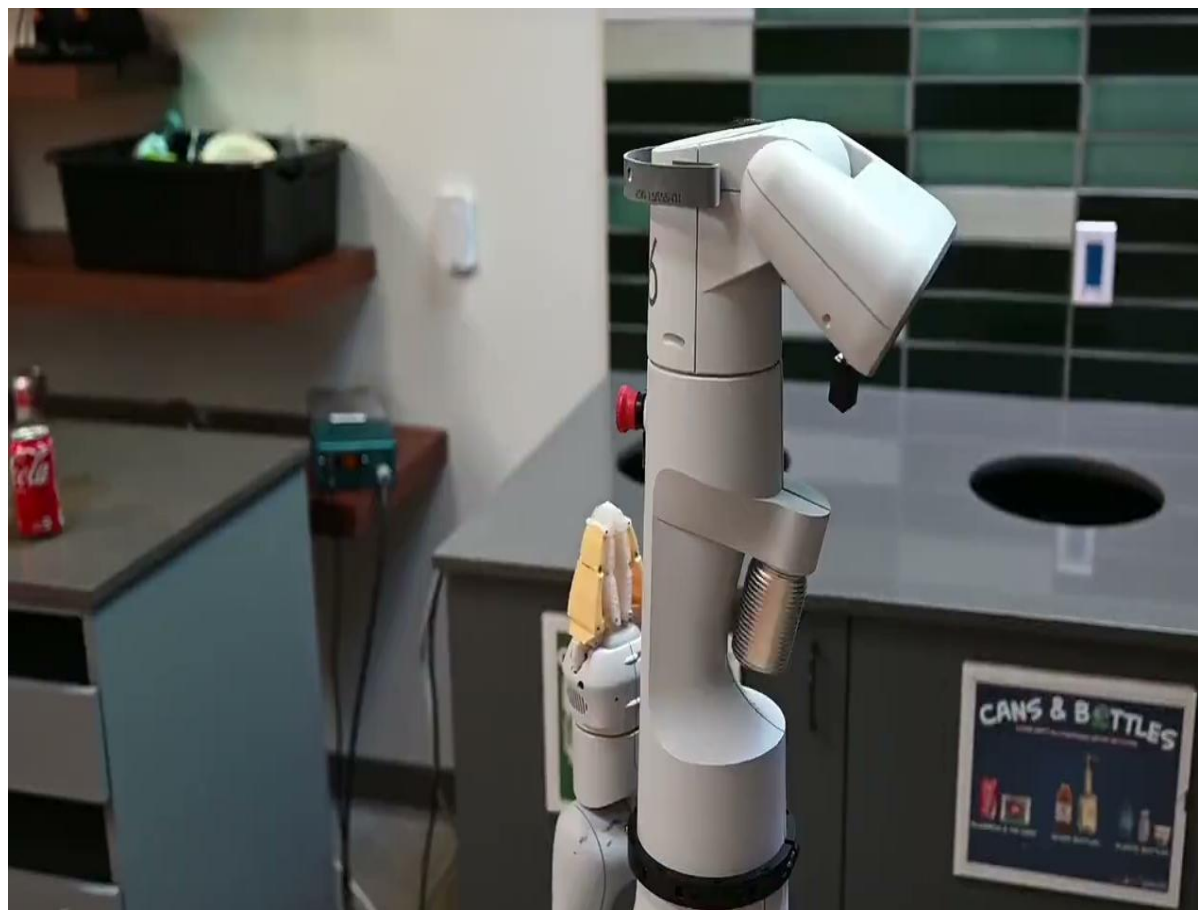
Very feasible...

To the Next Level!

- Executability
- Multi-modality
- Grounded reasoning
- Connecting Low-level skills
- Uncertainty
- etc.

Google SayCan (2022)

- Long-Horizon Task Planning using a LLM (GPT-3, FLAN, PaLM)
- Promoting Plan Validity by Visual Grounding and Affordance Estimation

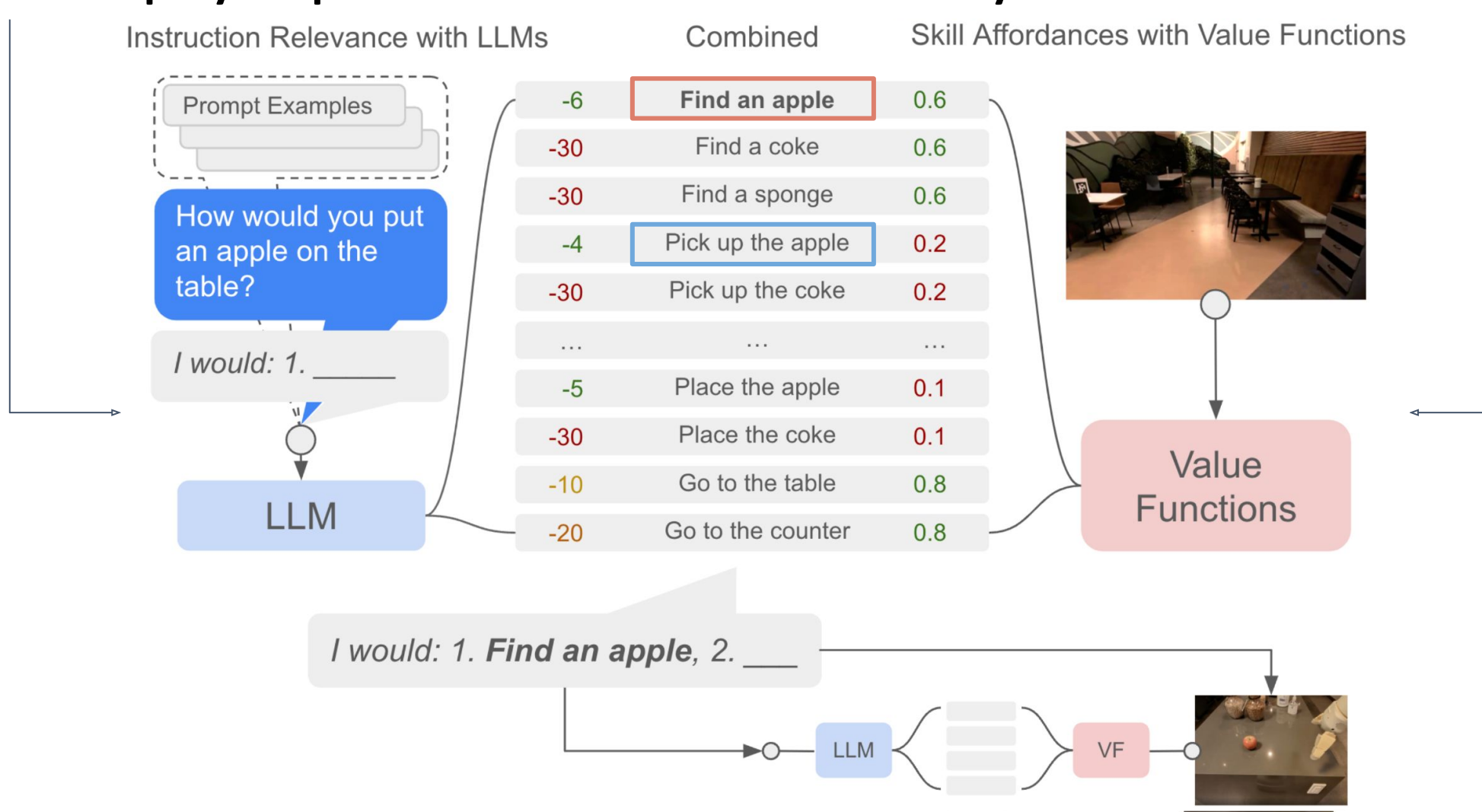


video link: https://say-can.github.io/img/demo_sequence_compressed.mp4

System Overview

LLM-based Step-by-Step Action Selection

Visually Grounded Affordance Estimation



System Overview

Algorithm 1 SayCan

Given: A high level instruction i , state s_0 , and a set of skills Π and their language descriptions ℓ_Π

```

1:  $n = 0, \pi = \emptyset$ 
2: while  $\ell_{\pi_{n-1}} \neq \text{"done"}$  do
3:    $\mathcal{C} = \emptyset$ 
4:   for  $\pi \in \Pi$  and  $\ell_\pi \in \ell_\Pi$  do
5:      $p_\pi^{\text{LLM}} = p(\ell_\pi | i, \ell_{\pi_{n-1}}, \dots, \ell_{\pi_0})$ 
6:      $p_\pi^{\text{affordance}} = p(c_\pi | s_n, \ell_\pi)$ 
7:      $p_\pi^{\text{combined}} = p_\pi^{\text{affordance}} p_\pi^{\text{LLM}}$ 
8:      $\mathcal{C} = \mathcal{C} \cup p_\pi^{\text{combined}}$ 
9:   end for
10:   $\pi_n = \arg \max_{\pi \in \Pi} \mathcal{C}$ 
11:  Execute  $\pi_n(s_n)$  in the environment, updating state  $s_{n+1}$ 
12:   $n = n + 1$ 
13: end while

```

▷ Evaluate scoring of LLM
 ▷ Evaluate affordance function

SayCan Prompt

Role Description

Robot: Hi there, I'm a robot operating in an office kitchen. You can ask me to do various tasks and I'll tell you the sequence of actions I would do to accomplish your task.

Context Info.

The following objects are in the scene: 7up, apple, tea, multigrain chips, kettle chips, jalapeno chips, rice chips, coke, grapefruit soda, Pepsi, Redbull, energy bar, lime soda, sponge, and water bottle. The following locations are in the scene: close counter, far counter, table, you, trash, bowl.

Human: Hold the sponge

Robot: 1. pick up the sponge, 2. done.

Human: Put the kettle chips here

Robot: 1. put down the kettle chips, 2. done.

In-Context Examples

Human: Move the grapefruit drink from the table to the close counter

Robot: 1. find a grapefruit soda, 2. pick up the grapefruit soda, 3. go to the counter, 4. put down the grapefruit soda, 5. done.

Human: Bring me some snacks

Robot: 1. find a jalapeno chips, 2. pick up the jalapeno chips, 3. bring it to you, 4. put down the jalapeno chips, 5. find an apple, 6. pick up the apple, 7. bring it to you, 8. put down the apple, 9. done.

Human: Bring me something that isn't a fruit

Robot: 1. find an energy bar, 2. pick up the energy bar, 3. bring it to you, 4. put down the energy bar, 5. Done.

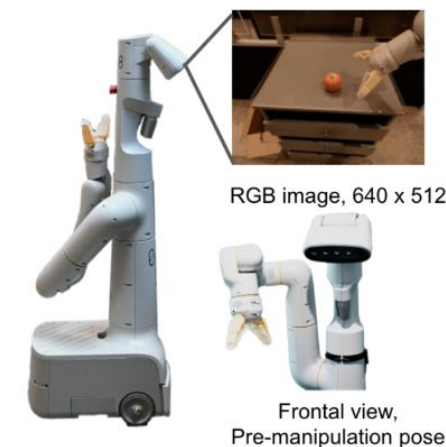
.....

Plan Request

Human: Put the rice chips in the bowl and then move the tea to the table

Robot: 1.

Evaluation: Environments and Instructions



Instruction Family	Num	Explanation	Example Instruction
NL Single Primitive	15	NL queries for a single primitive	Let go of the coke can
NL Nouns	15	NL queries focused on abstract nouns	Bring me a fruit
NL Verbs	15	NL queries focused on abstract verbs	Restock the rice chips on the far counter
Structured Language	15	Structured language queries, mirror NL Verbs	Move the rice chips to the far counter.
Embodiment	11	Queries to test SayCan's understanding of the current state of the environment and robot	Put the coke on the counter. (starting from different completion stages)
Crowd-Sourced	15	Queries in unstructured formats	My favorite drink is redbull, bring one
Long-Horizon	15	Long-horizon queries that require many steps of reasoning	I spilled my coke on the table, throw it away and bring me something to clean

Evaluation: Metrics

- **Plan Success Rate**

- This measures **whether the skills selected by the model are correct** for the instruction, regardless of whether or not they actually successfully executed.
- We ask **3 human raters** to indicate whether the plan generated by the model can achieve the instruction, and if **2 out of 3 raters agree that the plan is valid**, it is marked a success.

- **Execution Success Rate**

- This measures whether the full PaLM-SayCan system actually performs the desired instruction successfully.
- We ask **3 human raters** to watch the robot execution. The raters are asked to answer the question “whether the robot achieves the task specified by the task string?” We mark an execution successful if **2 out of 3 raters agree that it is successful**.

Performance

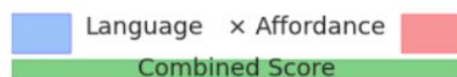
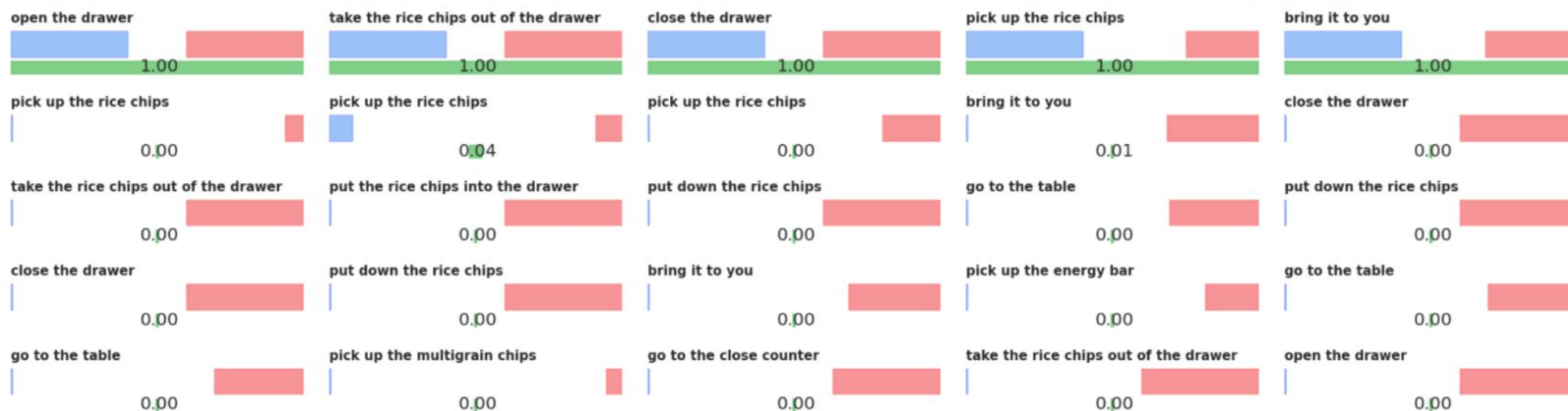
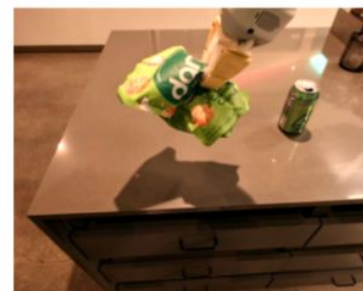
		Mock Kitchen		Kitchen		No Affordance		No LLM	
		PaLM-SayCan	PaLM-SayCan	PaLM-SayCan	PaLM-SayCan	No VF	Gen.	BC NL	BC USE
Family	Num	Plan	Execute	Plan	Execute	Plan	Plan	Execute	Execute
NL Single	15	100%	100%	93%	87%	73%	87%	0%	60%
NL Nouns	15	67%	47%	60%	40%	53%	53%	0%	0%
NL Verbs	15	100%	93%	93%	73%	87%	93%	0%	0%
Structured	15	93%	87%	93%	47%	93%	100%	0%	0%
Embodiment	11	64%	55%	64%	55%	18%	36%	0%	0%
Crowd Sourced	15	87%	87%	73%	60%	67%	80%	0%	0%
Long-Horizon	15	73%	47%	73%	47%	67%	60%	0%	0%
Total	101	84%	74%	81%	60%	67%	74%	0%	9%

Sample Execution

Human: Bring me the rice chips from the drawer.

Robot: I would

1. Go to the drawer
2. Open the drawer
3. Take the rice chips out of the drawer
4. Close the drawer
5. Pick up the rice chips
6. Bring it to you
7. Put down the rice chips
8. Done



New Capabilities

- **Easily expandable** by extending the skill set
- Improved performance simply **by prompt engineering**

CHAIN-OF-THOUGHT PLANNING ROLLOUT WITH PALM-SAYCAN.

Human: Can you bring a fruit-flavored drink without caffeine?

Explanation: The user has asked for a drink that is fruit-flavored and does not have caffeine, I will bring the lime soda.

Robot: 1. find a lime soda, 2. pick up the lime soda, 3. bring it to you, 4. put down the lime soda, 5. done

Human: Bring me the spicy chips.

Explanation: The user has asked for chips that are spicy, I will bring the Jalapeno chips.

Robot: 1. find the jalapeno chips, 2. pick up the jalapeno chips, 3. bring it to you, 4. put down the jalapeno chips, 5. done

Human: The snack is too small, bring me something more filling.

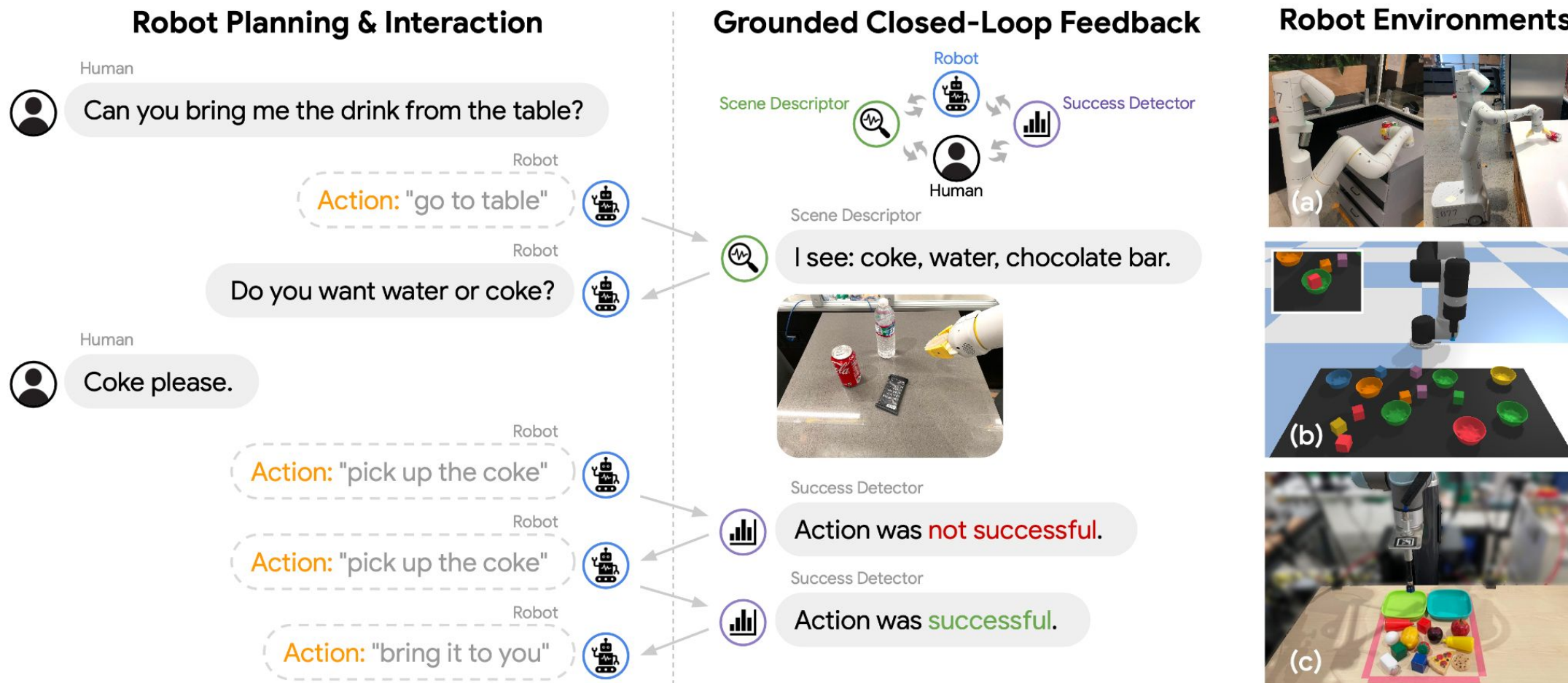
Explanation: The user has asked for a snack that is more filling, I will bring the multigrain chips.

Robot: 1. find the multigrain chips, 2. pick up the multigrain chips, 3. bring it to you, 4. put down the multigrain chips, 5. done

- **Multilingual** commands can be processed

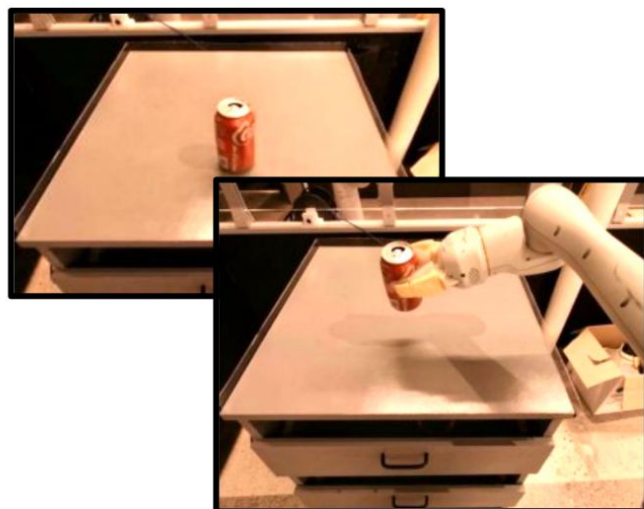
Google Inner Monologue (2022)

- Improving performance of LLM-based planner by **feedback from vision and human**



Sources of feedback

- Passive Scene Description
- Success Detection
- Active Scene Description



Success Detection in context

Robot Action: Pick up coke
Success: True
 Robot Action: Bring it to user



Passive Scene Description in context:

Robot Action: Go to table
Scene: lime soda, coke, energy bar
 Robot Action: pick up energy bar



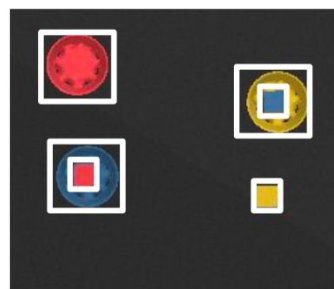
Active Scene Description in context:

Robot Action: Go to drawers
Robot Ask: Is the drawer open?
Human: The drawer is closed.
 Robot Action: Open the drawer

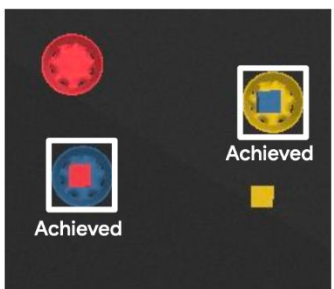
Experiments

Tabletop Rearrangement (Sim)

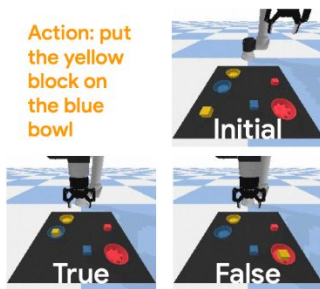
Human: move all the blocks into mismatching bowls.
 Scene: There is a yellow block, yellow bowl, blue block, blue bowl, red block, red bowl.
 Robot: My goal is ['yellow block in blue bowl', 'red block in yellow bowl', 'blue block in red bowl'].
 Robot: Pick up yellow block and place it in blue bowl.
 Scene: You achieved ['yellow block in blue bowl']
 Robot: I need red block in yellow bowl, blue block in red bowl
 Robot: Pick up red block and place it in yellow bowl.



Object Recognition



Task-Progress Scene Description

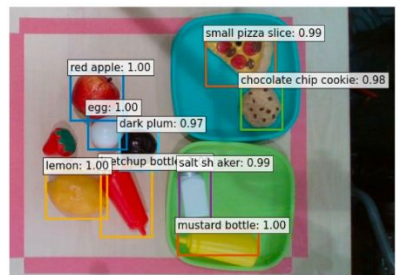


Success Detection

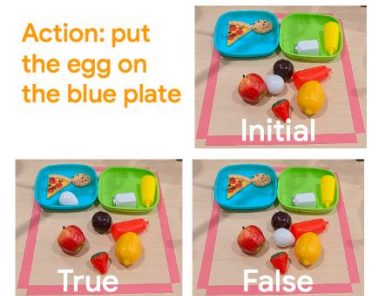
- InstructGPT for planning
- Scripted Modules for feedbacks

Tabletop Rearrangement (Real)

Human: Sort fruits and condiments in different plates
 Scene: objects = [apple, egg, plum, pizza, cookie, salt, mustard, ketchup, lemon]
 Scene: occluded objects = []
 Robot: robot.pickplace(strawberry, green plate)
 Robot: robot.pickplace(salt shaker, blue plate)
 Success Detector: False.
 Robot: robot.pickplace(salt shaker, blue plate)
 ...
 Robot: robot.stop()



Object Recognition (w/ Potential Occlusion)

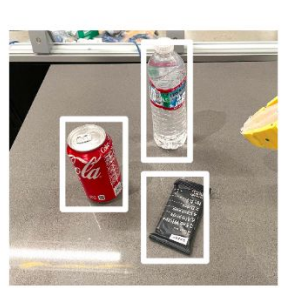


Success Detection

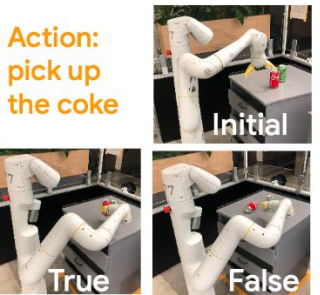
- InstructGPT for planning
- MDETR-based detector for Scene Description
- Heuristics for Success Detection

Kitchen Mobile Manipulation (Real)

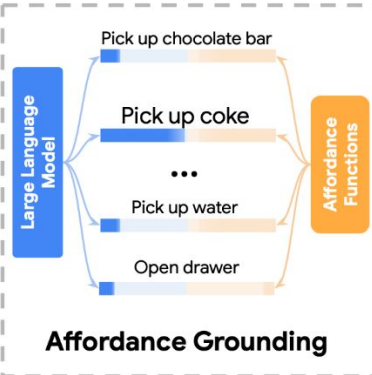
Human: Bring me a drink from the table
 Robot: go to the table
 Scene: I see a coke and lime soda
 Robot: What kind of drink would you like?
 Human: Something with caffeine
 Robot: Pick up coke
 Success Detector: False
 Robot: Pick up coke
 Robot: Bring it to you
 Robot: Put down the coke
 Robot: Done



Object Recognition



Success Detection



Affordance Grounding

- PaLM for planning
- Human-provided Scene Description
- Vision-based Success Detection

Results

Simulation

Tasks	CLIPort	+oracle	+LLM	+Inner Monologue		
			<i>Object</i>	<i>Object + Success</i>	<i>Object + Scene</i>	
Seen Tasks	“Pick and place”	24.0%	74.0%	80.0%	90.0%	94.0%
	“Stack all the blocks”	2.0%	32.0%	4.0%	10.0%	26.0%
	“Put all the blocks on the [x] corner/side”	2.0%	32.0%	30.0%	28.0%	30.0%
	“Put all the blocks in the [x] bowl”	32.0%	94.0%	52.0%	46.0%	56.0%
Unseen Tasks	“Put all the blocks in different corners”	0.0%	0.0%	20.0%	20.0%	26.0%
	“Put the blocks in their matching bowls”	0.0%	0.0%	56.0%	70.0%	82.0%
	“Put the blocks on mismatched bowls”	0.0%	0.0%	62.0%	76.0%	86.0%
	“Stack all the blocks on the [x] corner/side”	0.0%	0.0%	0.0%	4.0%	6.0%

Task Family	LLM	+Inner Monologue		
	<i>Object</i>	<i>Object</i>	<i>Success</i>	<i>Object + Success</i>
Finish 3-block stacking	20%	40%	40%	100%
Sort fruits from bottles	20%	50%	40%	80%
Total	20%	45%	40%	90%

Real-World (TableTop)

Real-World (Kitchen)

Task Family	SayCan	+Inner Monologue	
		<i>Success</i>	<i>Object + Success</i>
No Disturbances			
Manipulation	50.0%	62.5%	75.0%
Mobile Manipulation	50.0%	50.0%	75.0%
Drawers	83.3%	83.3%	100.0%
With Disturbances			
Manipulation	12.5%	25.0%	33.3%
Mobile Manipulation	0.0%	25.0%	75.0%
Drawers	0.0%	44.4%	44.4%
Total	30.8%	48.7%	60.4%

Emergent Capabilities

Human: throw away the snack on the close counter

Robot: 1. go to the close counter and ask: What snacks are on the counter?

Human: actually I changed my mind, I want you to throw away something on the table

2. go to the table and ask: What snacks are on the table?

Human: nevermind i want you to finish your previous task

3. go to the close counter and ask: What snacks are on the counter?

Human: apple and chip bag

4. pick up the apple and continue

5. go to the trash and continue

6. put down the apple and continue

7. done

Replan

(a) Continued Adaptation to New Instructions

Scene: There is a purple bowl, red block, purple block, blue block, orange bowl, orange block.

Human: Put any two blocks inside the purple bowl.

Robot thought: Goal state is ["Two blocks are inside the purple bowl."]

Robot action: Pick the red block and place it in the purple bowl.

Scene: Completed ["Red block is in purple bowl."]

Robot thought: One more block is needed to be inside the purple bowl.

Robot action: Pick the purple block and place it in the purple bowl.

Scene: The purple block is too heavy to be picked up. What's your new plan?

Robot thought: I need to find a lighter block.

Robot action: Pick the blue block and place it inside the purple bowl.

Scene: Completed ["Two blocks are inside the purple bowl."]

Robot thought: I'm done.

(b) Self-Proposing Goals under Infeasibility

Scene: There is a yellow bowl, yellow block, blue block, blue bowl.

Human: Put the blocks in the bowls with mismatching colors.

Robot thought: Goal state is ["Yellow block is in the blue bowl.", "Blue block is in the yellow bowl."]

Robot action: Pick the yellow block and place it in the blue bowl.

Scene: Completed ["Yellow block is in the blue bowl."]

Robot thought: Blue block is not in the yellow bowl.

Human: 我改变主意了, 请把蓝色方块也放到蓝色的碗里面。

Robot thought: Goal state is ["Yellow block is in the blue bowl.", "Blue block is in the blue bowl."]

Robot action: Pick the blue block and place it in the blue bowl.

Scene: Completed ["Yellow block is in the blue bowl.", "Blue block is in the blue bowl."]

Robot thought: I'm done.

*Translation: "I changed my plan. Please also put the blue block inside the blue bowl."

(c) Multilingual Interaction

Scene: There is a purple bowl, red block, purple block, blue block, orange bowl, orange block.

Human: Put any two blocks inside the purple bowl.

... Robot action: Pick the red block and place it inside the purple bowl.

... Robot action: Pick the purple block and place it inside the purple bowl.

Scene: The purple block is too heavy to be picked up. What's your new plan?

Robot thought: I need to find a lighter block.

Robot action: Pick the blue block and place it inside the purple bowl.

... Human: What objects are in the purple bowl?

Robot thought: The purple bowl contains the red block and the blue block.

Human: Where is the purple block?

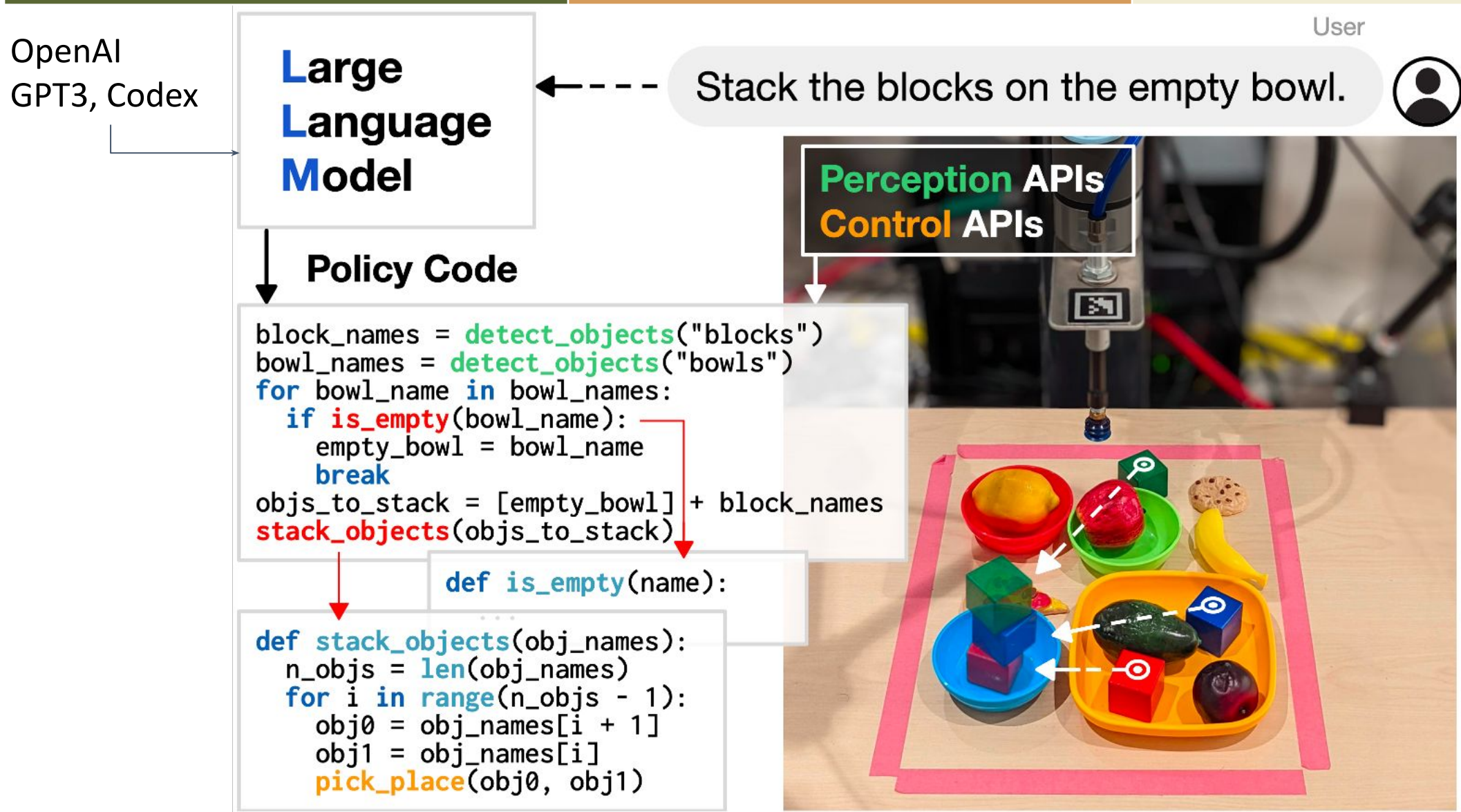
Robot thought: The purple block is on the table.

Human: Any objects in the orange bowl?

Robot thought: The orange bowl is empty.

(d) Interactive Scene Understanding

Google Code as Policies (2023)



OpenAI
GPT3, Codex

**Large
Language
Model**

User

Stack the blocks on the empty bowl.

Policy Code

```

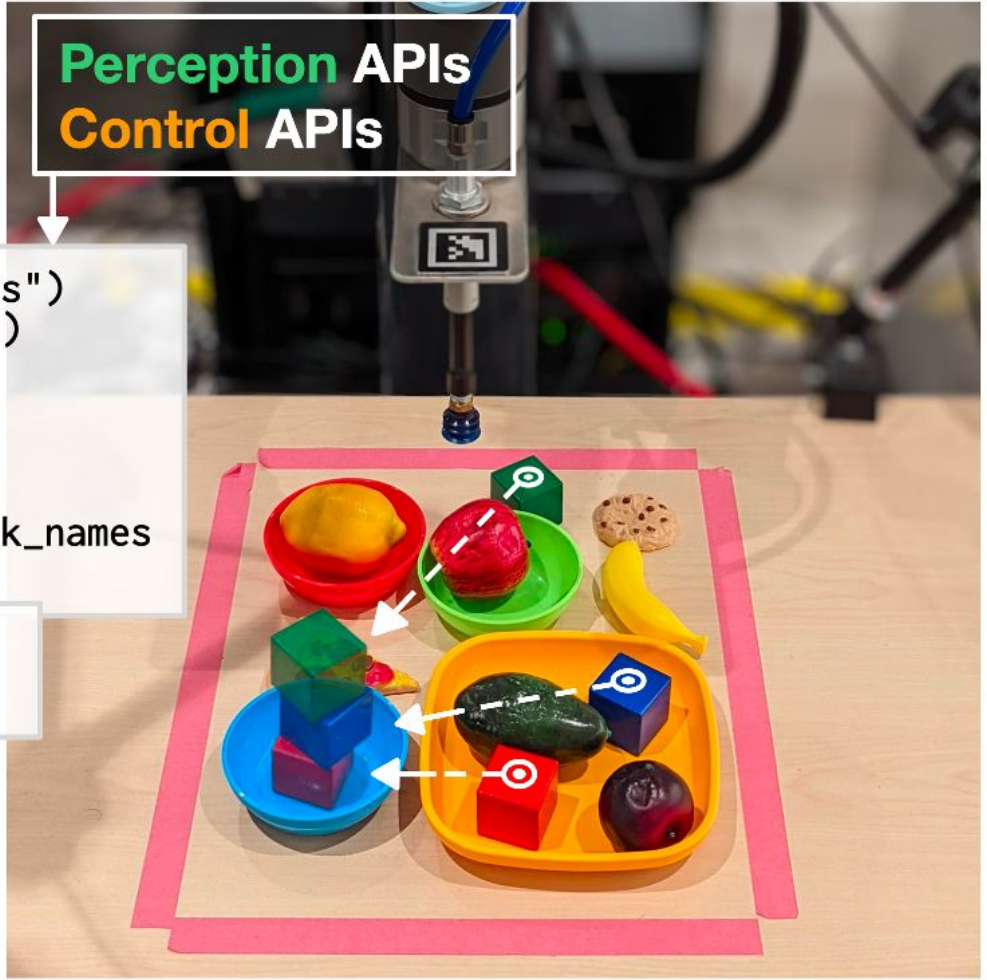
block_names = detect_objects("blocks")
bowl_names = detect_objects("bowls")
for bowl_name in bowl_names:
    if is_empty(bowl_name):
        empty_bowl = bowl_name
        break
objs_to_stack = [empty_bowl] + block_names
stack_objects(objs_to_stack)
    
```

```

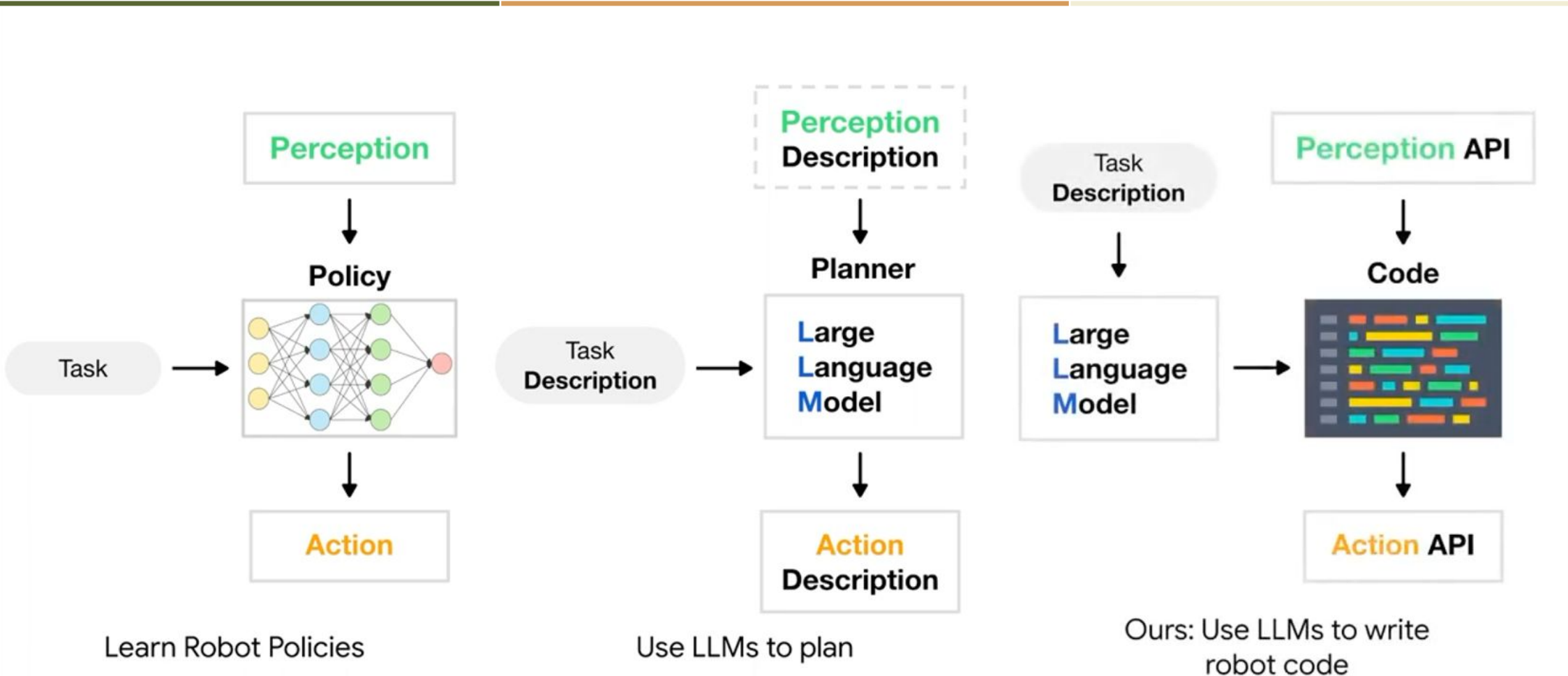
def is_empty(name):
    ...
    
```

```

def stack_objects(obj_names):
    n_objs = len(obj_names)
    for i in range(n_objs - 1):
        obj0 = obj_names[i + 1]
        obj1 = obj_names[i]
        pick_place(obj0, obj1)
    
```



Planning/Coding Methods (1/2)



Planning/Coding Methods (2/2)

LLM Plan [14], [17], [18]

1. Pick up coke can
2. Move a bit right
3. Place coke can

Socratic Models Plan [16]

```
objects = [coke can]
```

1. robot.grasp(coke can) open vocab
2. robot.place_a_bit_right()

Code as Policies (ours)

```
while not obj_in_gripper("coke can"):
    robot.move_gripper_to("coke can")
robot.close_gripper()
pos = robot.gripper.position
robot.move_gripper(pos.x, pos.y+0.1, pos.z)
robot.open_gripper()
```

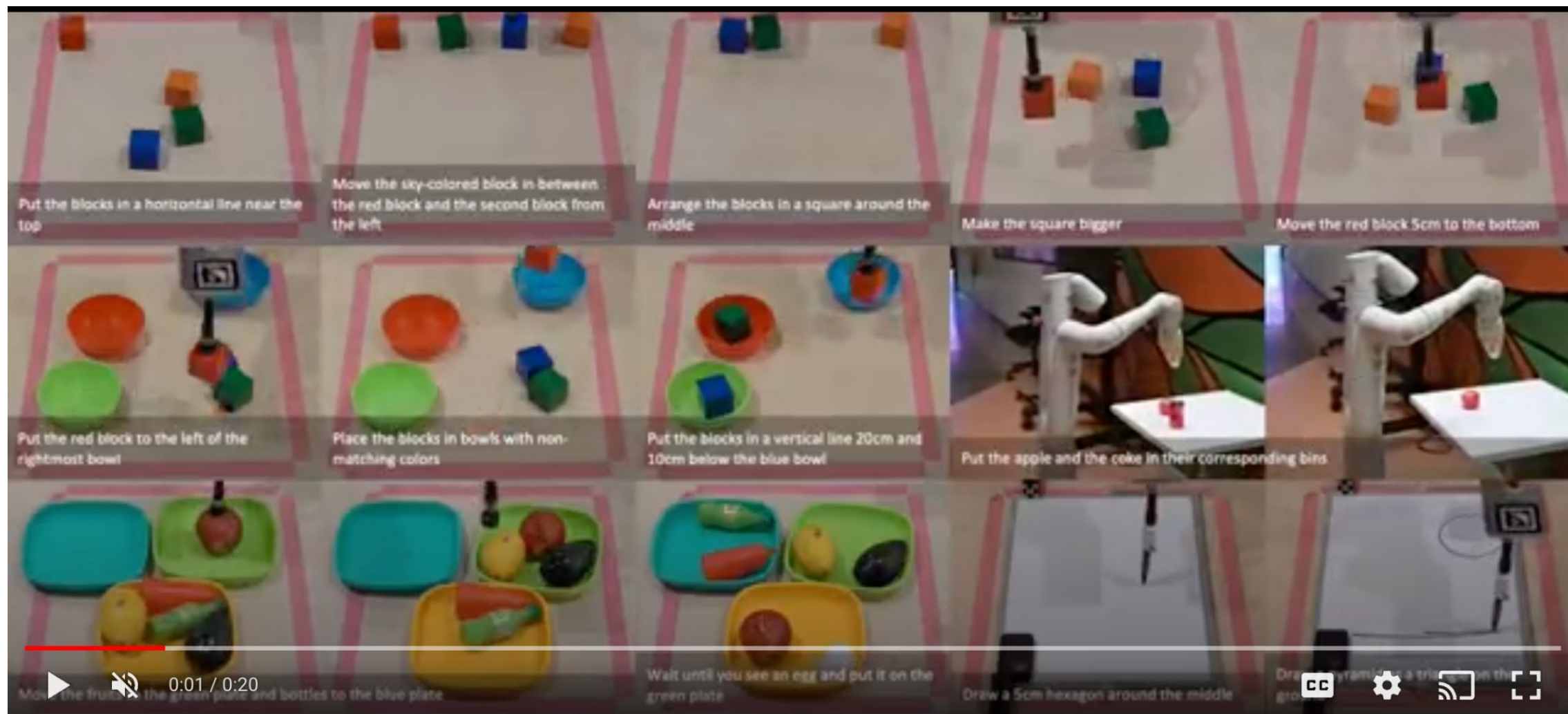
Prompts (excerpts)

```
# stack the blocks in the empty bowl.  
empty_bowl_name = parse_obj('empty bowl')  
block_names = parse_obj('blocks')  
obj_names = [empty_bowl_name] + block_names  
stack_objs_in_order(obj_names=obj_names)
```

```
# define function stack_objs_in_order(obj_names).  
def stack_objs_in_order(obj_names):  
    for i in range(len(obj_names) - 1):  
        put_first_on_second(obj_names[i + 1], obj_names[i])
```

```
# while the red block is to the left of the blue bowl, move it to the  
right 5cm at a time.  
while get_pos('red block')[0] < get_pos('blue bowl')[0]:  
    target_pos = get_pos('red block') + [0.05, 0]  
    put_first_on_second('red block', target_pos)
```

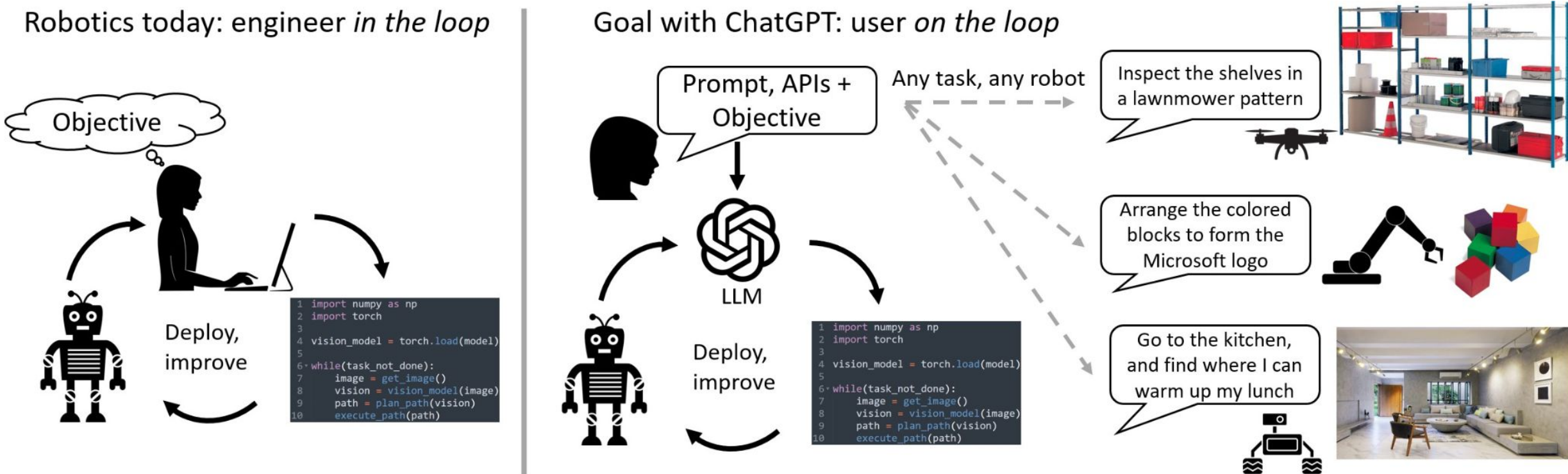
Demonstrations



video link: <https://code-as-policies.github.io/videos/tasks.mp4>

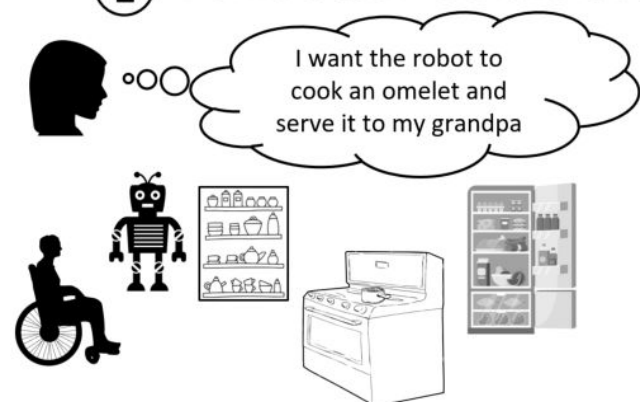
Microsoft ChatGPT for Robotics (2023)

- **User-In-The-Loop** way of developing robotics codes
- Users provide high-level feedback to the large language model (LLM) while monitoring the robot's performance.



Microsoft ChatGPT for Robotics (2023)

① Define a task-relevant robot API library*



```

1 def locate_object(obj_name):
2     # do something
3     return
4
5 def move_to_location(X,Y,Z):
6     # do something
7     return
8
9 def cook_item(obj_name):
10    # do something
11    return
12
13 def grab_object(obj_name):
14    # do something
15    return
    
```

*APIs should be easily implementable on the robot and have descriptive text names for the LLM. They can be chained together to form more complex functions.

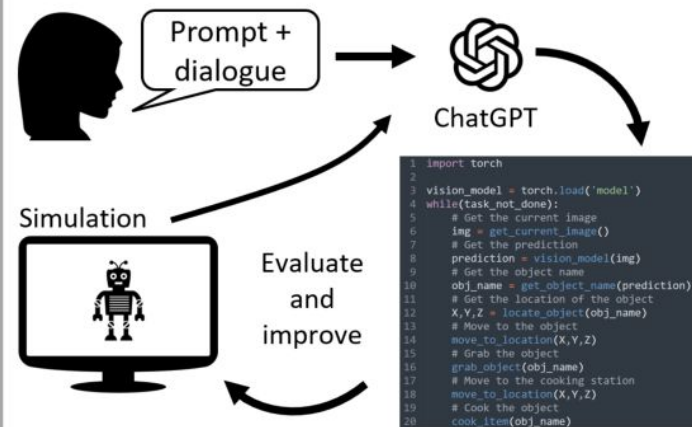
② Build prompt following engineering principles

Consider you are a home assistant robot. Your goal is to prepare an omelette for an elderly person. You are equipped with functions:

- `locate_object(obj_name)`: returns a X,Y,Z tuple representing the location of the desired object defined by string "obj_name";
- `move_to_location(X,Y,Z)`: moves the robot's hands to a specific X,Y,Z location in space. Returns nothing;
- `cook_item(obj_name)`: cooks a particular item defined by "obj_name". Returns nothing;
- `grab_object(obj_name)`: picks a particular object defined by "obj_name". Returns nothing;

Output python code with the sequence of steps that achieves your objective.

③ User on the loop: iterate on solution quality and safety



```

1 import torch
2
3 vision_model = torch.load("model")
4 while(task_not_done):
5     # Get the current image
6     img = get_current_image()
7     # Get the prediction
8     prediction = vision_model(img)
9     # Get the object name
10    obj_name = get_object_name(prediction)
11    # Get the location of the object
12    X,Y,Z = locate_object(obj_name)
13    # Move to the object
14    move_to_location(X,Y,Z)
15    # Grab the object
16    grab_object(obj_name)
17    # Move to the cooking station
18    move_to_location(X,Y,Z)
19    # Cook the object
20    cook_item(obj_name)
    
```

④ Execute!



project site: <https://www.microsoft.com/en-us/research/group/autonomous-systems-group-robotics/articles/chatgpt-for-robotics/>

Prompting

Imagine we are working with a household robot. The job of this robot is to make an omelette. The objects available around are:

fridge, bowl, pan, oil, stove

The main functions you can use are:

`locate_object(object_name)`: Returns the XYZ coordinates of an object of interest.

`go_to_location(object_name)`: Moves robot to a location specified by XYZ coordinates. Returns nothing.

`pick_up(object_name)`: Picks up the object of interest. Returns nothing.

`use_item(object_name)`: Takes the name of an object as input. Performs an appropriate action on the object at hand (pour oil, crack eggs). Returns nothing.

Can you make use of these to write code to go to the kitchen and make an omelette?

🗨️ Sure, here is an example Python code that utilizes the provided functions to prepare an omelette for an elderly person:

```
fridge_location = locate_object("fridge")
bowl_location = locate_object("bowl")
pan_location = locate_object("pan")
stove_location = locate_object("stove")
oil_location = locate_object("oil")

# Start the omelette-making process
go_to_location(fridge_location)
pick_up("eggs")
go_to_location(bowl_location)
use_item("eggs") # Crack eggs into the bowl
go_to_location(pan_location)
pick_up("pan")
use_item("pan") # Place the pan on the stove
go_to_location("oil")
pick_up("oil")
go_to_location(pan_location)
use_item("oil") # Pour oil into the pan
go_to_location("bowl")
pick_up("bowl")
```

Structured & Interactive Prompting

Imagine you are helping me interact with a mobile ground robot. You are in a room that contains several objects: three chairs, a glass table, a mirror, and two couches. All of your outputs need to be identified by one of the following tags:

<question> Always ask me a clarification questions if you are unsure </question>

<reason> Explain why you did something the way you did it </reason>

<command> Output code command that achieves the desired goal </command>

For example:

Me: Go to the chair

You: <question> There are three chairs in the world, which one do you want me to go to? </question>

Me: Go to "chair_3"

You: <command> target_location = `get_location("chair_3"); move_to_location(target_location);` </command> <reason> I found the location of the chair with the `get_location` function, and moved the robot towards the desired location </reason>

Are you ready?

🗨️ Yes, I am ready. Please give me your first command.

Go to the couch

🗨️ <question> There are two couches in the room. Which couch do you want me to go to? </question>

Go to the second sofa

🗨️ <command> target_location = `get_location("second_sofa"); move_to_location(target_location);` </command> <reason> I used the `get_location` function to find the location of the second sofa and then moved the robot towards that location. </reason>

Now move to the other one

🗨️ <command> target_location = `get_location("first_sofa"); move_to_location(target_location);` </command> <reason> I used the `get_location` function to find the location of the first sofa and then moved the robot towards that location. </reason>

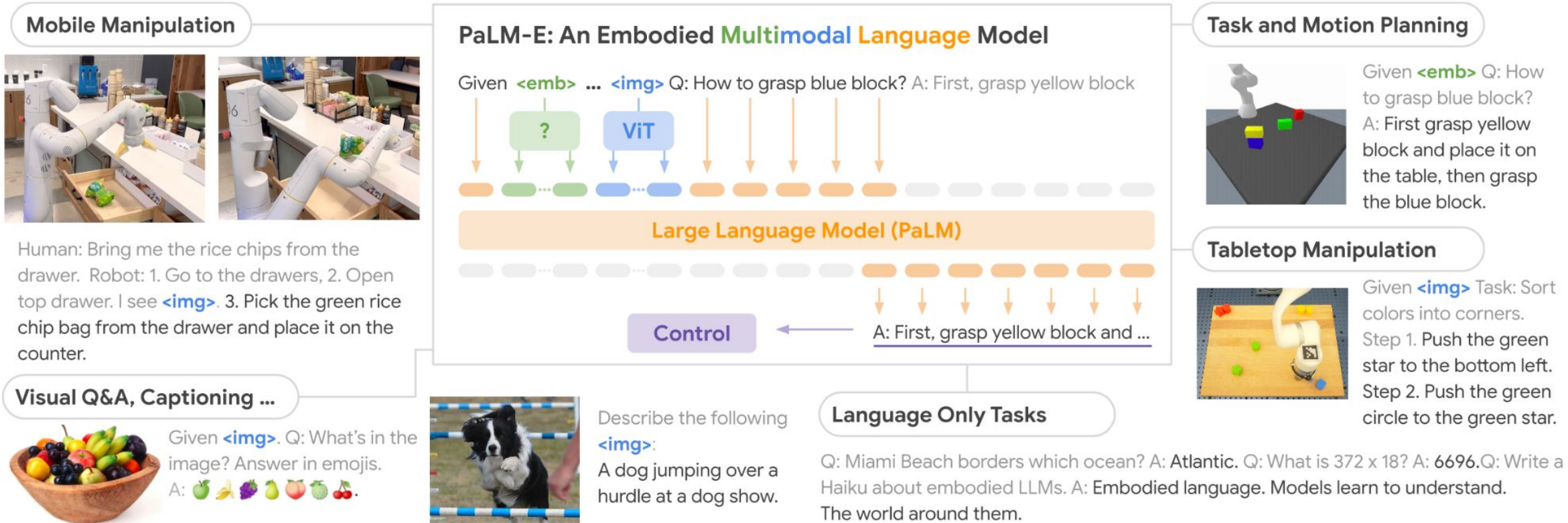
Demonstrations



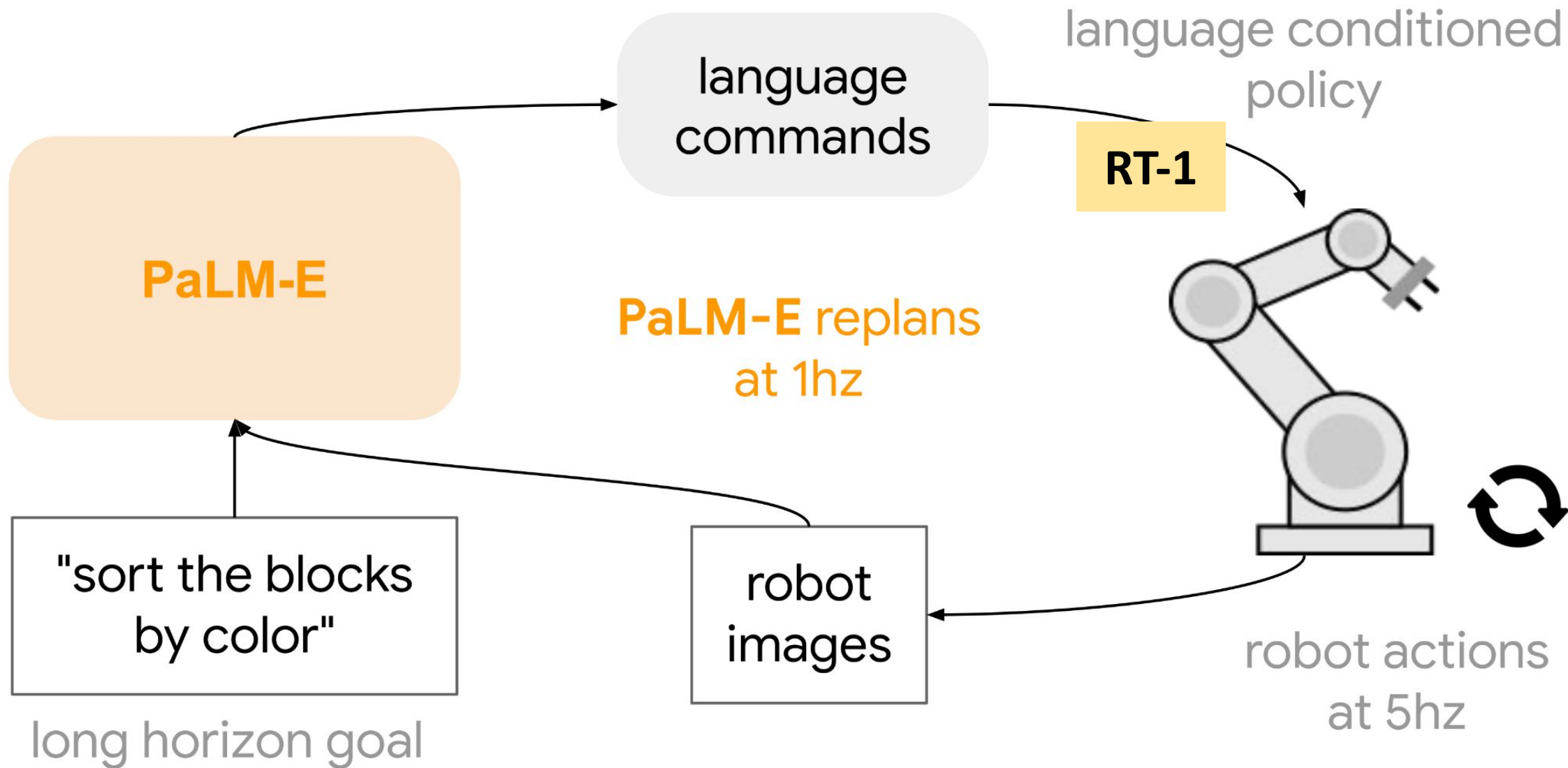
video link: <https://youtu.be/wLOChUtdqoA>

Google PaLM-E: An Embodied Multimodal Language Model (2023)

- LLM + ViT → Action Directions
- Integrated embodied reasoning: *affordance prediction, failure detection*
- Generalist Model: Planning, VQA, Image Captioning etc.



System Architecture



Embodied reasoning

start → goal

PaLM-E guiding a real robot through a long horizon mobile manipulation task

Instruction: "bring me the rice chips from the drawer"

Failure detection and Retry

Go to the drawers	Open the top drawer	Take the rice chips out of the drawer	<p style="color: red; font-weight: bold;">Adversarial Disturbance:</p> human knocks the rice chips back into the drawer	Take the rice chips out of the drawer	Bring it to the user	Put it down

success

PaLM-E guiding a real robot through one-shot and zero-shot tabletop manipulation tasks

Move the green circle to the yellow hexagon	Move the blue triangle to the group	<u>success</u>	Move the green star to the top left corner	Move the green star to the green circle	<u>success</u>

one-shot: "Move the remaining blocks to the group"

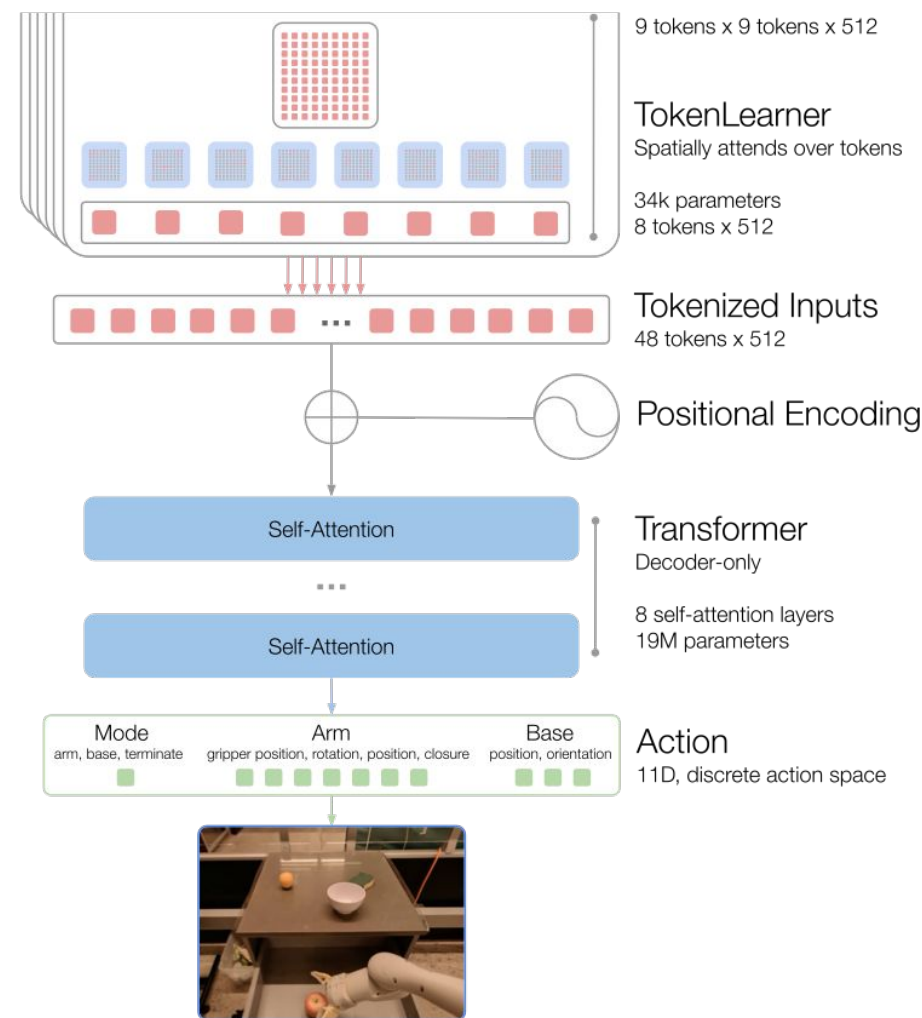
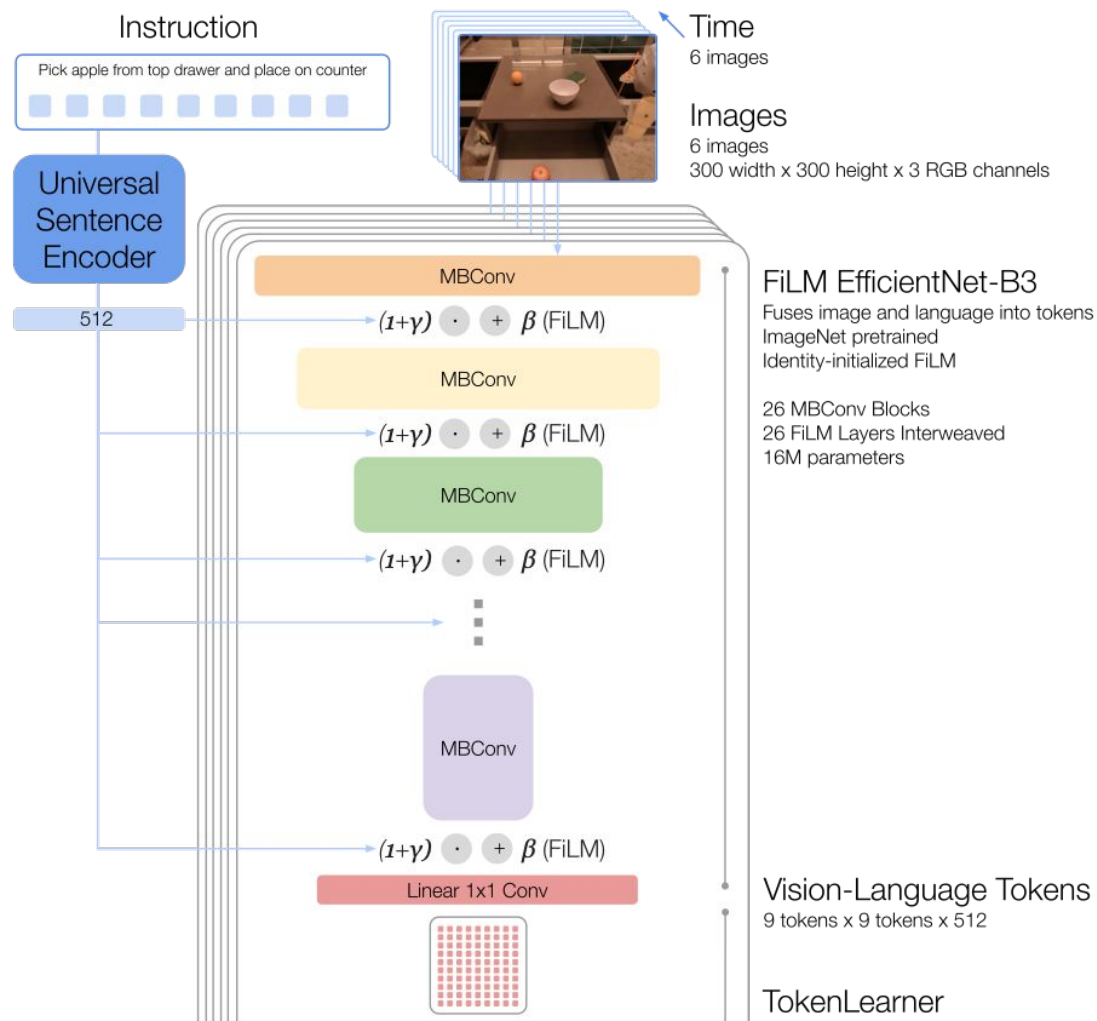
zero-shot: "Move the green blocks to the turtle"

Performance

	Object- centric	LLM pre-train	Embodied VQA				Planning	
			q ₁	q ₂	q ₃	q ₄	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	-	-	-
<i>PaLM-E</i> (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

RT-1: Robot Transformer 1

- Vision + Language → Control Commands



RT-1 Dataset

- **13 EDR robot** manipulators, each with a 7-degree-of-freedom arm, a 2-fingered gripper, and a mobile base
- **700+ Tasks**
- **130k episodes over 17 months**



(a)



(b)



(c)



(d)



(e)



(f)

RT-1 Tasks

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		

Evaluations

Evaluations	Methods
Seen Tasks	<ul style="list-style-type: none"> ● evaluates performance on 200 instructions sampled from the training set <ul style="list-style-type: none"> ○ 36 for picking objects, 35 for knocking objects, 35 for placing things upright, 48 for moving objects, 18 for opening and closing various drawers, and 36 for picking out of and placing objects into drawers ● involves varying the conditions (e.g., time of day, robot position)
Unseen Tasks	<ul style="list-style-type: none"> ● Evaluates performance on 21 novel, unseen tasks
Distractor Robustness	<ul style="list-style-type: none"> ● Evaluates with 30 instructions <ul style="list-style-type: none"> ○ pick coke can, place coke can upright, move coke can near green rice chip bag ● 3 levels of difficulty: easy (0-5 distractors), medium (9 distractors), hard (9 distractors and occluded object)
Background Robustness	<ul style="list-style-type: none"> ● Evaluates with 22 instructions ● 3 levels of difficulty: easy (original environment), medium (patterned tablecloth), hard (new kitchen)
Long-horizon Scenarios	<ul style="list-style-type: none"> ● 15 SayCan instructions in the real-world office kitchen <ul style="list-style-type: none"> ○ tasks involve max 10 steps

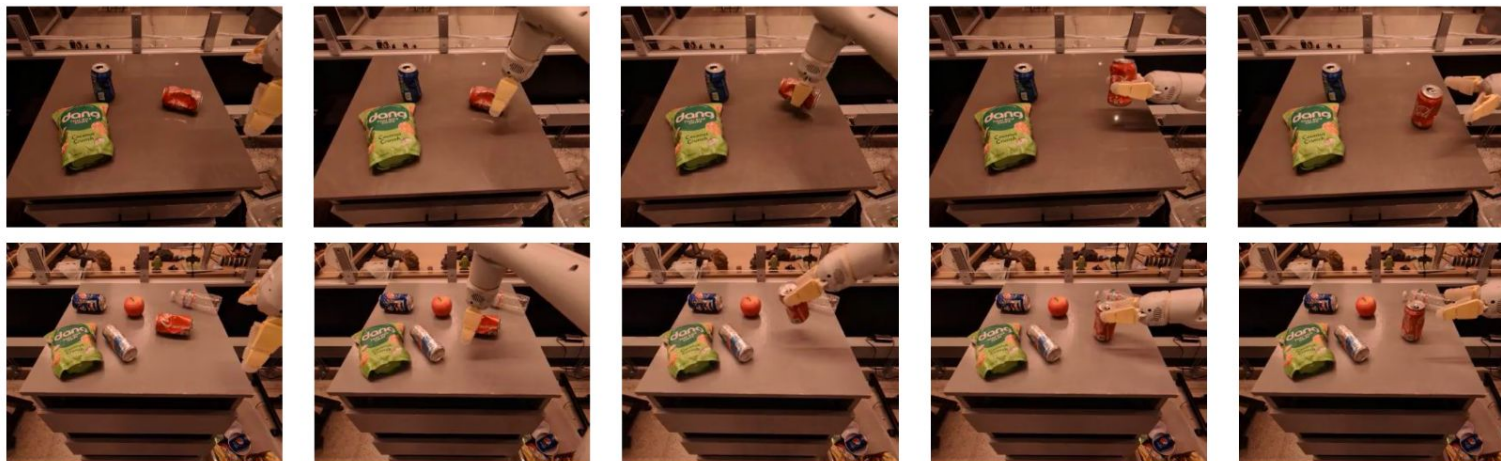
Unseen Commands

- 1.pick coke can from top drawer and place on counter
- 2.pick green can from top drawer and place on counter
- 3.pick green rice chip bag from middle drawer and place on counter
- 4.pick redbull can from top drawer and place on counter
- 5.place 7up can into bottom drawer
- 6.place brown chip bag into top drawer
- 7.place green can into middle drawer
- 8.move 7up can near redbull can
- 9.move apple near green rice chip bag
- 10.move apple near paper bowl
- 11.move apple near redbull can
- 12.move blue chip bag near blue plastic bottle
- 13.move blue chip bag near pepsi can
- 14.move blue chip bag near sponge
- 15.move brown chip bag near apple
- 16.move brown chip bag near green rice chip bag
- 17.move brown chip bag near redbull can
- 18.move coke can near green jalapeno chip bag
- 19.move coke can near water bottle
- 20.move green can near 7up can
- 21.move green can near apple
- 22.move green can near coke can
- 23.move green jalapeno chip bag near blue chip bag
- 24.move green rice chip bag near orange
- 25.move green rice chip bag near orange can
- 26.move green rice chip bag near paper bowl
- 27.move orange can near brown chip bag
- 28.move pepsi can near orange can
- 29.move redbull can near coke can
- 30.move rxbar blueberry near blue plastic bottle
- 31.move rxbar blueberry near orange can
- 32.move rxbar chocolate near paper bowl
- 33.move rxbar chocolate near rxbar blueberry
- 34.move sponge near apple
- 35.move water bottle near 7up can
- 36.move water bottle near sponge
- 37.move white bowl near orange can
- 38.pick blue plastic bottle
- 39.pick green rice chip bag
- 40.pick orange
- 41.pick rxbar chocolate
- 42.pick sponge
- 43.place pepsi can upright
- 44.knock orange can over
- 45.pick blue plastic bottle from paper bowl and place on counter
- 46.pick brown chip bag from white bowl and place on counter
- 47.pick green can from paper bowl and place on counter
- 48.pick green jalapeno chip bag from white bowl and place on counter
- 49.pick orange can from white bowl and place on counter
- 50.pick redbull can from white bowl and place on counter
- 51.place blue plastic bottle into paper bowl
- 52.place coke can into paper bowl
- 53.place orange can into paper bowl

Distractors

Easy

2 - 5 distractors,
no occlusion



Medium

9 distractors,
no occlusion



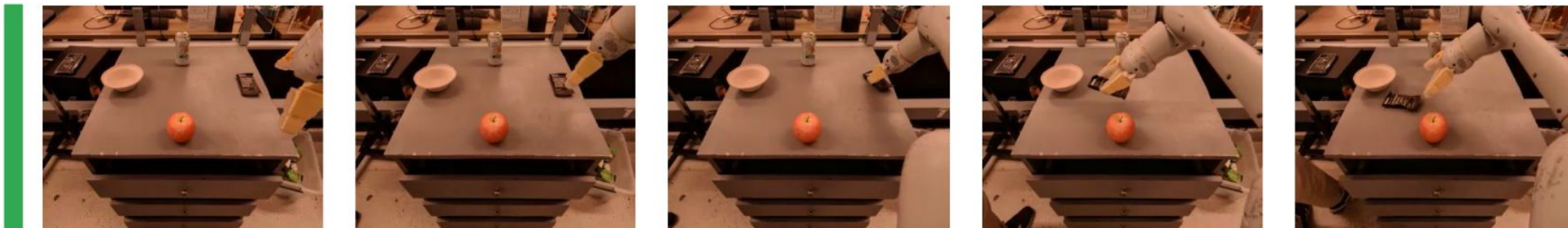
Hard

9 distractors,
occlusion



Backgrounds

Easy
same background,
same texture



Medium
same background,
new texture



Hard
new background,
new texture



Generalization

Level 1

Generalization

new real office kitchen with new lighting conditions



Level 2

Generalization

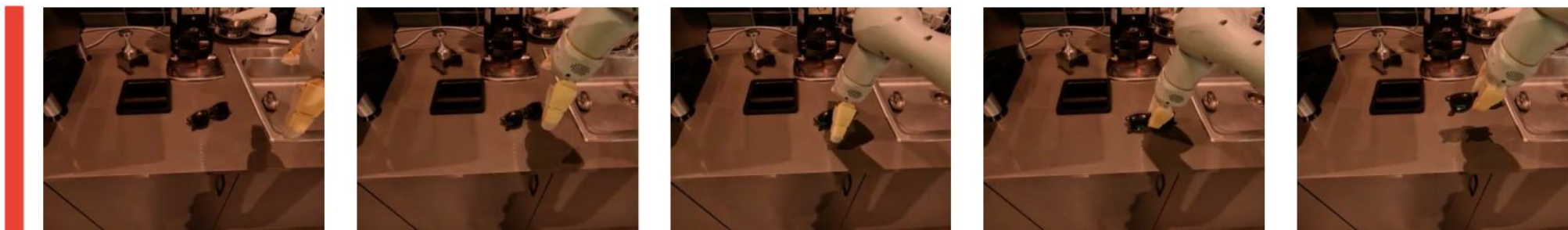
+ unseen distractor objects



Level 3

Generalization

+ new objects or objects in new locations, such as next to a sink



Overall Performance

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

Generalization Performance

Generalization Scenario Levels

Models	All	Generalization Scenario Levels		
		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

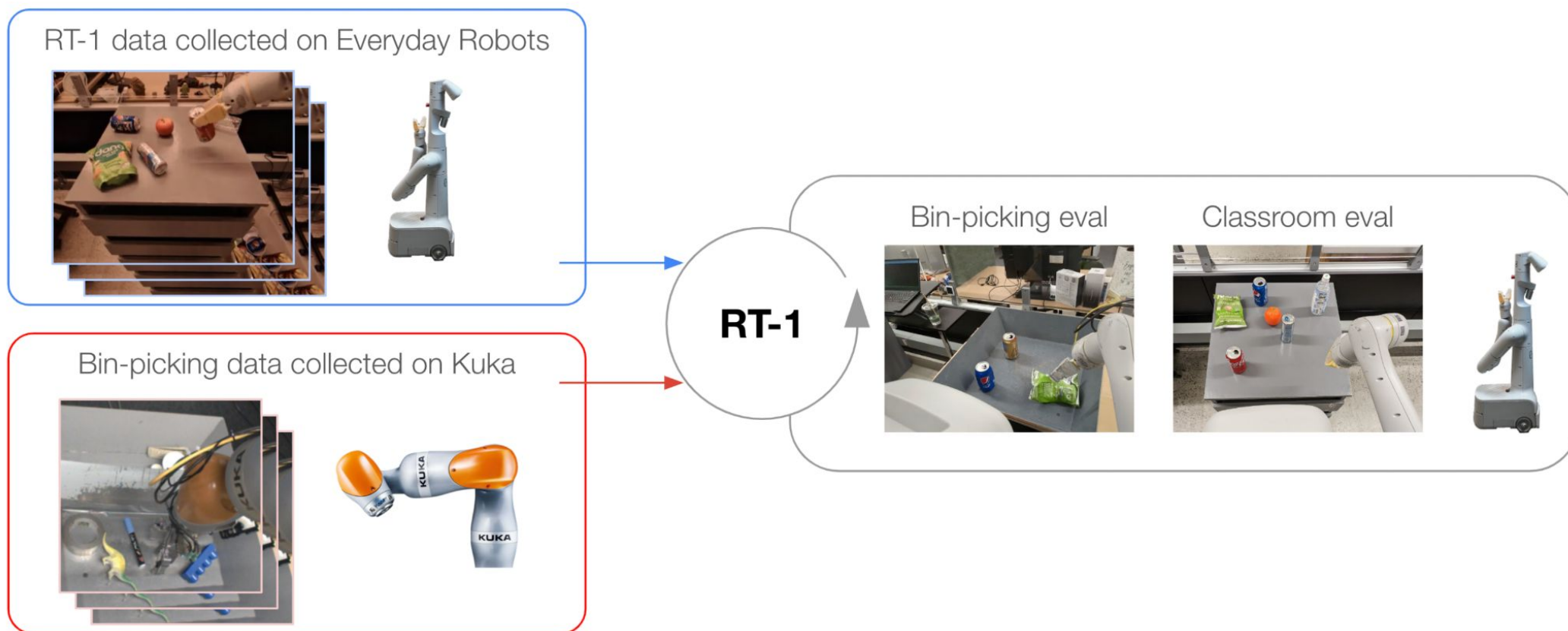
Data augmentation with synthetic data

- Performance improved for objects and tasks only seen in the simulation
→ RT-1 can effectively be augmented with synthetic data

Models	Training Data	Real Objects	Sim Objects (not seen in real)	
		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)

Generalization over Embodiment Gap

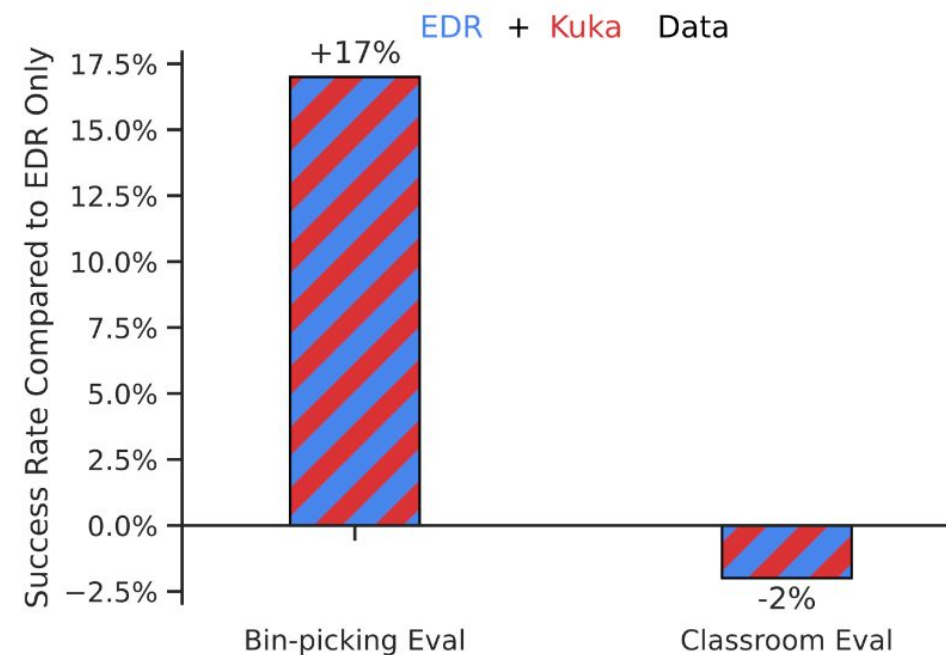
- RT-1 Dataset + Kuka Bin-Picking Dataset (209K Episodes)
- Evaluation of EDR Robot for Bin-Picking and Classroom Eval Tasks



Generalization over Embodiment Gap

- RT-1 Dataset + Kuka Bin-Picking Dataset (209K Episodes)
- Evaluation of EDR Robot for Bin-Picking and Classroom Eval Tasks

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



Google RT-2 (2023)

- Vision-Language-Action Model
- RT-1 on VLM (PaLI-X 5B, 55B ('23), PaLM-E 12B ('23))

Internet-Scale VQA + Robot Action Data



Q: What is happening in the image?
 A: 311 423 170 55 244
 A grey donkey walks down the street.

Q: Que puis-je faire avec ces objets?

A: 3455 1144 189 25673

Faire cuire un gâteau.



Q: What should the robot do to <task>?
 A: 132 114 128 5 25 156
 Δ Translation = [0.1, -0.2, 0]
 Δ Rotation = [10°, 25°, -7°]

RT-1 Dataset

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

Δ T = [0.1, -0.2, 0]
 Δ R = [10°, 25°, -7°]

Robot Action

Co-Fine-Tune

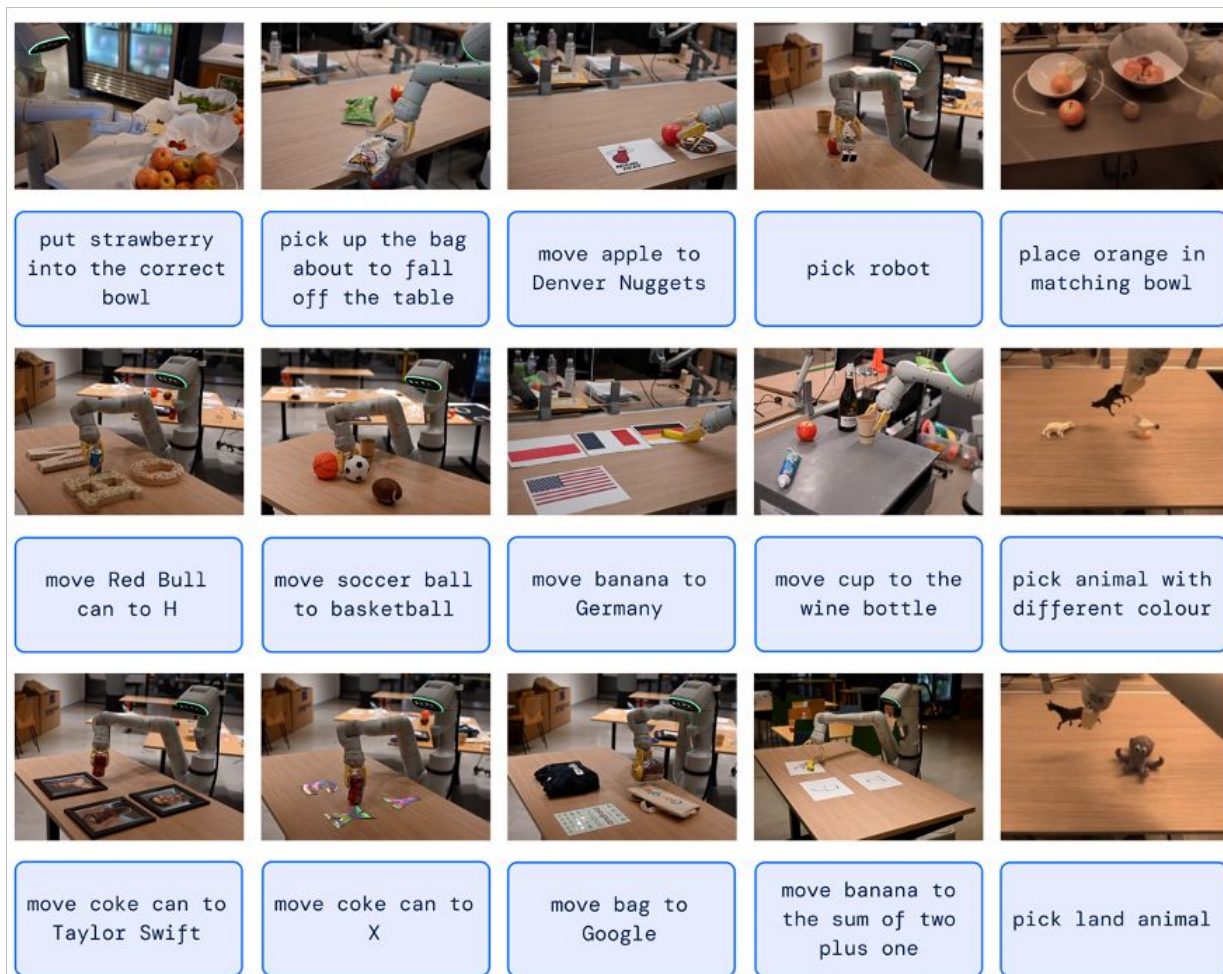
Deploy

Closed-Loop Robot Control

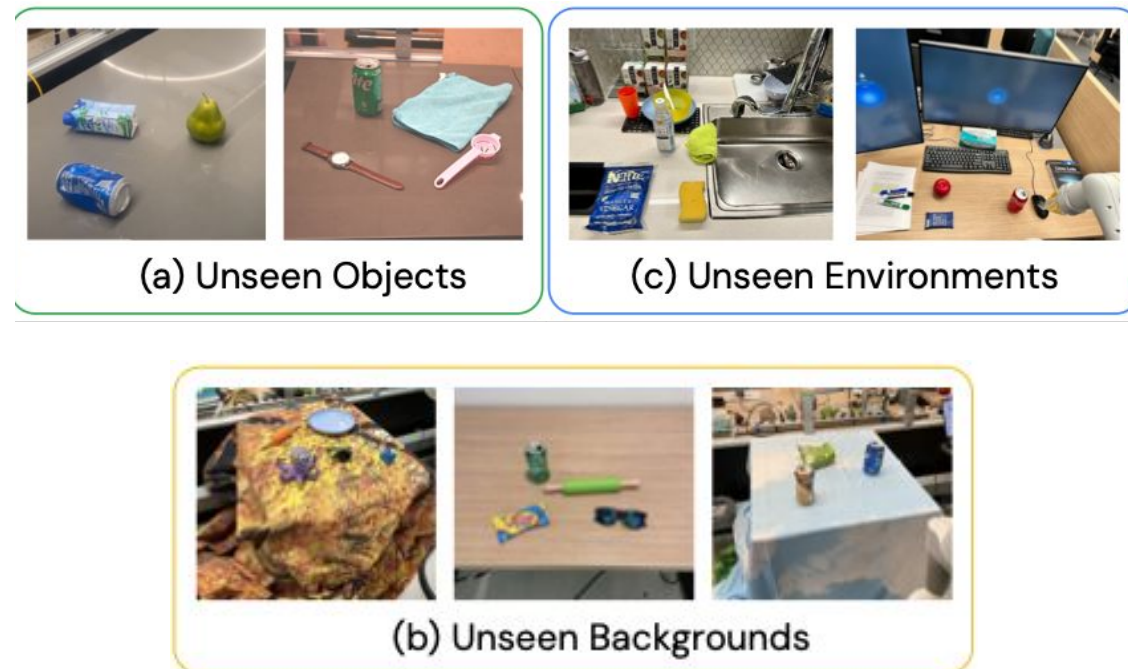


Evaluation

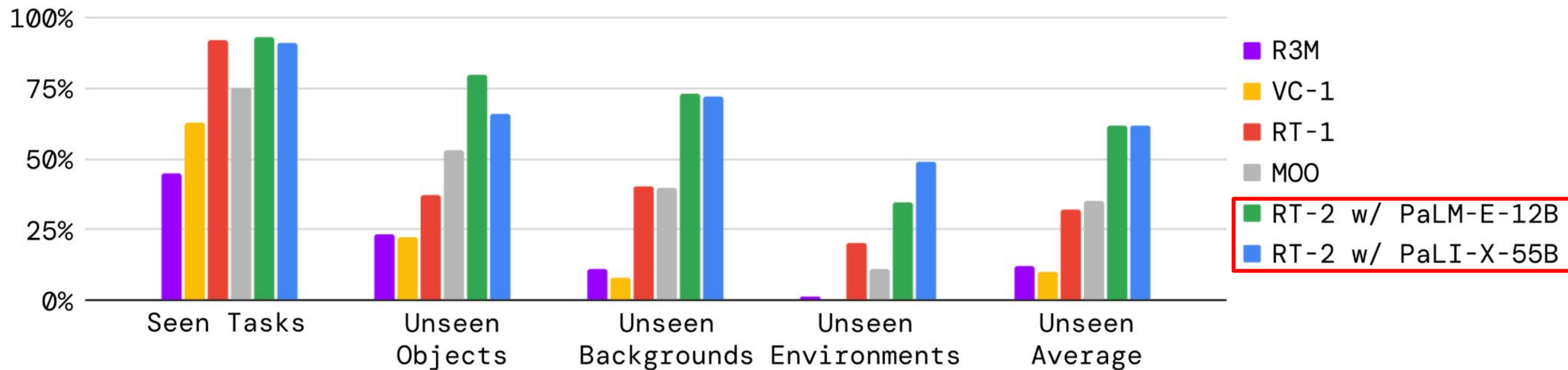
Does RT-2 generalize better than RT-1?



Novel Tasks



Performance



Emergent capability

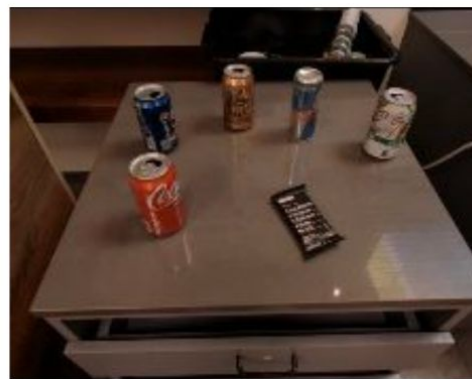
Prompt:
 Given Instruction:
 Bring me a drink.
 Prediction:
 Plan: pick 7up can.
 Action: 1 143 129 123 145
 114 115 127



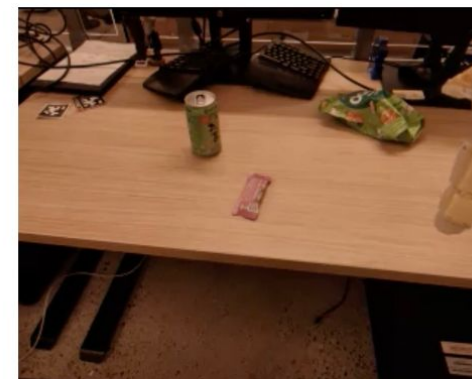
Prompt:
 Given Instruction:
 Move all the objects
 together.
 Prediction:
 Plan: move green can near
 green rice chip bag.
 Action: 1 128 126 127 135
 123 119 127



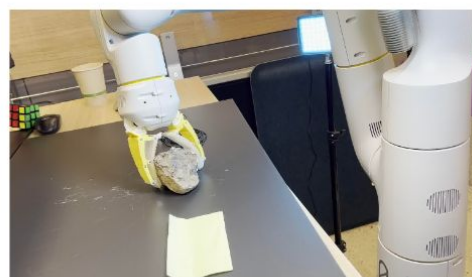
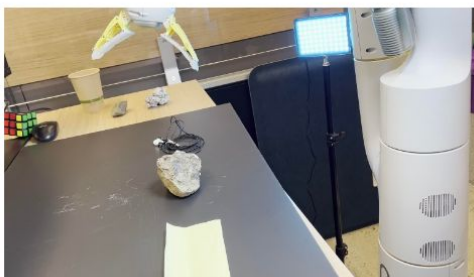
Prompt:
 Given Instruction:
 Pick the object that is
 different from all other
 objects
 Prediction:
 Plan: pick rxbar
 chocolate. Action: 1 128
 129 125 131 125 128 127



Prompt:
 Given 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 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



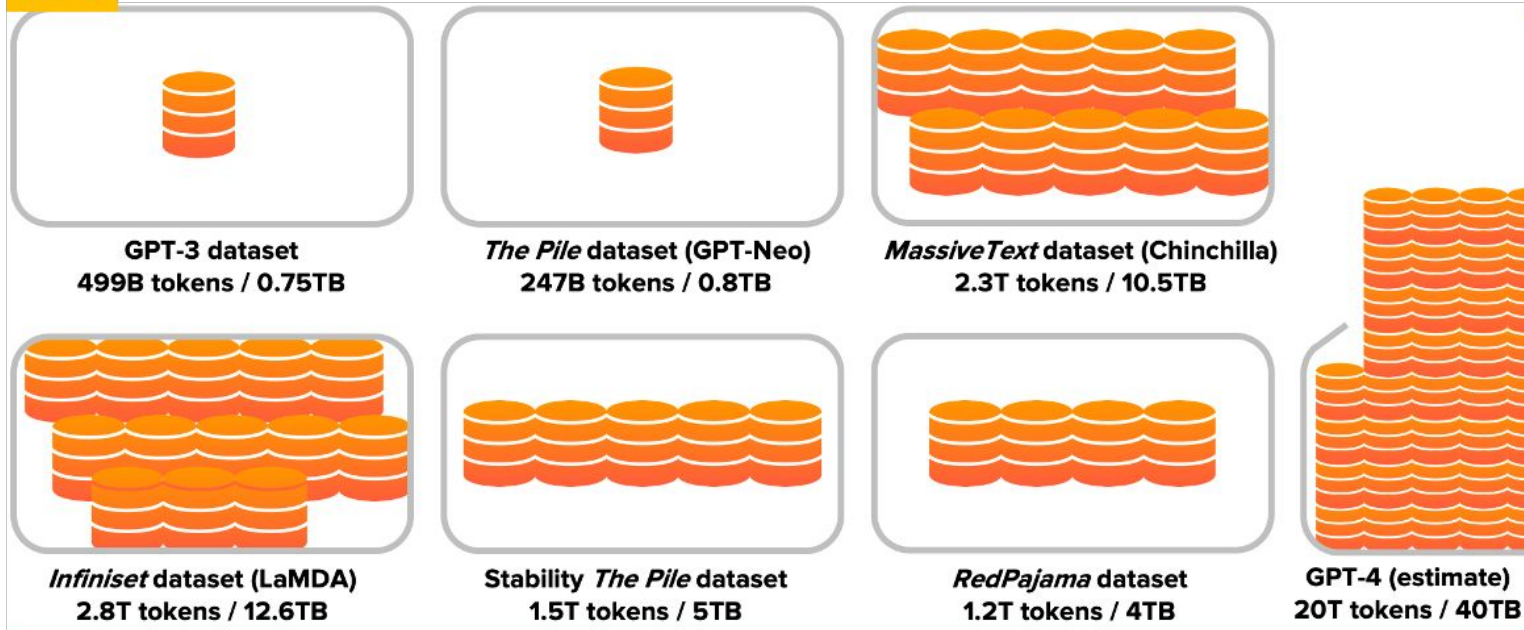
Issues in Foundation Models for Robotics

Research Issues

- Multi-Modality
 - Audio, Lidar, Depth, Haptic, UWB, Ultrasonic,
- Context Length
 - Long-Horizon Tasks + Many Modality...
- Real-time
 - Optimizations needed for RT-1, RT-2 to make them work in 2~3hz
- Safety, Ethics, Trustworthiness, Responsibility
 - Alignment with human and social values
 - How can we validate?

Robot Data at Scale

언어



LifeArchitect.ai/models

영상

Method	Public	Multimodal Pretraining data	
		Dataset(s)	Size
CLIP [215]	✗	WebImageText [215]	400M
ALIGN [129]	✗	ALIGN1.8B [129]	1800M
WenLan [123]	✗	RUC-CAS-WenLan [123]	30M
Florence [321]	✗	FLD-900M [321]	900M
FILIP [311]	✗	FILIP300M [311], CC3M [235], C12M [30], YFCC100M [258]	340M
SLIP [200]	✓	YFCC15M [258, 214]	15M
FLIP [160]	✓	LAION400M [226]	400M
MaskCLIP [67]	✓	YFCC15M [258, 214]	15M
CLIPA [159]	✓	LAION-400M [226]	400M
CLIPAv2 [158]	✓	LAION-2B [226], DataComp-1B [83]	3000M
EVA [80]	✓	IN21K [82], CC12M [30], CC3M [235], O365 [234], COCO [163], ADE [356]	29.6M
EVA-CLIP [249]	✓	Merged-2B [249]	2000M
EVA-02 [79]	✓	Merged-2B [249]	2000M
OpenCLIP [49]	✓	LAION-400M [226] LAION-5B [227]	5400M

로봇



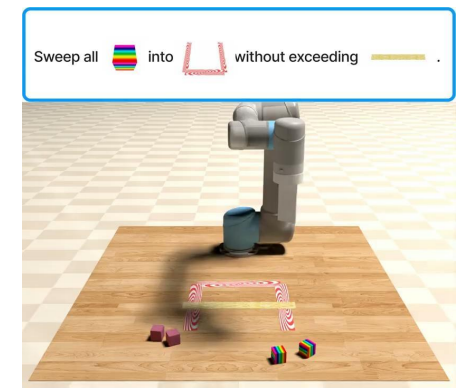
[Video](#)

RT-1
130K
(real)



[Video](#)

Language-Table
~600K
(real+synth)




[Video](#)

VIMA
650K
(synth)

Stanford PhysObjects (2023)

- Object-centric dataset of 36.9K crowd-sourced and 417K automated physical concept annotations of common household objects

Distance, Camera Motion, Background Complexity, Lighting

Sponsored by  Meta



near, horizontal, simple, bright



medium, vertical, busy, bright



medium, diagonal, busy, bright



near, horizontal, simple, dim



medium, horizontal, busy, dim



far, horizontal, busy, dim

Google Diffusion Rosie (2023)

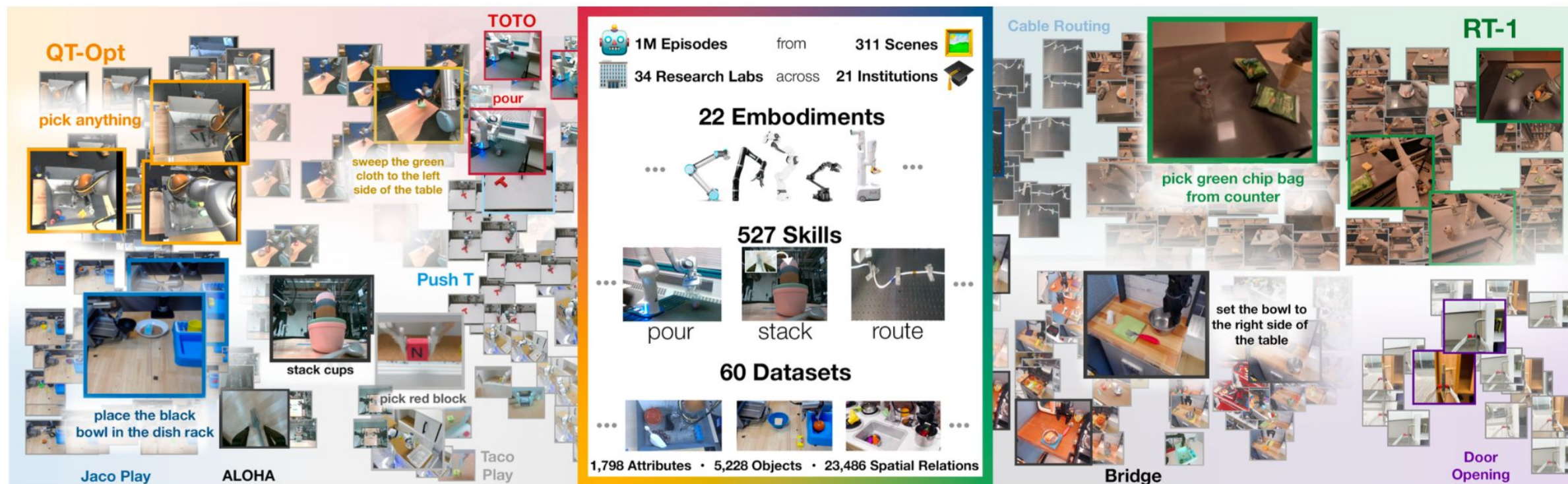
- Realistic Scene and Object Synthesis using Diffusion



video link: https://diffusion-rosie.github.io/videos/coke_compressed.mp4

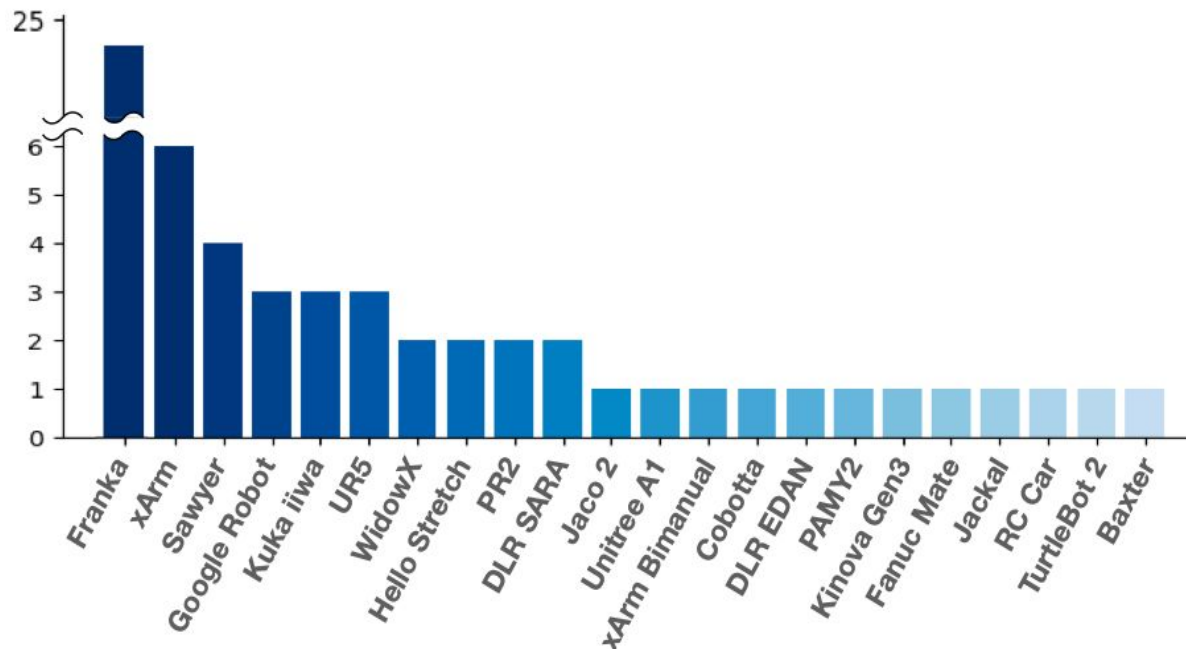
Open-X Embodiment Dataset (2023)

- Open, large-scale dataset for robot learning curated from **21 institutions** across the globe
- **X-Embodiment Robotic Learning**: diversity in behaviors, robot embodiments and environments → **Generalized robotic policies**

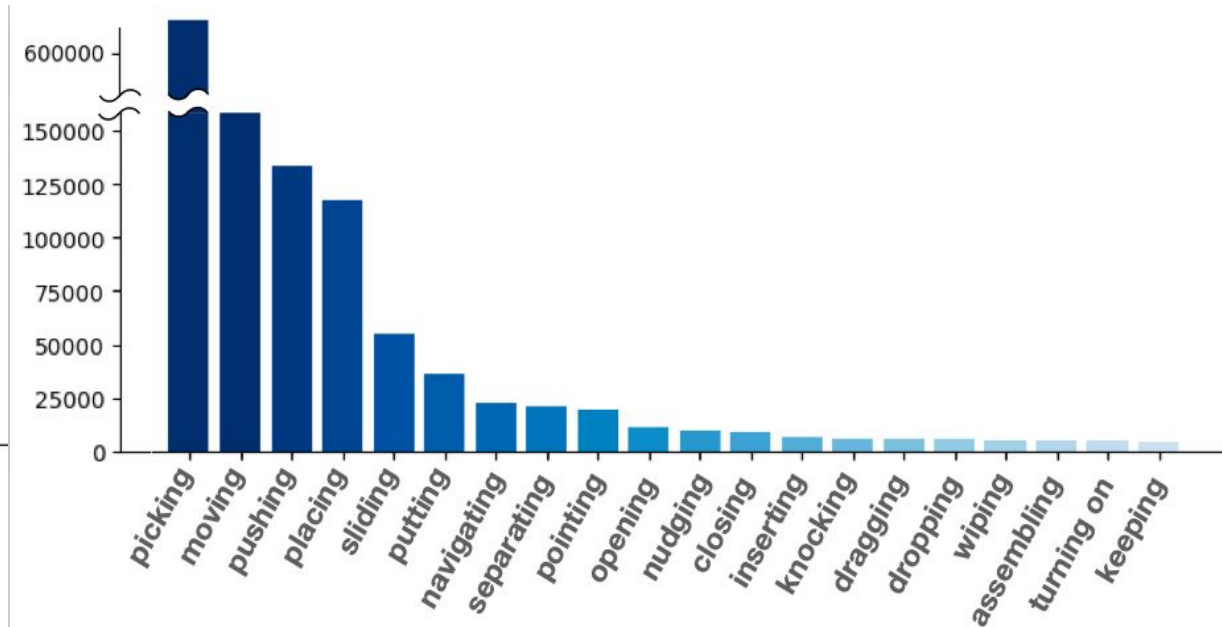


Open-X Embodiment Dataset

- Robots and Skills



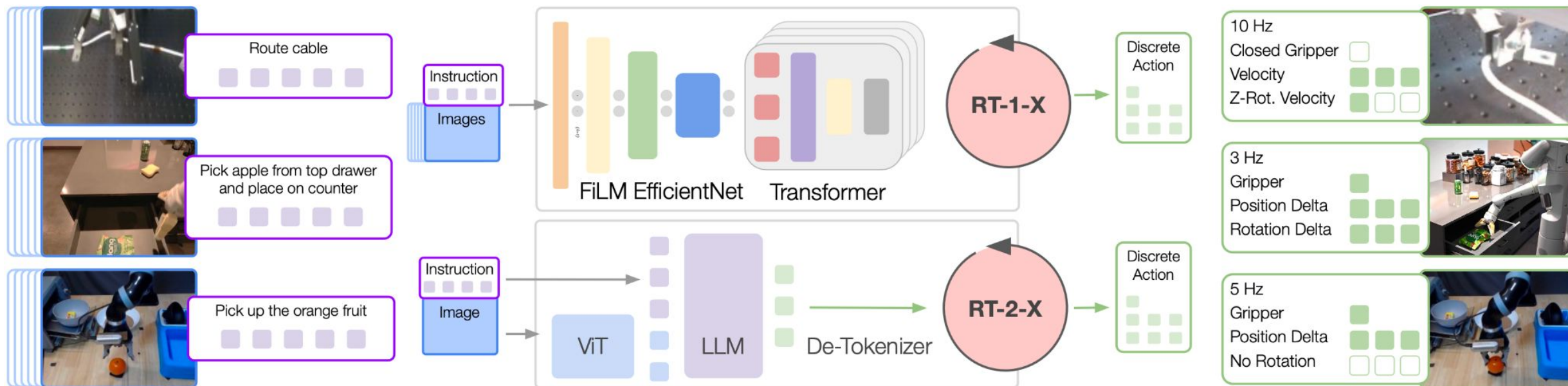
(a) # Datasets per Robot Embodiment



(d) Common Dataset Skills

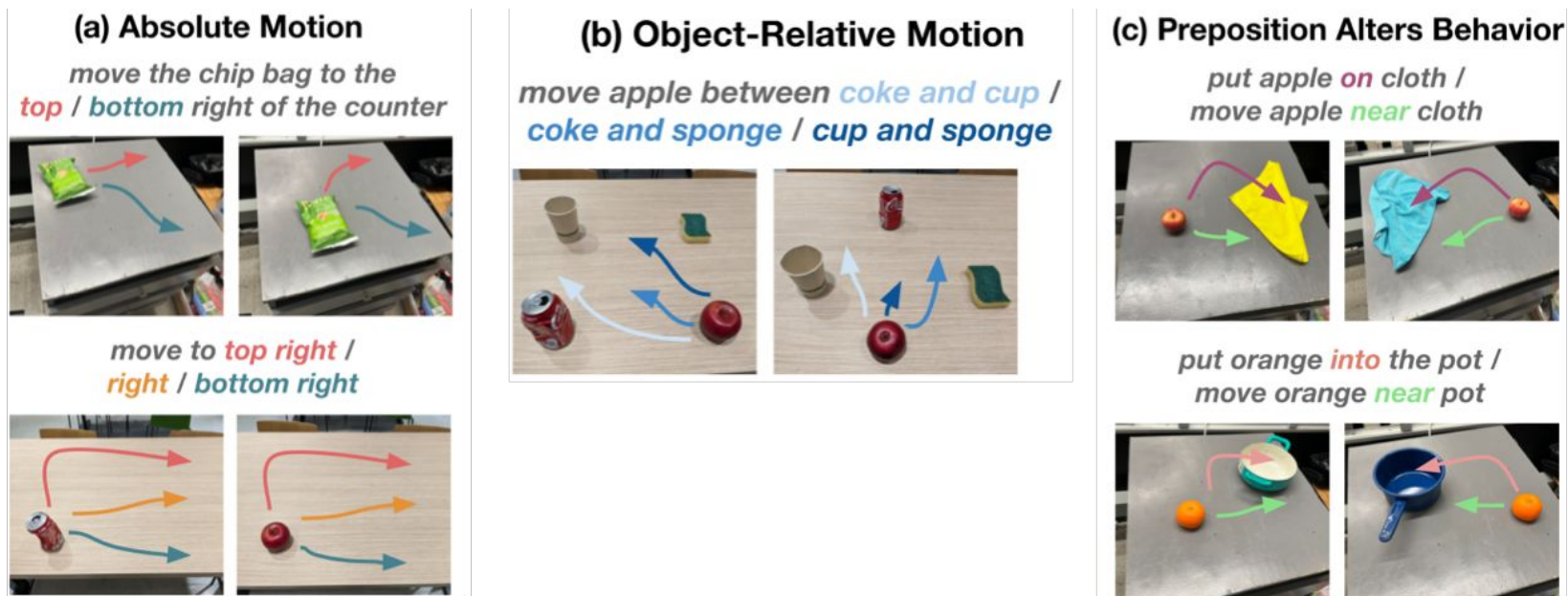
Open-X Embodiment Dataset

- Training RT-1 and RT-2 with Open-X Embodiment Dataset



Open-X Embodiment Dataset

- Emergent Skills (OOD Skills): transfer of skills across robots



Row	Model	Size	History Length	Dataset	Co-Trained w/ Web	Initial Checkpoint	Emergent Skills Evaluation	RT-2 Generalization Evaluation
(1)	RT-2	55B	none	Google Robot action	Yes	Web-pretrained	27.3%	62%
(2)	RT-2-X	55B	none	Robotics data	Yes	Web-pretrained	75.8%	61%
(3)	RT-2-X	55B	none	Robotics data except Bridge	Yes	Web-pretrained	42.8%	54%
(4)	RT-2-X	5B	2	Robotics data	Yes	Web-pretrained	44.4%	52%
(5)	RT-2-X	5B	none	Robotics data	Yes	Web-pretrained	14.5%	30%
(6)	RT-2-X	5B	2	Robotics data	No	From scratch	0%	1%
(7)	RT-2-X	5B	2	Robotics data	No	Web-pretrained	48.7%	47%

Boston Dynamics AI Institute



**Boston
Dynamics**
AI INSTITUTE

*“What we are aiming for is to have AI advance in robots so that it can **be shown a task by a human, learn how its done, do it itself, and then even communicate to other robots how to do that task.**” – Marc Raibert (ICRA’23)*



Boston Dynamics AI Institute

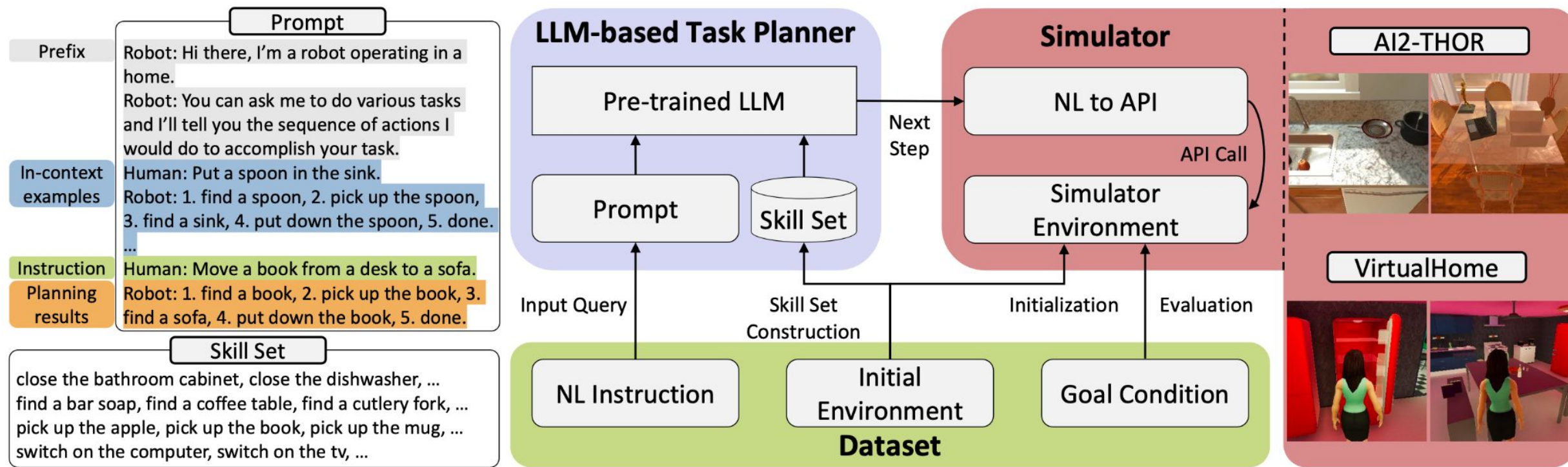
- Watch - Understand - Do



Research@ETRI on Foundation Models for Robotics

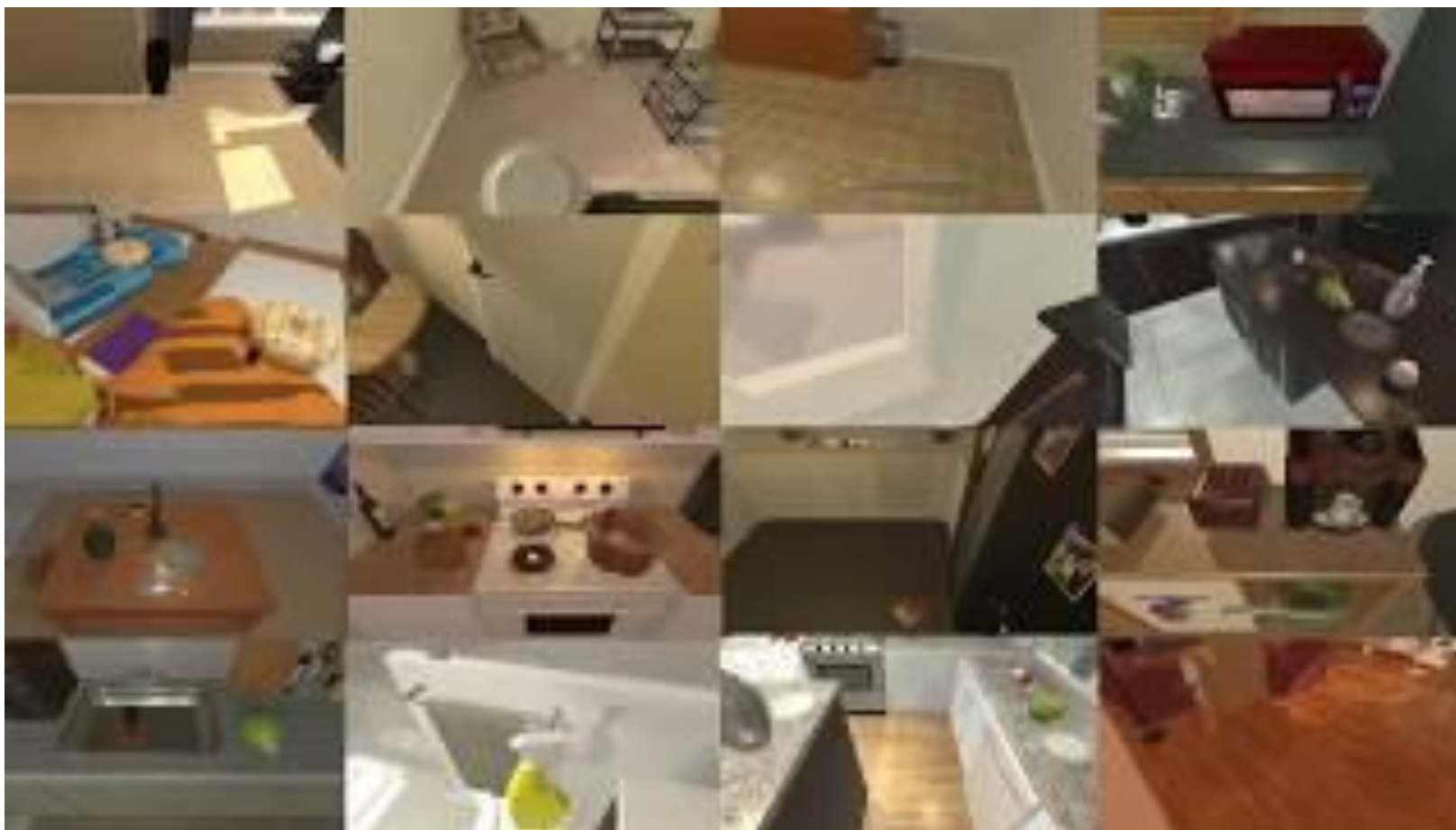
LoTA-Bench: Benchmarking Language-oriented Planners for Embodied Agents

- Automatic evaluation of LLM-based embodied task planners
- Embodied task domains: ALFRED, Watch-And-Help
- Environments: AI2-Thor, VirtualHome
- **No human supervision is needed**



ALFRED

- A Simulation and a dataset for training and testing **Domestic Task Planning**
- AI2Thor simulator





video: <https://www.youtube.com/watch?v=1XoRLNmXffo&t=1s>

ALFRED Tasks

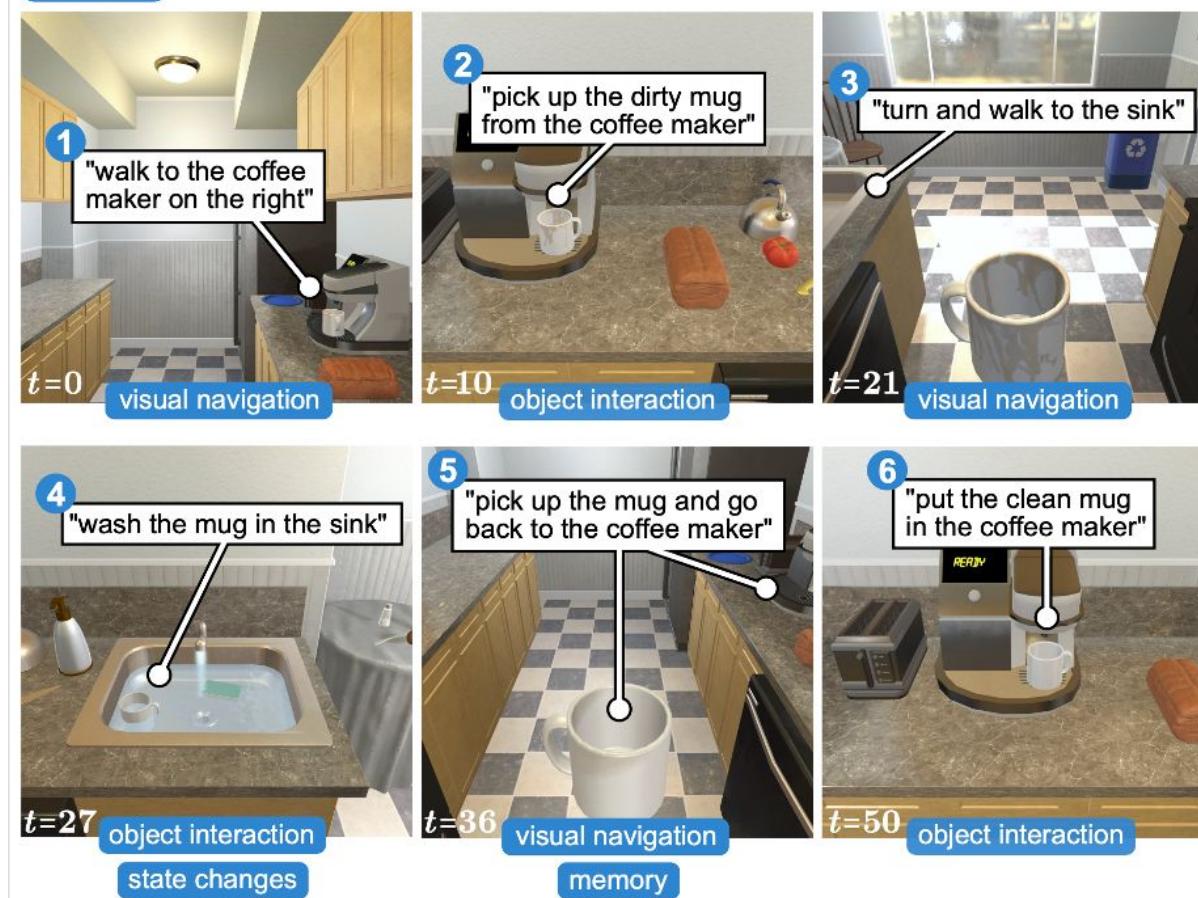
7 Task Types

	Pick & Place	Stack & Place	Pick Two & Place
item(s)	Book	Fork (in) Cup	Spray Bottle
receptacle	Desk	Counter Top	Toilet Tank
scene #	Bedroom 14	Kitchen 10	Bathroom 2
expert demonstration			

Clean & Place	Heat & Place	Cool & Place	Examine in Light
Dish Sponge	Potato Slice	Egg	Credit Card
Cart	Counter Top	Side Table	Desk Lamp
Bathroom 1	Kitchen 8	Kitchen 21	Bedroom 24
			

A Task Sample

Goal: "Rinse off a mug and place it in the coffee maker"



The task sample is a sequence of six steps:

- t=0**: "walk to the coffee maker on the right" (visual navigation)
- t=10**: "pick up the dirty mug from the coffee maker" (object interaction)
- t=21**: "turn and walk to the sink" (visual navigation)
- t=27**: "wash the mug in the sink" (object interaction, state changes)
- t=36**: "pick up the mug and go back to the coffee maker" (visual navigation, memory)
- t=50**: "put the clean mug in the coffee maker" (object interaction)

VirtualHome



video: http://virtual-home.org/images/video_teaser.mp4

VirtualHome Tasks

5 Task Types

Task Type	Goal Condition	Instruction
<i>Setup a dinner table</i>	ON(plate, kitchen table): 1, ON(water glass, kitchen table): 1, ON(wine glass, kitchen table): 1, ON(cutlery fork, kitchen table): 1	put the following on the kitchen table - 1 cutlery fork, 1 wine glass, 1 water glass and one plate
<i>Put groceries</i>	INSIDE(cupcake, fridge): 1, INSIDE(pancake, fridge): 1, INSIDE(pound cake, fridge): 1, INSIDE(apple, fridge): 1	Please get the apple, the pancake, the pound cake and the cupcake and put them all in the fridge.
<i>Prepare a meal</i>	ON(pancake, kitchen table): 1, ON(pudding, kitchen table): 1	Robot, please put the pancake and pudding on the kitchen table.
<i>Wash dishes</i>	INSIDE(plate, dishwasher): 1, INSIDE(wine glass, dishwasher): 1, SWITCHON(dishwasher):1	Place one wine glass and one plate in the dishwasher and turn it on.
<i>Prepare snacks</i>	ON(juice, coffee table): 1, ON(apple, coffee table): 1	Put one cupcake and one apple on the coffee table

Baseline Language-model based Planner

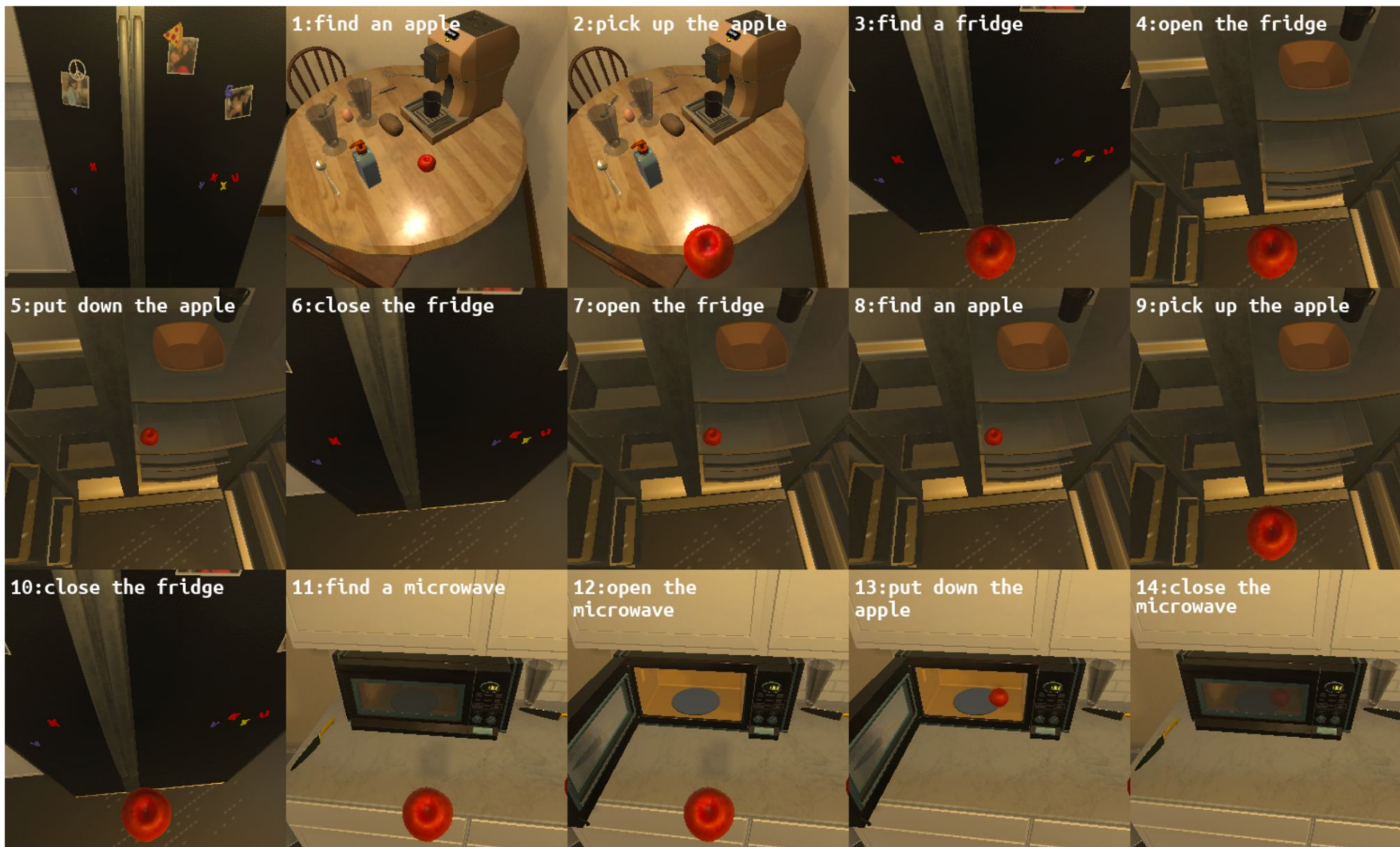
- Given a natural language instruction i , a plan P is constructed via a greedy search based on the skill probability p which is calculated as:

$$p(s|i, s_1, \dots, s_{t-1}) = p_{\text{LLM}}(s|P) = \prod_{n=1}^{n_s} p_{\text{LLM}}(w_n^s | P, w_0^s, \dots, w_{n-1}^s)$$

- s_t : a skill to perform at time t
- *LLM*: a pre-trained large-language model
- A prompt P consists of a prefix, in-context examples, an instruction i , and a history of previously executed skills
- A skill s is described by n_s subword tokens $s = (w_1^s, w_1^s, \dots, w_1^s)$
- w_n^s : n -th subword for a skill s
- $w_0^s = \{\}$

Planning Example

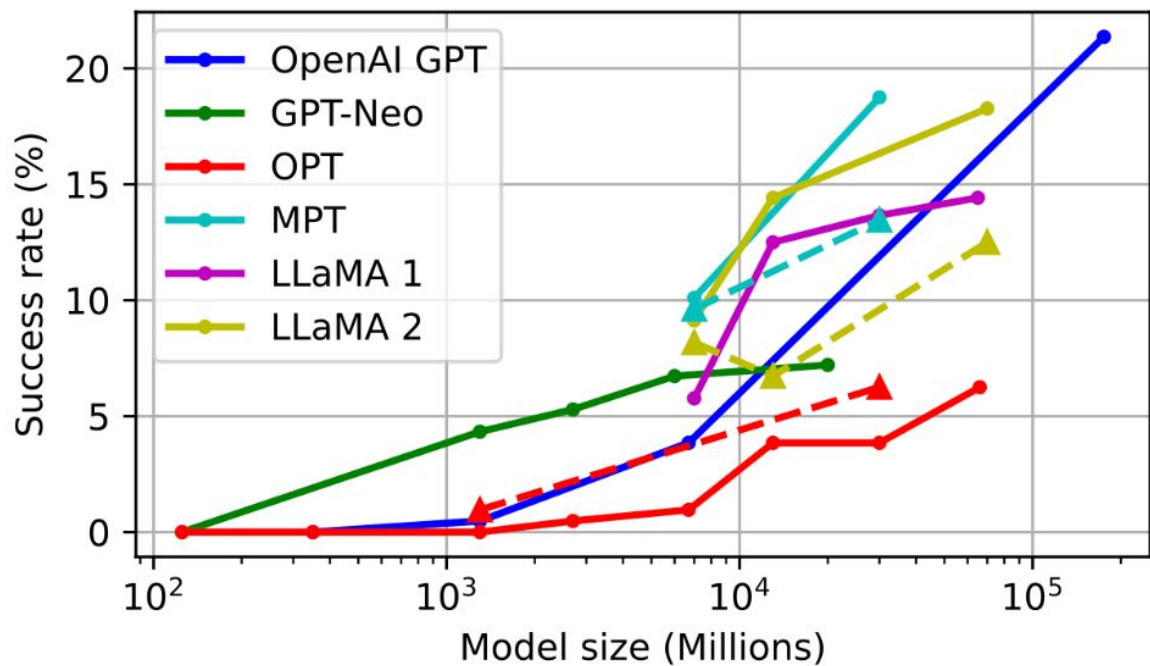
Instruction: Put a chilled apple in the microwave.



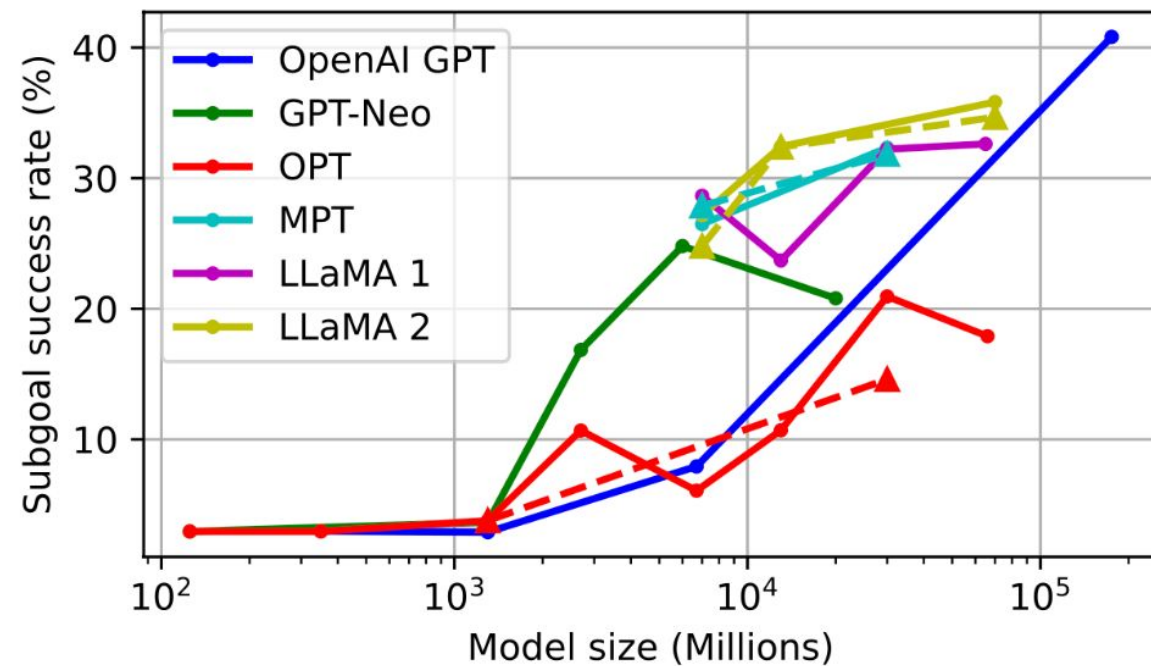
Experiments with many LLMs

Class	Model name	Model size	Remark	Class	Model name	Model size	Remark
OpenAI GPT	ada	350M		MPT	mosaicml/mpt-7b	7B	
	babbage	1.3B			mosaicml/mpt-30b	30B	
	curie	6.7B			mosaicml/mpt-7b-instruct	7B	Instruction-tuned
	text-davinci-003	175B			mosaicml/mpt-30b-instruct	30B	Instruction-tuned
GPT Neo	EleutherAI/gpt-neo-125m	125M		LLaMA 1	huggyllama/llama-7b	7B	
	EleutherAI/gpt-neo-1.3B	1.3B			huggyllama/llama-13b	13B	
	EleutherAI/gpt-neo-2.7B	2.7B			huggyllama/llama-30b	30B	
	EleutherAI/gpt-j-6b	6B			huggyllama/llama-65b	65B	
	EleutherAI/gpt-neox-20b	20B					
OPT	facebook/opt-125m	125M		LLaMA 2	meta-llama/Llama-2-7b-hf	7B	
	facebook/opt-1.3b	1.3B			meta-llama/Llama-2-13b-hf	13B	
	facebook/opt-2.7b	2.7B			meta-llama/Llama-2-70b-hf	70B	
	facebook/opt-6.7b	6.7B			meta-llama/Llama-2-7b-chat-hf	7B	Chat-tuned
	facebook/opt-13b	13B			meta-llama/Llama-2-13b-chat-hf	13B	Chat-tuned
	facebook/opt-30b	30B			meta-llama/Llama-2-70b-chat-hf	70B	Chat-tuned
	facebook/opt-66b	66B					
	facebook/opt-impl-max-1.3b	1.3B	Instruction-tuned				
	facebook/opt-impl-max-30b	30B	Instruction-tuned				

Performance of Baseline Planners

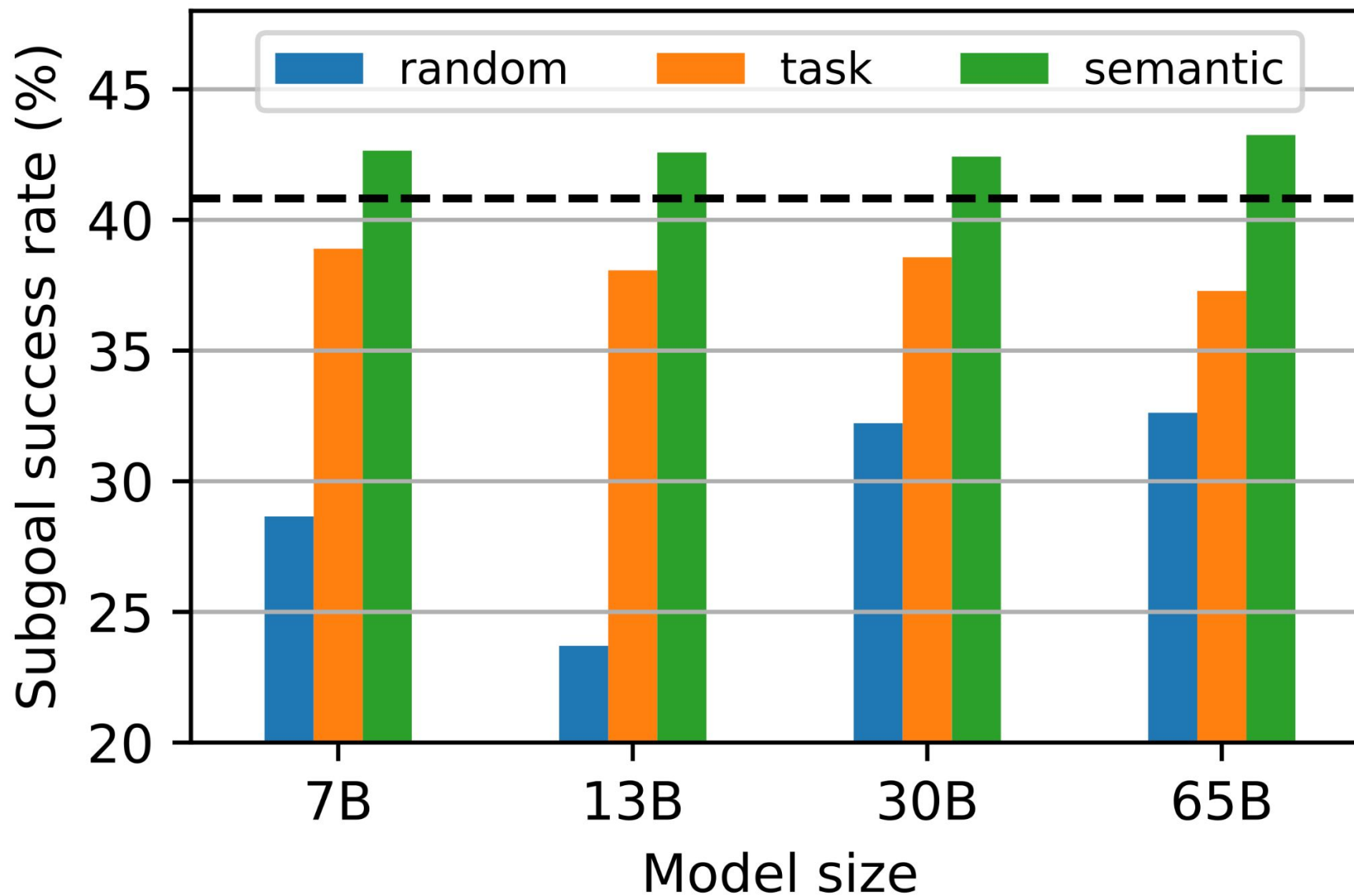


(a) ALFRED

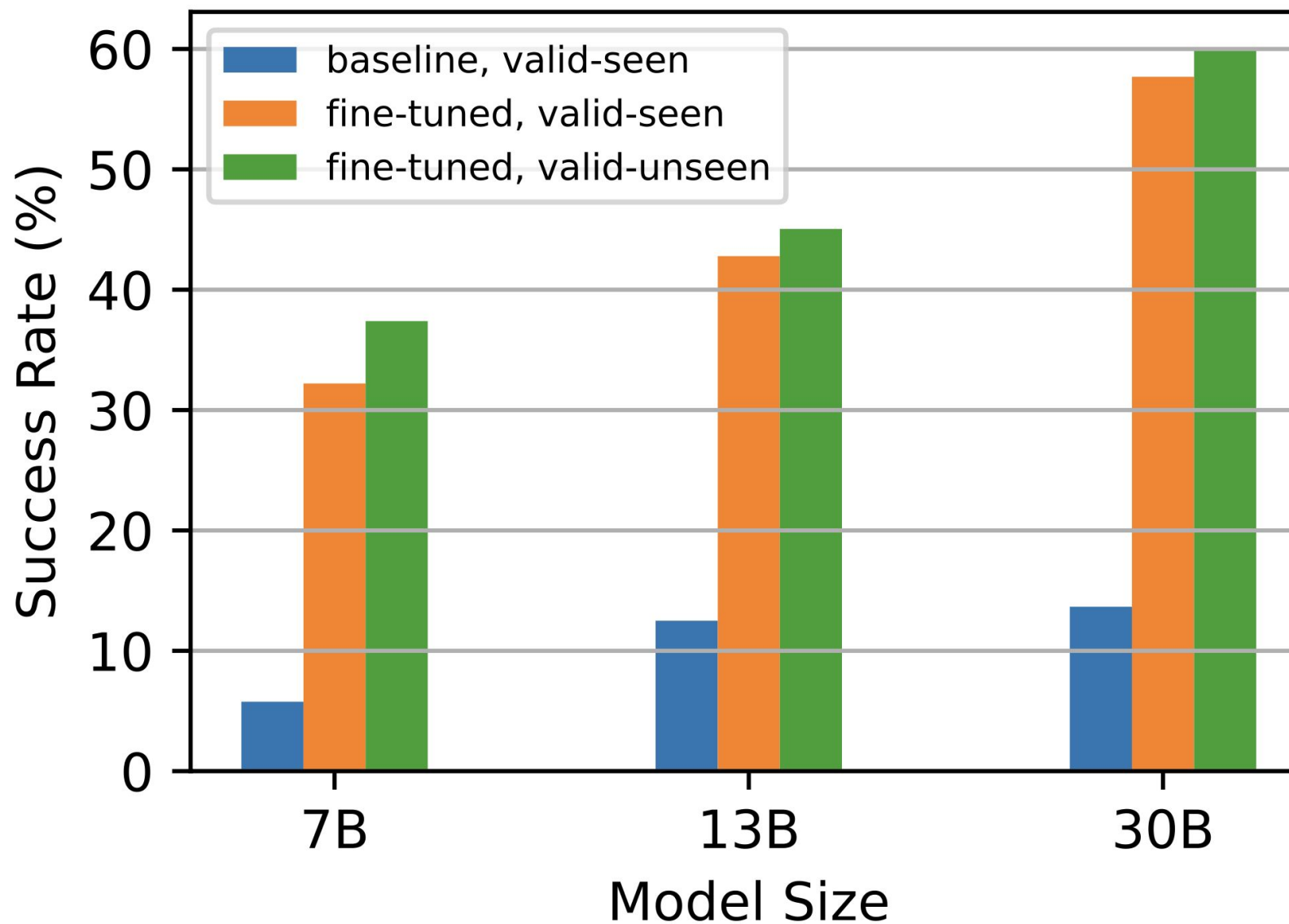


(b) WAH-NL

In-Context Sample Selection (WAH / LLaMA1)



Finetuning (ALFRED / LLaMA1)



Summary

- Foundation models makes it possible to build “Generalist Robots”
 - Reason and Act based on common-sense and embedded knowledge.
 - Watch, hear, read and learn new skills.
 - Teach by language.





감사합니다

ETRI
한국전자통신연구원

National AI Research Institute - Making a Better Tomorrow