



Robotics and embodied interaction

HUANG Yihua, CHANG Chirui





the future can be quite amazing; most products/services will be elevated, and new categories created.



if you value intelligence above all other human qualities, you're gonna have a bad time

Robots that have developed self-awareness are even riskier ...

Outline

- Why Robotics need LLM?
 1. Role of LLM in Robotics
 2. Benefits of LLM for Robotics
- Related Technologies
 1. Perception
 2. Decision-making
 3. Control
 4. Interaction
- Robotics Based on LLMs
- Discussion & QA

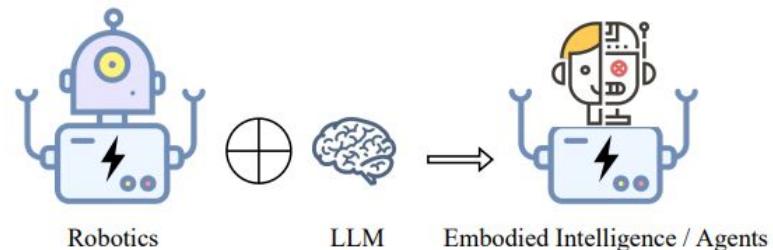
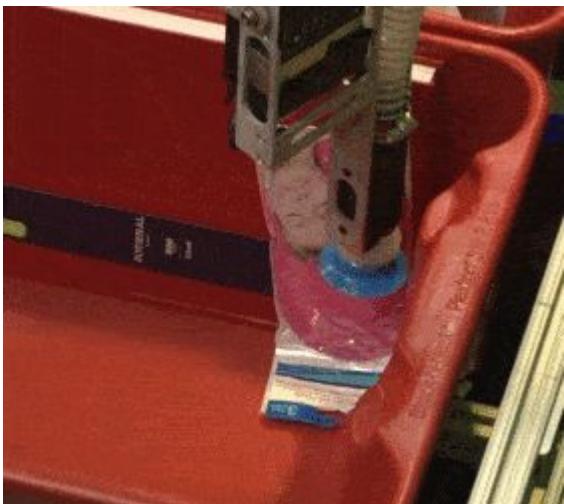


Why Robotics need LLM?

CHANG Chirui

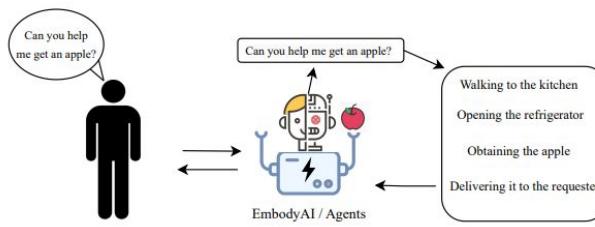
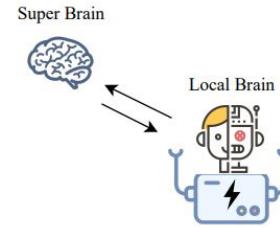
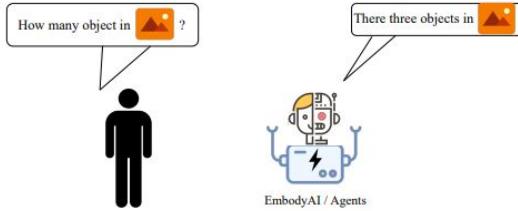
Role of LLM in Robotics

- Robotics w/o LLM V.S. Robotics w/ LLM



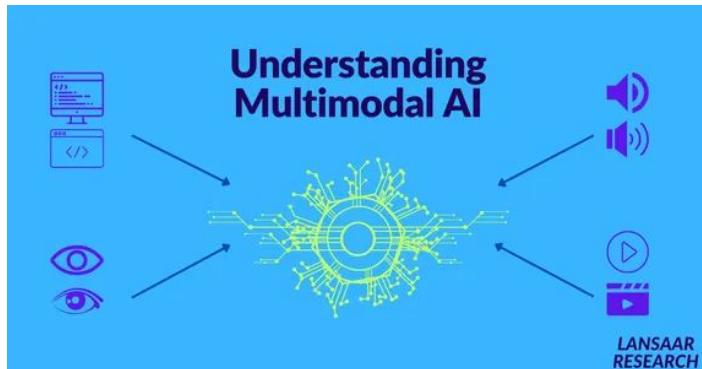
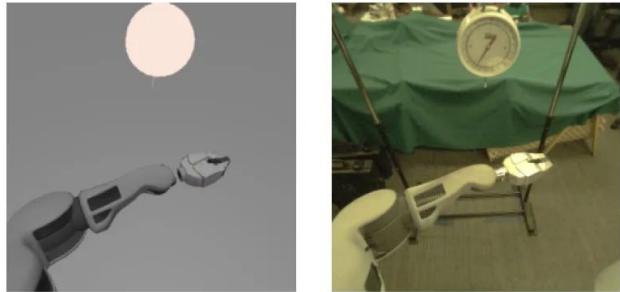
Benefits of LLM in Robotics

- Natural language interaction
- Task execution
- Knowledge acquisition and reasoning



Benefits of LLM in Robotics

- Flexibility and adaptability
- Learning and improvement
- Multimodal interaction



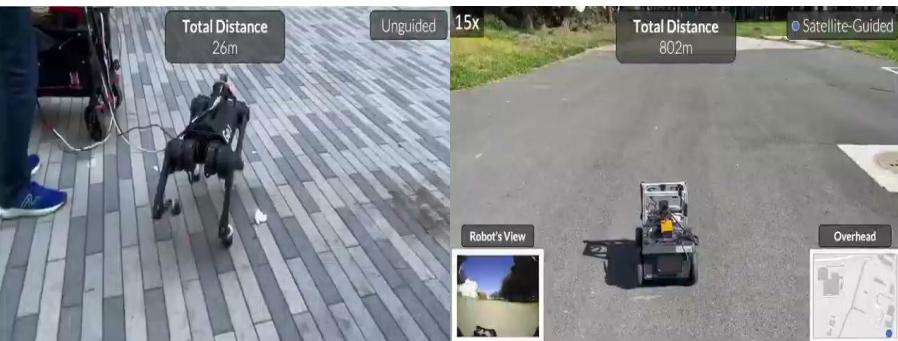


Related Techniques

CHANG Chirui

Perception

- *Vision-navigation model (VNM)*
- *Vision-language model (VLM)*
- *Vision-and-language navigation model (VLN)*
- *Vision-language-action model (VLA)*

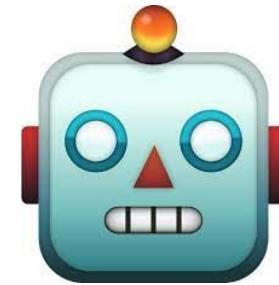


Leave the bedroom, and enter the kitchen. Walk forward, and take a left at the couch. Stop in front of the window.

Interaction

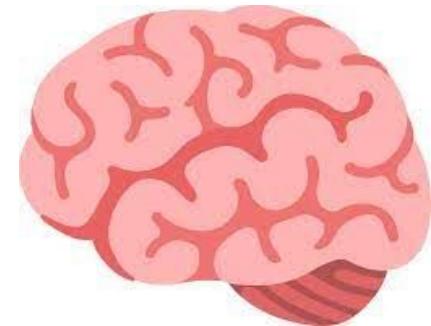
- Interact with both the environment and humans.

- Game
 - Generative agents
- Language-based human-robot interaction
 - GUI (Graphical User Interface) and LUI (Language User Interface)
 - Translate the human instructions into commands that robots can understand



Decision-making

- Goal: to enable robots to make optimal decisions
- Robotics intelligence with LLM
 - Knowledge
 - Pre-trained data: **Prior**
 - Memory: **in-context**
 - Reasoning
 - CoT
 - Planning
 - Break down the execution task into small tasks



Control

- behavior based on the understanding of language
 - **Imitation** learning techniques
 - Offline **reinforcement** learning
- To execute or not to execute, that is the question
 - A value function to evaluate whether the skill is executable or not: A temporal-difference-based (TD) reinforcement learning





Robotics based on LLM

HUANG Yihua, CHANG Chirui



GenSim: Generating Robotic Simulation Tasks via Large Language Models

Task Library is Urgently Needed !

- Real-world interaction data to train general robotic policies (for **imitation learning**) is expensive.
- Existing data generation methods are designed for scene-level diversity (instances/poses) rather than **task-level** diversity (due to **human effort** and **novelty**).
- To enable the **general** capabilities (epically for the **task-level**) for robots, a large generated task library is urgently needed.
- **LLMs** helps task generation for building a task library.

We need more data ...

GENSIM: Generating Robotic Simulation Tasks via Large Language models

GENSIM: Generating Robotic Simulation Tasks via Large Language Models

Can you generate the task "build-car"?



Large Language Model
Task Creator

```
class BuildCar(Task):
    """Construct a simple car structure using blocks and cylinders."""
    ...
    # Add wheels.
    wheel_size = (0.02, 0.02, 0.02)
    wheel_urdf = 'cylinder/cylinder-template.urdf'
    ...
    self.add_goal(
        objs=wheels,
        matches=np.ones((4, 4)),
        ...
        language_goal="For the wheels,
        place a black cylinder on each
        side of the base blocks.")
```

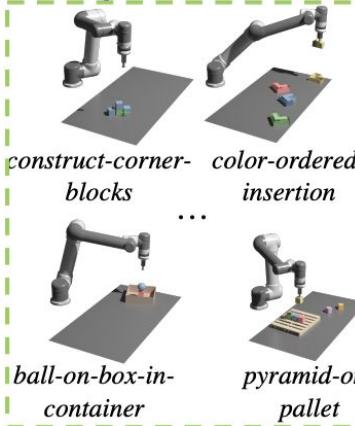
Can you generate a new task
that is different from the
existing ones?

Distillation



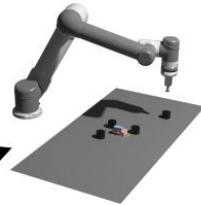
Simulation Engine

Task Library

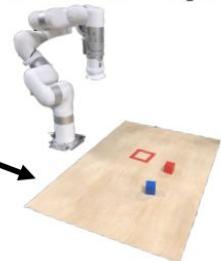


Task-Level Generalization

Reflection

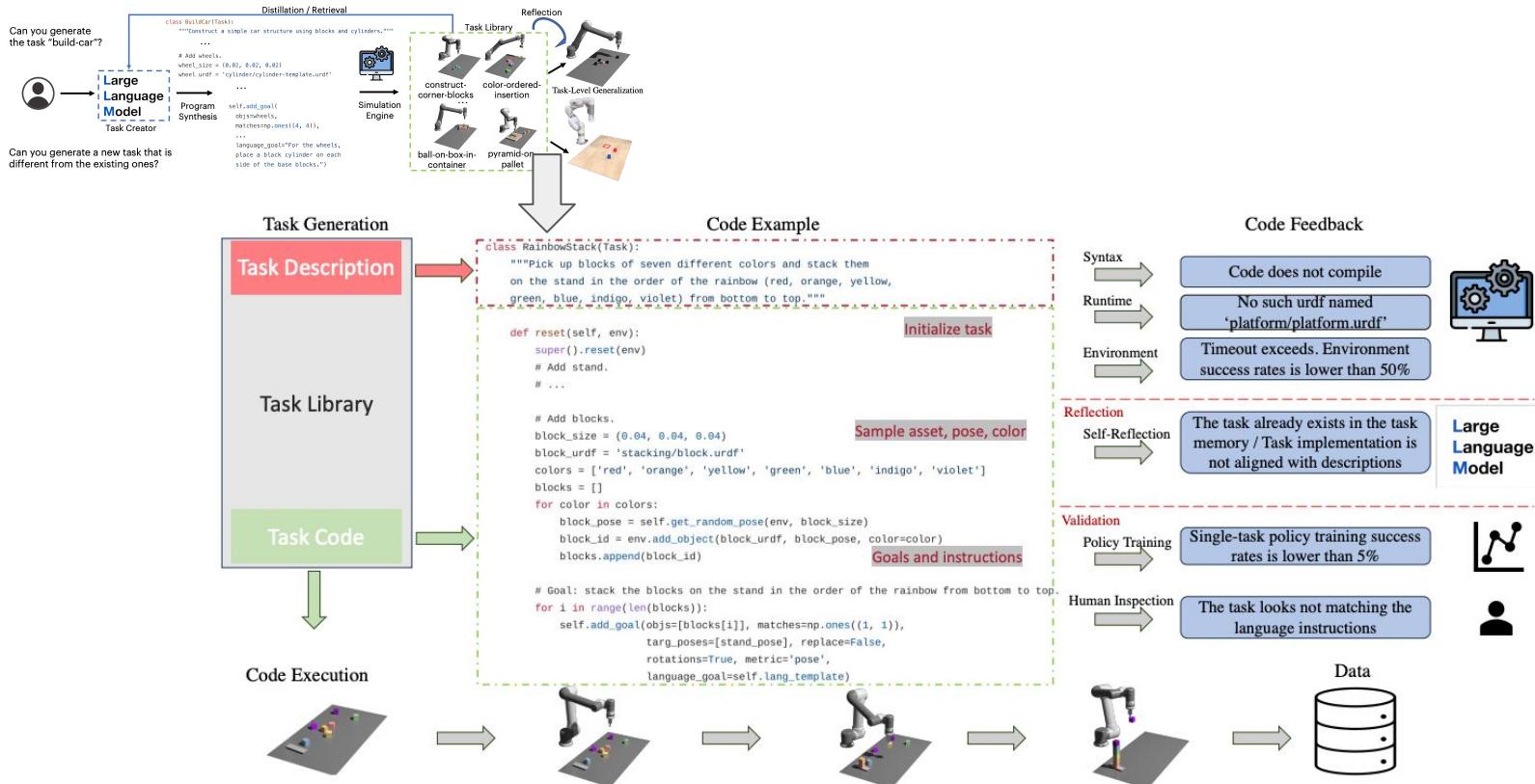


Sim-to-Real Adaptation

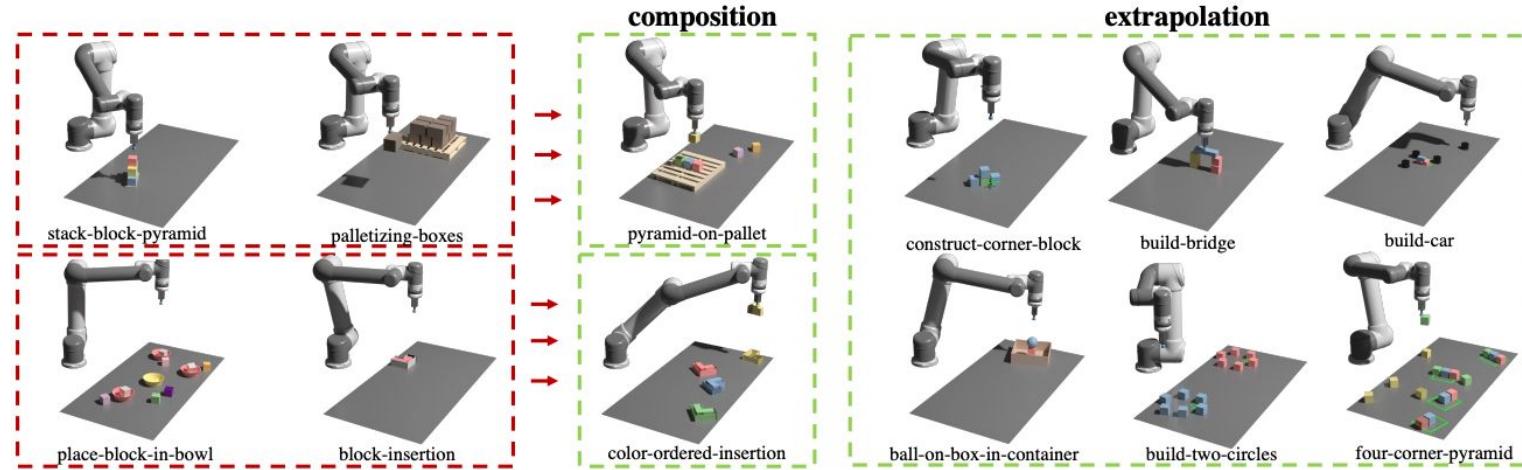


- Input: instruction (mode) Output: code
- prompting mechanisms: goal-directed, exploratory
- task library: instruction code
- language-conditioned multi-task policy training procedure: to enhance task-level generalization

GENSIM: Generating Robotic Simulation Tasks via Large Language models

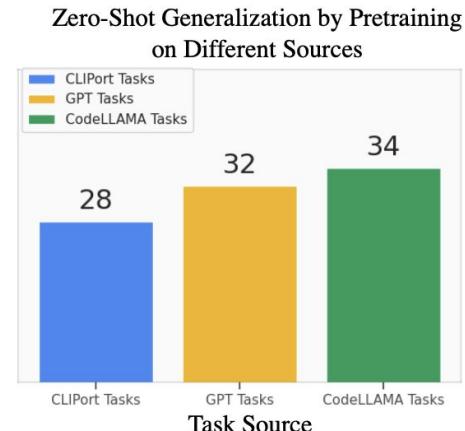
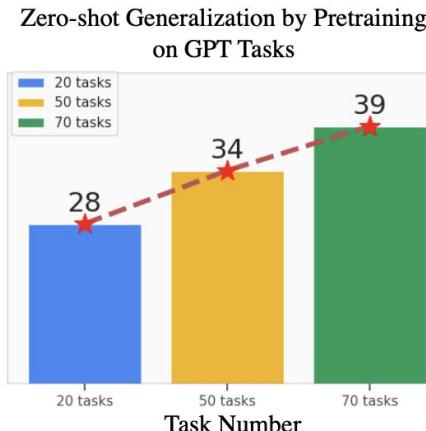
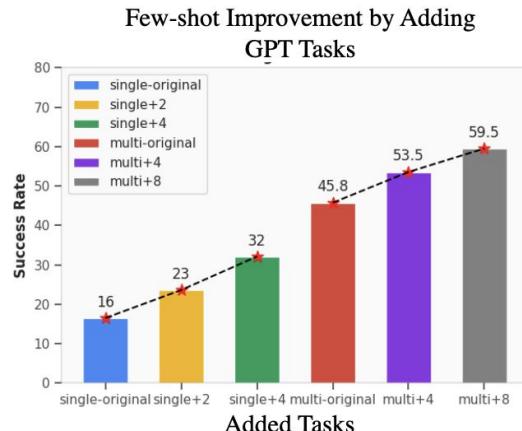
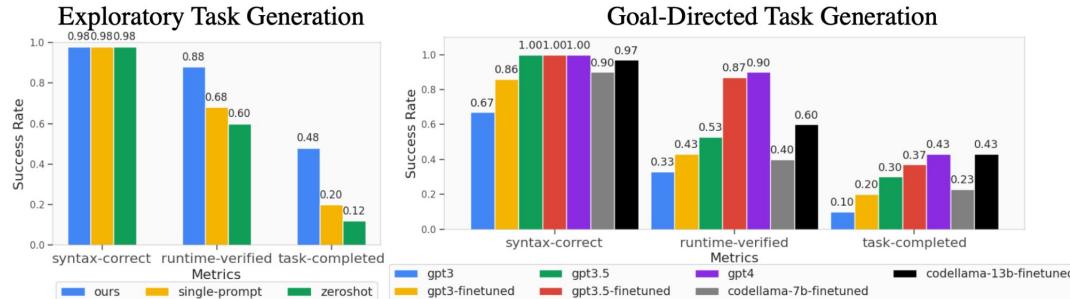


GENSIM: Generating Robotic Simulation Tasks via Large Language models



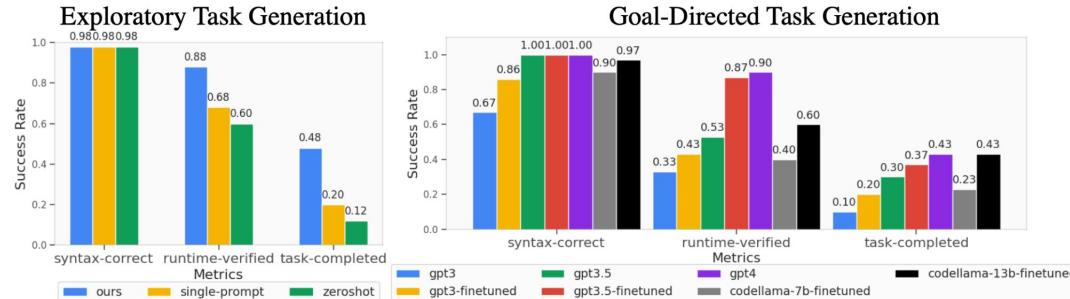
GENSIM: Generating Robotic Simulation Tasks via Large Language models

- How well can LLM design and implement simulation tasks?
- Does LLM generated tasks improve policy generalizations?
- Can pretraining on LLM-generated simulation tasks benefit real-world robot policy deployment?



GENSIM: Generating Robotic Simulation Tasks via Large Language models

- How well can LLM design and implement simulation tasks?
- Does LLM generated tasks improve policy generalizations?
- Can pretraining on LLM-generated simulation tasks benefit real-world robot policy deployment?



Task	Episodes(Sample)	No Adaptation Success	No Pretrain Success	CLIPort Success	GenSim (50 Tasks) Success	GenSim (70 Tasks) Success
put-block-in-bowl	20(20)	0/10	5/10	5/10	8/10	8/10
stack-block	25(25)	0/10	2/10	6/10	9/10	10/10
block-tower-on-corner	20(80)	0/10	0/10	2/10	3/10	4/10
block-in-bowl-in-zone	10(20)	0/10	10/10	7/10	10/10	10/10
put-block-at-zone-corner	10(40)	0/10	0/10	10/10	9/10	10/10
pair-blocks-on-line	20(80)	0/10	0/10	0/10	2/10	4/10
align-block-in-zone	20(60)	0/10	0/10	2/10	3/10	6/10
build-wheel	10(80)	0/10	0/10	0/10	0/10	3/10
pack-spheres	20(80)	0/10	6/10	7/10	4/10	7/10
average	155(485)	0%	25.5%	43.3%	53.3%	68.8%

GENSIM: Generating Robotic Simulation Tasks via Large Language models

- Some interesting reviews:
 - What is done in the cases where the output of the LLM has a syntactical error and cannot be run?
 - **Response:**

One of the properties of coding simulation tasks is that we can just execute the robot simulation and check the results. Therefore, if the output of LLM is non-executable, we will do rejection sampling and move to the next trial of LLM generation. We have added a new Appendix Section D.5 on the feedback in GenSim. Feel free to check that section out.
 - How much prompt engineering went into this?
 - **Response:**

*Creating the task and implementing the task code: Generative v.s. reasoning
The two subtask pipeline is important and this can avoid complicated prompt engineering*

GENSIM: Generating Robotic Simulation Tasks via Large Language models

- Some interesting reviews:
 - Could you provide more details on the types of syntax errors and lack of grounding observed in the generated code? Understanding these specific issues would be valuable for refining the framework. Additionally, how do these errors impact the subsequent policy training and task performance?
 - **Response:**

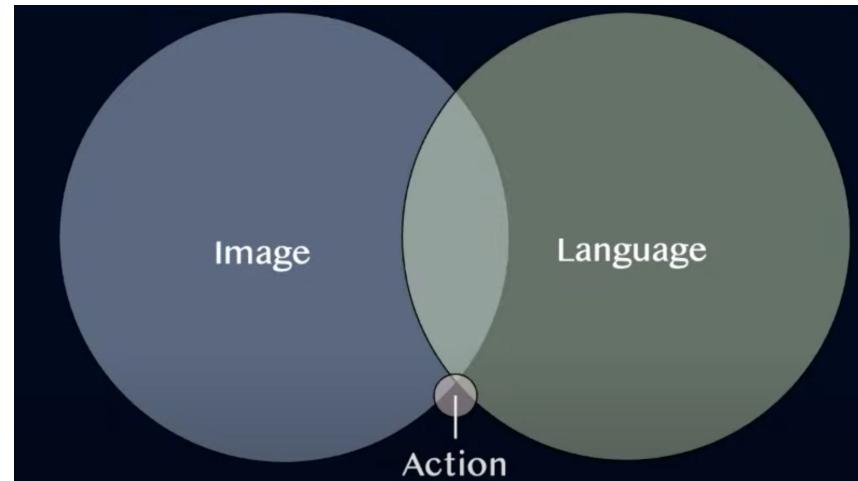
There are low-level mistakes and high-level mistakes. The low mistakes can be simple syntax errors such as “`TypeError: can only concatenate str (not "list") to str`” and “`IndexError: arrays used as indices must be of integer (or boolean) type`” or failures to set the right goal for task completion. The high-level mistakes can be language misalignment and hallucination. Luckily, we have a full automatic mechanism of task filtering such that these tasks will likely be rejected during the task generation process, and therefore will not impact policy training.



Video Language Planning

Hiii bot, what's ur plan?

- Given a long horizon task instruction and current image observation, what's the plan?
- Intelligently interacting with the physical world:
 - semantic abstraction of the task (what to do next)
 - underlying dynamics of the world (what's the current situation)



What we have makes what we can do

- Video language planning (VLP), an algorithm that consists of a tree search procedure, where we train (i) **vision-language models** to serve as both policies and value functions, and (ii) **text-to-video models** as dynamics models.
- Generated video plans can be translated into real robot **actions** via goal-conditioned policies, conditioned on each intermediate frame of the generated video.

Video Language Planning

VLM as Text Policies
image → text actions



PaLM-E

- Action 1. push red star to left ...
- Action 2. move green star to ...
- Action 3. move arm to yellow ...

VLM as Heuristic Functions
image → caption # steps left



PaLM-E

There are 28 steps left to ...

Video Language Planning

Long Horizon Visual Planning with Tree Search



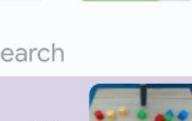
make a horizontal line of blocks



Step 1: push red star to left ... OR Step 1: push blue triangle to ...



Step 2: (re-plan)



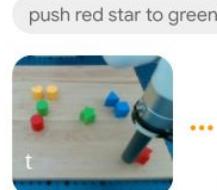
Step 2: (re-plan)

Execute plan from tree search



Video Model as Dynamics Model

text + image → video rollouts of the future



Text-to-Video

Actions from Goal-Conditioned Policies



t current



t goal

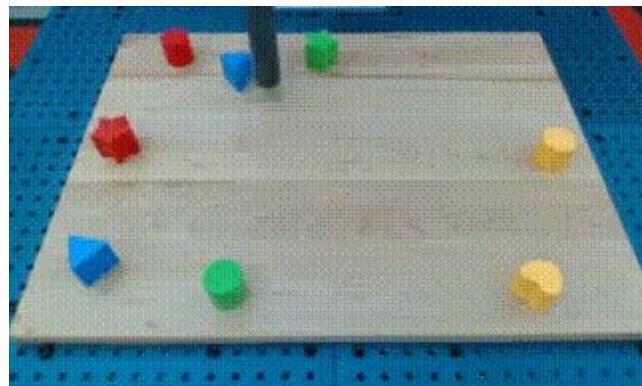
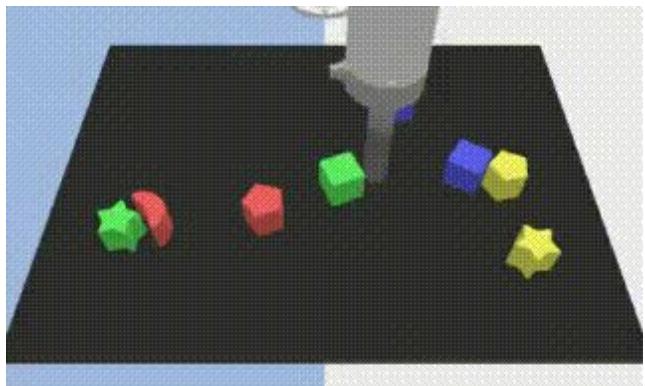
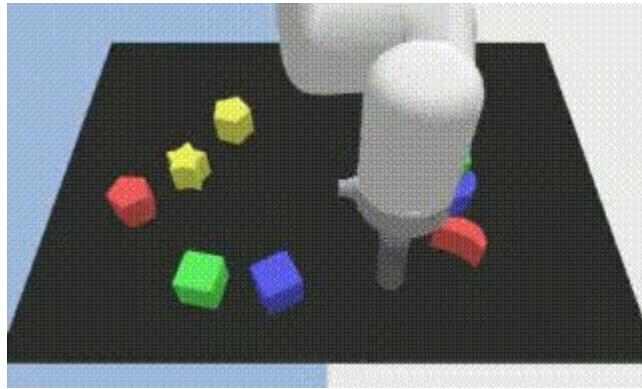
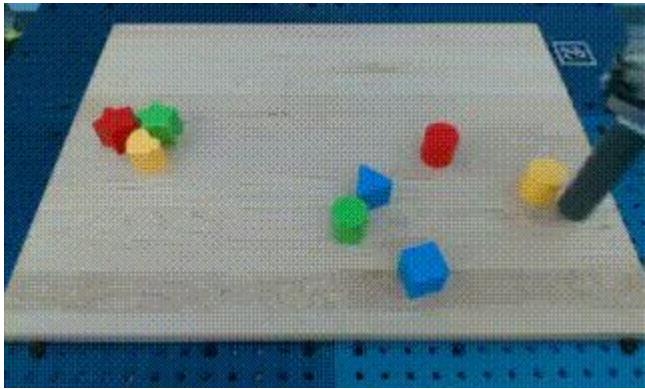
Policies



Experiments across Multiple Robots



Video Language Planning

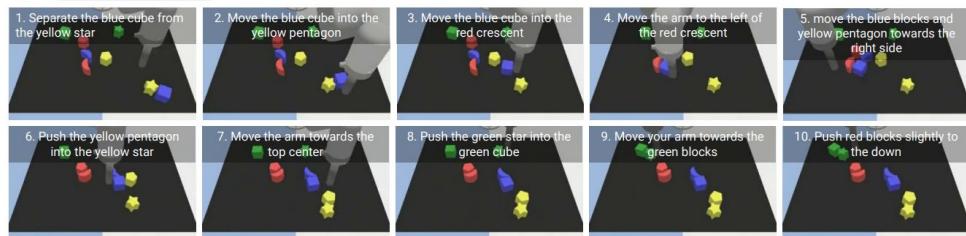


Video Language Planning

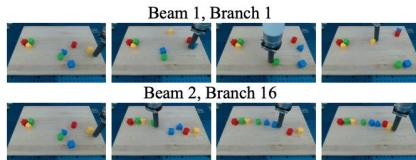
- VLP's ability to synthesize long-horizon video plans for different tasks
- VLP's ability to execute generated video plans in various environments
- VLP's generalization capabilities

Model	Move Area			Group Color		
	Move Area	Group Color	Make Line	Move Area	Group Color	Make Line
UniPi	2%	4%	2%	4%	12%	4%
VLP (No Value Function)	10%	42%	8%	20%	64%	4%
VLP (Ours)	58%	98%	66%	78%	100%	56%

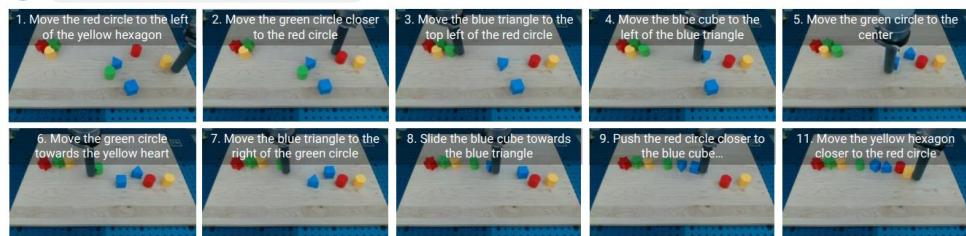
group blocks by color



Beams	Language Branch	Video Branch	Line Performance
	1	1	
1	1	4	4%
1	4	4	10%
2	4	4	22%
			56%



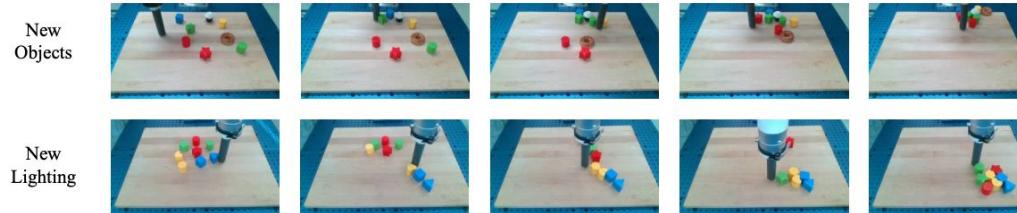
make a horizontal line



Video Language Planning

- VLP's ability to synthesize long-horizon video plans for different tasks
- VLP's ability to execute generated video plans in various environments
- VLP's generalization capabilities

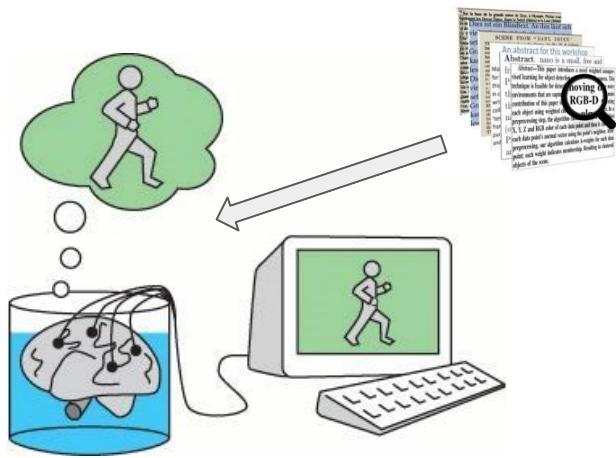
Model	Move to Area		Group by Color		Make Line	
	Reward	Completion	Reward	Completion	Reward	Completion
UniPi (Du et al., 2023b)	30.8	0%	44.0	4%	44.0	4%
LAVA (Lynch et al., 2023)	59.8	22%	50.0	2%	33.5	0%
RT-2 (Brohan et al., 2023)	18.5	0%	46.0	26%	36.5	2%
PALM-E (Driess et al., 2023)	36.5	0%	43.5	2%	26.2	0%
VLP (Ours)	87.3	64%	95.8	92%	65.0	16%



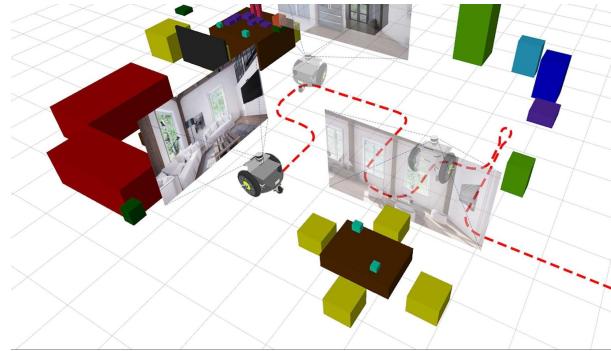


Do As I Can, Not As I Say: Grounding Language in Robotic Affordances

Do As I Can, Not As I Say: Grounding Language in Robotic Affordances

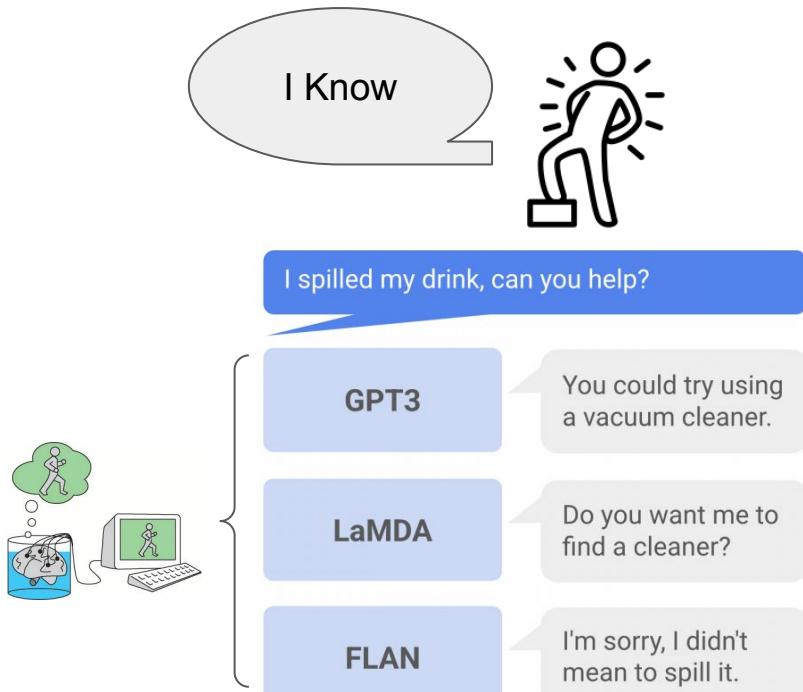


LLM is a brain in a vat

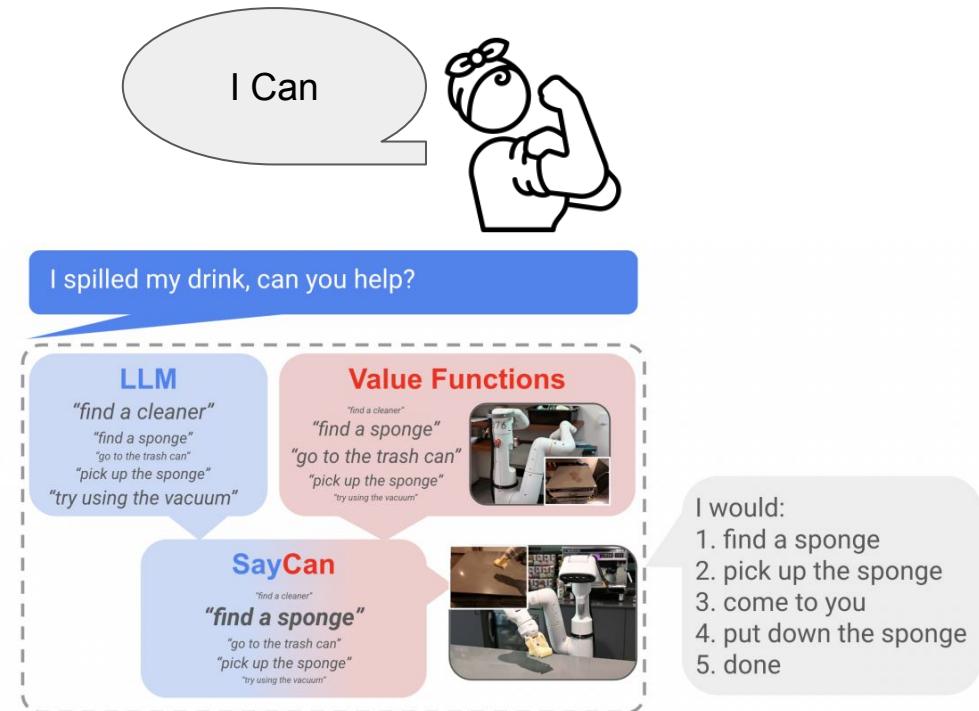


Embodied AI: interaction with
real-world environments

I Can Do v.s I Can Say

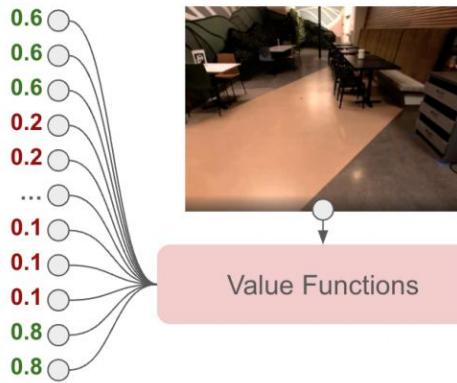


I Can Say

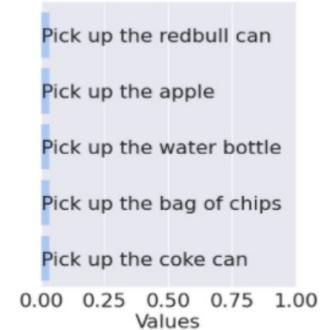
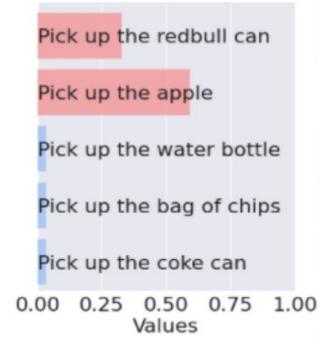


I Can Do

What I Can Do? —Affordance Function



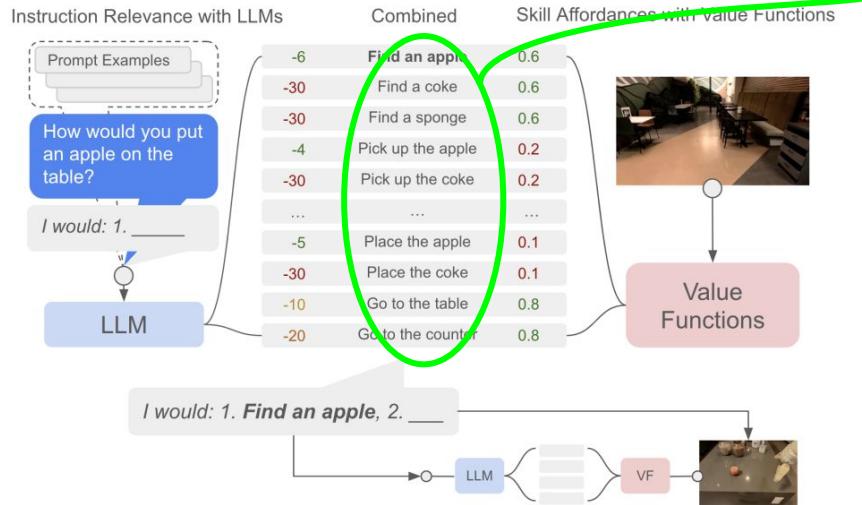
Affordance values evaluated from observations



Grounded LLM with Affordance Values

Implementation Process

1. Generate **action** recommendations using LLM for the next stage.



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:    $C = \emptyset$ 
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:      $C = C \cup p_\pi^{\text{combined}}$ 
9:   end for
10:   $\pi_n = \arg \max_{\pi \in \Pi} C$ 
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

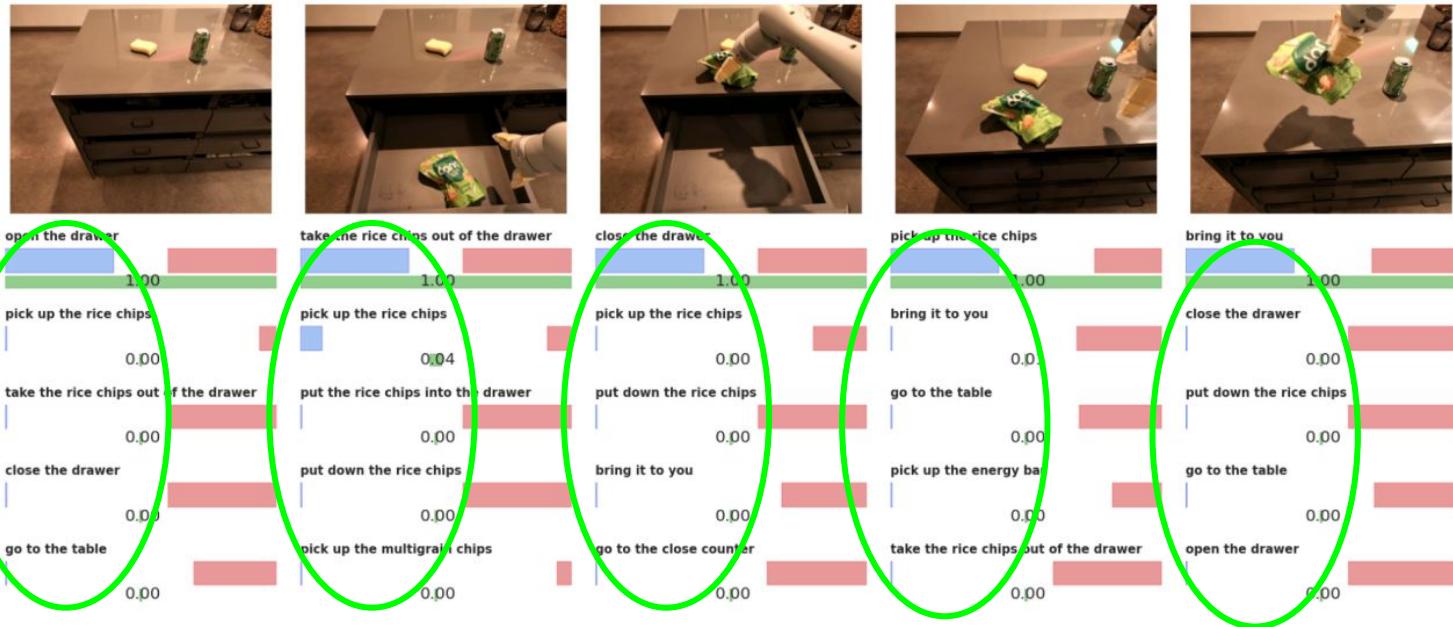
2. Always pick the action with **maximum combined probability** until the task is finished.

Example of Picking Up Chips

Human: Bring me the rice chips from the drawer.

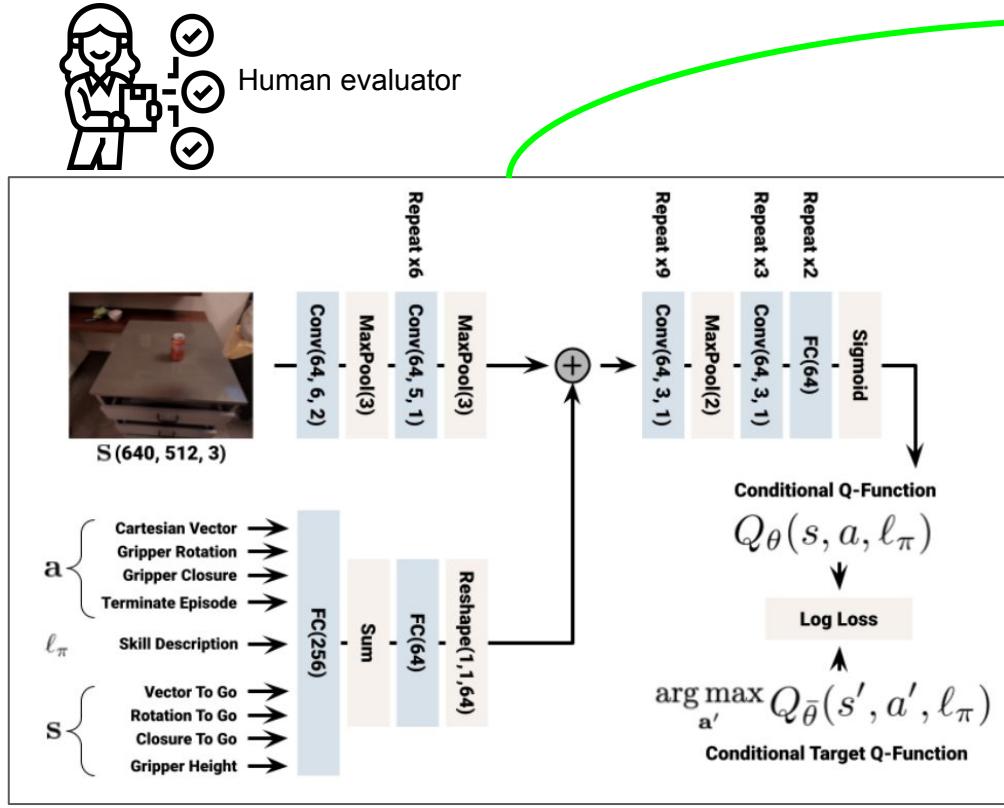
Robot: I would

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



LLM provides sufficiently accurate actions

Model Architecture and Training



Affordance Model trained with Reinforcement Learning (RL)

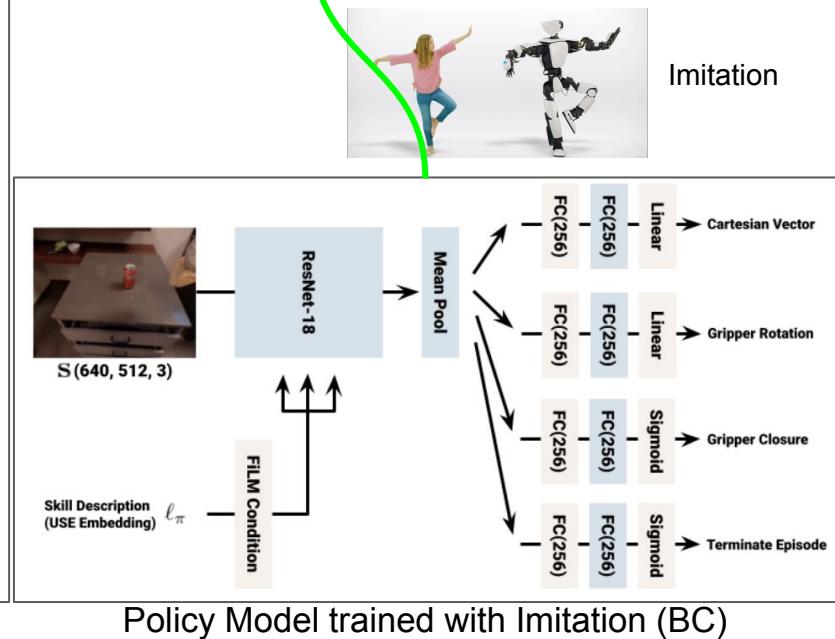
Algorithm 1 SayCan

```

Given: A high level instruction  $i$ , state  $s_0$ , and a set of skills  $\Pi$  and their language descriptions  $\ell_\Pi$ 
1:  $n = 0, \pi = \emptyset$ 
2: while  $\ell_{\pi_{n-1}} \neq \text{"done"}$  do
3:    $C = \emptyset$ 
4:   for  $\pi \in \Pi$  and  $\ell_\pi \in \ell_\Pi$  do
5:      $p_{\pi, n}^{\text{LLM}} = p(\ell_\pi | s_n, \ell_{\pi_0})$ 
6:      $p_{\pi, n}^{\text{affordance}} = p(c_\pi | s_n, \ell_\pi)$ 
7:      $p_\pi^{\text{combined}} = \frac{p_{\pi, n}^{\text{affordance}}}{p_{\pi, n}^{\text{LLM}} + p_{\pi, n}^{\text{affordance}}}$ 
8:      $C = C \cup p_\pi^{\text{combined}}$ 
9:   end for
10:   $\pi_n = \arg \max_{\pi \in \Pi} C$ 
11:  Execute  $\pi_n(s_n)$  in the environment, updating state  $s_{n+1}$ 
12:   $n = n + 1$ 
13: end while

```

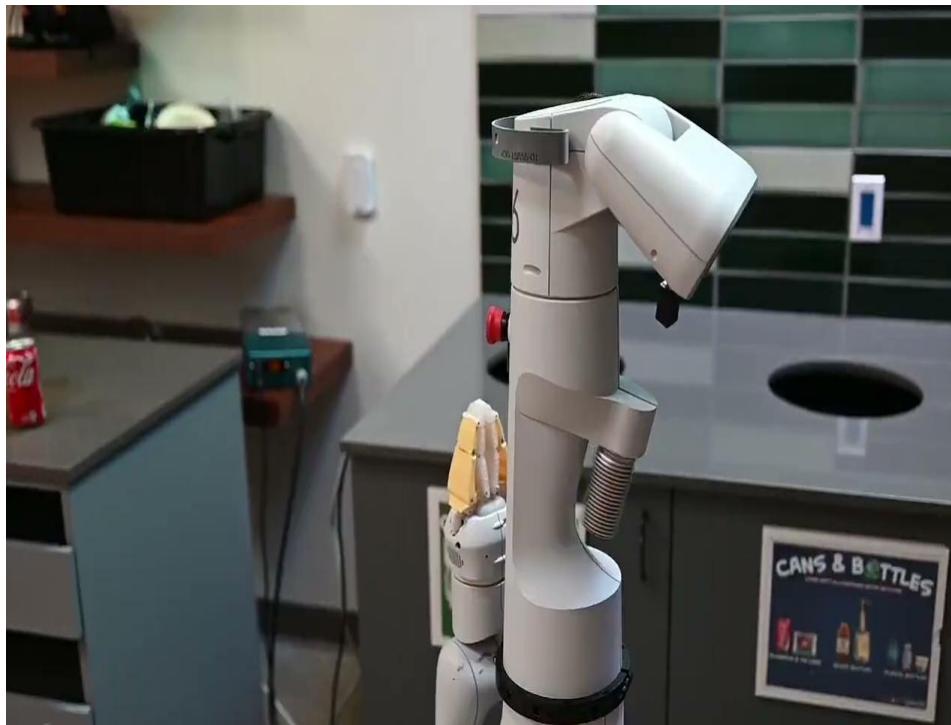
▷ Evaluate scoring of LLM
▷ Evaluate affordance function



Policy Model trained with Imitation (BC)

Imitation

SayCan Demo





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

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

I spilled my drink, can you help?



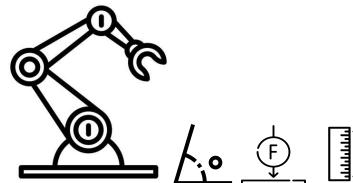
Affordable Policies

High-level Policy

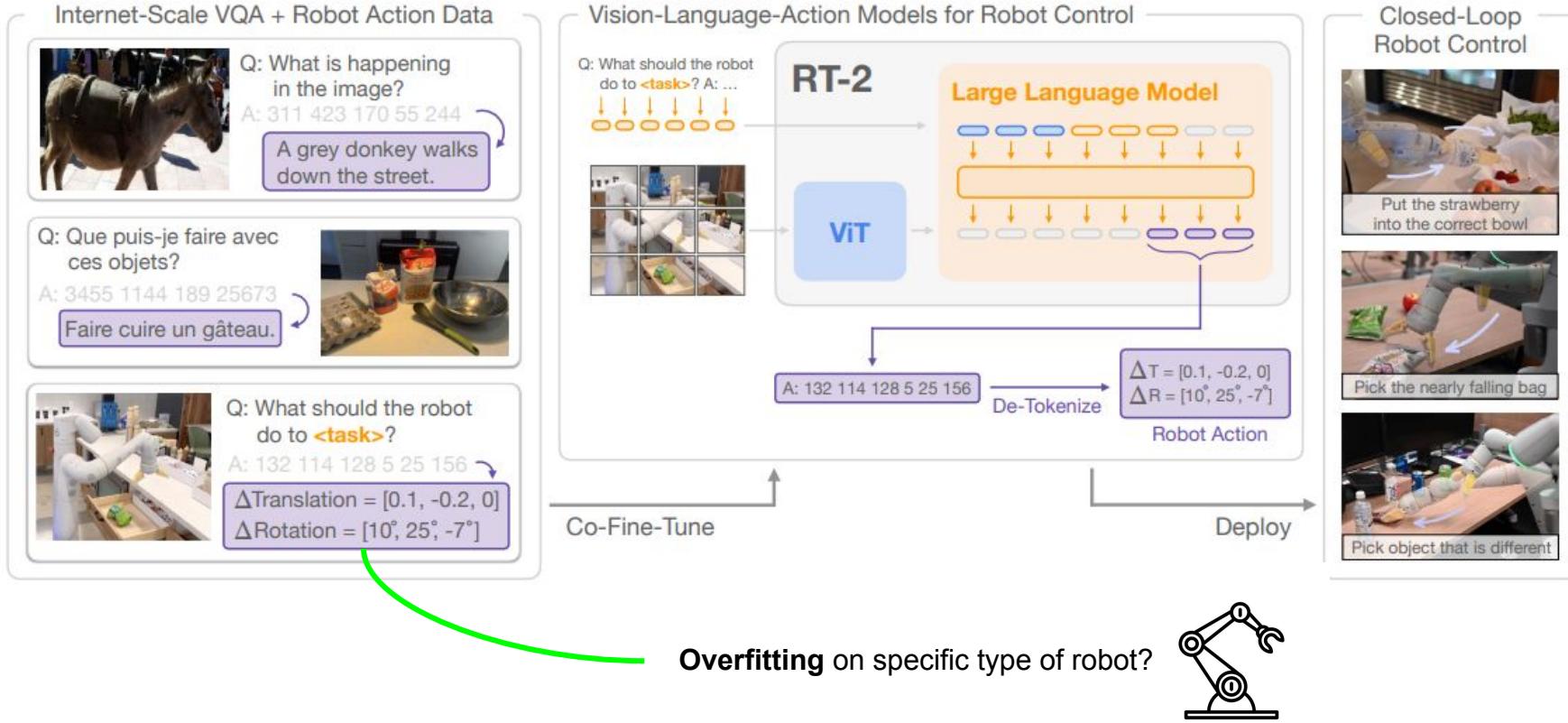
Low-level Robot Action



GAP



Enhance Robot Action with Internet-scale Vision-language Datasets



Generalization to Unseen Situations



put strawberry
into the correct
bowl



pick up the bag
about to fall
off the table



move apple to
Denver Nuggets



pick robot



place orange in
matching bowl



move redbull can
to H



move soccer ball
to basketball



move banana to
Germany



move cup to the
wine bottle



pick animal with
different colour



move coke can to
Taylor Swift



move coke can to
X



move bag to
Google

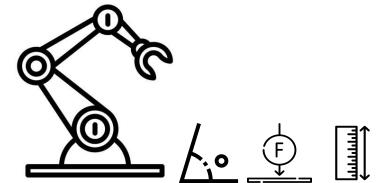


move banana to
the sum of two
plus one



pick land animal

Model



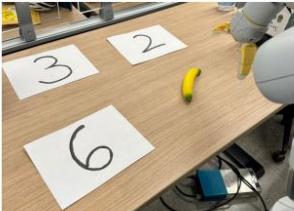
Robot Action
(Parameters)

Generalization to Unseen Situations

(a) Reasoning



"move apple to cup with
same color"



"move banna near the
sum of two plus one"



"déplacer les frites verts
dans la tasse rouge"



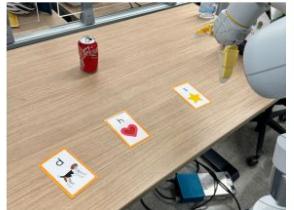
"pick a healthy drink"



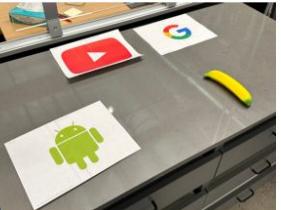
"move coke can to
Taylor Swift"



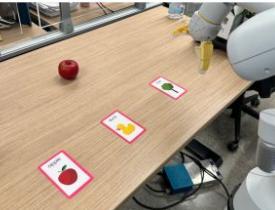
"move coke can
near Y"



"put coke can close
to dog"



"move banana to
android"



"move apple to tree"



"move coke can to
person with glasses"

(b) Symbol Understanding

(c) Human
Recognition

Chain-of-thought Reasoning

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



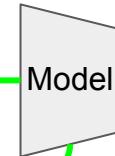
Prompt:
Given Instruction:
Move all the objects together.
Prediction:
Plan: move green can near green rice chip bag.
Action: 1 128 126 127 135
123 119 127



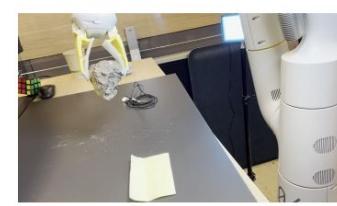
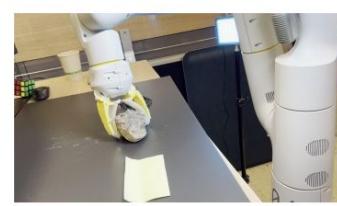
Prompt:
Given Instruction:
Pick the object that is different from all other objects
Prediction:
Plan: pick rxbar chocolate.
Action: 1 128 129 125 126 128 127



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



Chain-of-thought Reasoning



User
Pick up the object that is different from all other objects

Plan: Pick rxbar chocolate



User
Move the green objects together

Plan: Move green can near green rice chip bag



User
Move the blue objects together

Plan: Move blue chip bag near pepsi



User
I am sleepy, bring me a drink to help.

Plan: Pick redbull can





VOYAGER: An Open-Ended Embodied Agent with Large Language Models

MineCraft



Equipment & Inventory



Monsters



Crafting Recipes (Tech)



Health & Hunger

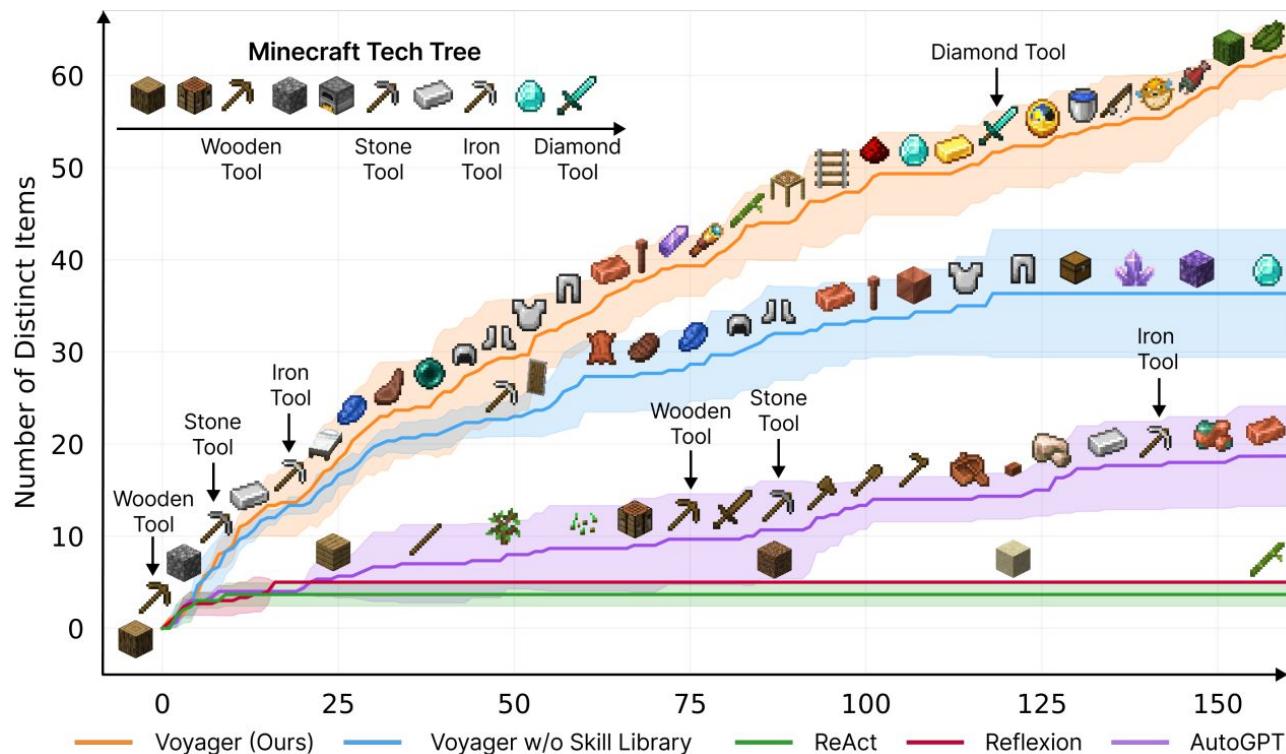


Biome & Terrain



Building

VOYAGER: An Open-Ended Embodied Agent with Large Language Models



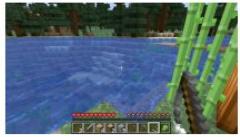
Task Proposal with LLM



Inventory (5/36): {'oak_planks': 3, 'stick': 4, 'crafting_table': 1, 'stone': 3, 'wooden_pickaxe': 1}

GPT-4

Reasoning: Since you have a wooden pickaxe and some stones, it would be beneficial to upgrade your pickaxe to a stone pickaxe for better efficiency.
Task: Craft 1 stone pickaxe.



Biome: river

Inventory (4/36): {'sugar_cane': 3, 'paper': 3, 'fishing_rod': 1, 'wooden_pickaxe': 1}

GPT-4

Reasoning: Since you have a fishing rod in your inventory and you are near a river biome, it would be a good opportunity to catch some fish for food and experience.
Task: Catch 1 fish.



Nearby entities: pig, cat, villager
Health: 12/20
Hunger: 0/20

GPT-4

Reasoning: Your hunger is at 0, which means you need to find food to replenish your hunger. Since there are pigs nearby, you can kill one to obtain raw porkchops.
Task: Kill 1 pig.



Inventory (6/36): {'furnace': 1, 'stone_pickaxe': 1, 'oak_planks': 7, 'crafting_table': 1, 'raw_iron': 4, 'coal': 1}

GPT-4

Reasoning: You have raw iron and coal, and you have a furnace. It's time to smelt the iron to make iron ingots, which can be used to craft better tools and armor.
Task: Smelt 4 raw iron.



Time: night
Nearby entities: zombie
Equipment: 'stone_sword', 'shield'

GPT-4

Reasoning: Since it's night and there's a zombie nearby, it's a good opportunity to try killing the zombie now that you have a stone sword and a shield equipped.
Task: Kill 1 zombie.

Skill Library Creation and Retrieval

Program Generated by GPT-4

```
async function combatZombie(bot) {  
    // Equip a weapon  
    const sword =  
        bot.inventory.findInventoryItem(  
            mcData.itemsByName[  
                "stone_sword"  
            ].id  
        );  
    if (sword) {  
        await bot.equip(sword, "hand");  
    } else {  
        await craftStoneSword(bot);  
    }  
    ...  
    // Craft and equip a shield  
    ...  
    // Recover hunger  
    ...  
    // Look for and combat a zombie  
    ...  
}
```

Program Description

```
async function combatZombie(bot) {  
    // The function is about  
    equipping a stone sword to combat  
    a zombie. If a stone sword is not  
    found, it will craft one.  
    Additionally, it crafts and equips  
    a shield for added protection.  
    Afterwards, it proceeds to cook  
    sticks in order to restore hunger.  
    Once hunger is replenished, it  
    actively searches for a zombie and  
    engages in combat with it.  
}
```

GPT-3.5

Embedding

Key

Value

Skill Library

- Mine Wood Log
- Make Crafting Table
- Craft Wooden Pickaxe
- Craft Stone Sword
- Make Furnace
- ...
- Combat Cow
- Cook Steak
- Craft Iron Axe
- Combat Zombie

Add

Task: Craft Iron Pickaxe

How to craft an iron pickaxe in Minecraft?

GPT-3.5

To craft an iron pickaxe, you
need to 3 iron ingots and 2
sticks. Once you have gathered
the materials,

Environment Feedback

Embedding

Query

Skill Library

Retrieve

Top-5 Relevant Skills

- Smelt Iron Ingot
- Craft Stick
- Make Crafting Table
- Make Furnace
- Craft Wooden Pickaxe

RL with both Environment Feedback and Execution Error

Environment Feedback

I cannot make stick because I need: 2 more planks
I cannot make stone_shovel because I need: 2 more stick

GPT-4

```
async function craftStoneShovelWithTable(bot) {  
    // If not enough cobblestone, mine cobblestone  
    ...  
    + // If not enough sticks, check if there are  
    + // enough planks in the inventory  
    if (sticksCount < 2) {  
        + const planksCount = bot.inventory.count(  
        +     mcData.itemsByName.acacia_planks.id);  
        + if (planksCount < 2) {  
        +     // Collect or craft planks  
        +     await mineBlock(bot, "acacia_log", 1);  
        +     await craftItem(bot, "acacia_planks", 1);  
        + }  
        + // Craft sticks using planks  
        await craftItem(bot, "stick", 1);  
    await craftItem(bot, "stone_shovel", 1);  
}
```

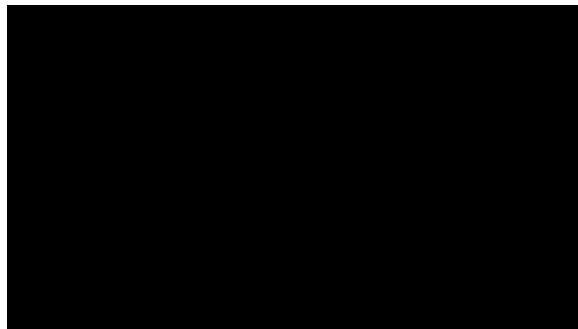
Execution Error

```
throw new Error(`No item named ${name}`);  
No item named acacia_axe  
at line 18:await craftItem(bot, "acacia_axe", 1);
```

GPT-4

```
-async function craftAcaciaAxe(bot) {  
+async function craftWoodenAxe(bot) {  
    // Craft anacia planks and sticks  
    ...  
  
    // Place the crafting table near the bot  
    ...  
  
    - // Craft an acacia axe using 3 acacia planks  
    - // and 2 sticks  
    - await craftItem(bot, "acacia_axe", 1);  
    - bot.chat("Acacia axe crafted.");  
    + // Craft a wooden axe using 3 acacia planks  
    + // and 2 sticks  
    + await craftItem(bot, "wooden_axe", 1);  
    + bot.chat("Wooden axe crafted.");  
}
```

Results





Discussion & QA

HUANG Yihua, CHANG Chirui

Questions

- Is the current capability of large language models sufficient for the general fields of robotics and embodied interaction, with the technological bottleneck lying in the field of robotics?
- If the AGI era comes, will embodied agents be the most beneficial form of AGI to human beings, or the most risky form of existence to human beings?
- How does LLM comprehend real-world concepts rather than solely learning from text datasets? Does it inherently distinguish between knowledge acquired from the real world and that obtained from digital text data?



Reference

HUANG Yihua, CHANG Chirui

1. Ahn, Michael, et al. "Do as i can, not as i say: Grounding language in robotic affordances." arXiv preprint arXiv:2204.01691 (2022).
2. Du, Yilun, et al. "Video Language Planning." arXiv preprint arXiv:2310.10625 (2023).
3. Wang, Lirui, et al. "GenSim: Generating Robotic Simulation Tasks via Large Language Models." arXiv preprint arXiv:2310.01361 (2023).
4. Zeng, Fanlong, et al. "Large Language Models for Robotics: A Survey." arXiv preprint arXiv:2311.07226 (2023).
5. Brohan, Anthony, et al. "Rt-2: Vision-language-action models transfer web knowledge to robotic control." arXiv preprint arXiv:2307.15818 (2023).
6. Wang, Guanzhi, et al. "Voyager: An open-ended embodied agent with large language models." arXiv preprint arXiv:2305.16291 (2023).