### LLM as A Robotic Brain: Unifying Egocentric Memory and Control

### Grounded Decoding: Guiding Text Generation with Grounded Models for Robot Control

[1] Mai J, Chen J, Li B, et al. LLM as A Robotic Brain: Unifying Egocentric Memory and Control[J]. arXiv preprint arXiv:2304.09349, 2023.
[2] Huang W, Xia F, Shah D, et al. Grounded Decoding: Guiding Text Generation with Grounded Models for Robot Control[J]. arXiv preprint arXiv:2303.00855, 2023.

# LLM as A Robotic Brain: Unifying Egocentric Memory and Control

[1] Mai J, Chen J, Li B, et al. LLM as A Robotic Brain: Unifying Egocentric Memory and Control[J]. arXiv preprint arXiv:2304.09349, 2023.
[2] Huang W, Xia F, Shah D, et al. Grounded Decoding: Guiding Text Generation with Grounded Models for Robot Control[J]. arXiv preprint arXiv:2303.00855, 2023.

### Motivation

- Key parts of embodied system: Memory & Control
- LLM as a robotic brain:

A uniform framework for egocentric memory (actions and observations) and control (follow the given instructions).

Integrates multiple multimodal language models by languages.

A zero-shot learning approach.

- Break embodied task into three subtasks:
  - Given a egocentric observation by visual sensor, answer a relating question.
  - Ask questions for details to understand environment comprehensively, and summarizing language descriptions and action.
  - Memorize previous actions and observations, then plan and control

- Break embodied task into three subtasks:
  - Given a egocentric observation by visual sensor, answer a relating question.

Type I. EyeVLM: BLIP2, a VAQ model

(1) Input: image observations and language-based questions.

(2) Ability: express visual data via natural language, according to the questions

(3) **Output**: nature language that answers the questions.

• Ask questions for details to understand environment comprehensively, and summarizing language descriptions and action.

- Break embodied task into three subtasks:
  - Given a egocentric observation by visual sensor, answer a relating question.

Type I. EyeVLM: BLIP2, a VAQ model

(1) Input: image observations and language-based questions.

(2) Ability: express visual data via natural language, according to the questions

(3) **Output**: nature language that answers the questions.

• Ask questions for details to understand environment comprehensively, and summarizing language descriptions and action.

Type II. Nerve LLM: ChatGPT text-davinci-003

(1) Input: robot actions and text observations regarding the questions.

(2) *Ability:* ask multiple contextual questions, summarize the text observations along with the robot action as perception.

(3) **Output**: nature language that describes the egocentric frame and ego-motion.

- Break embodied task into three subtasks:
  - Given a egocentric observation by visual sensor, answer a relating question.
  - Ask questions for details to understand environment comprehensively, and summarizing language descriptions and action.
  - Memorize previous actions and observations, then plan and control

#### Type III. Brain LLM: ChatGPT text-davinci-003

(1) Input: perception stream and possible human instructions in text.
 (2) Ability: memorize past perception, plan, and control the robot.
 (3) Output: nature language describing the action that the robot needs to take; nature language answering questions grounded in the explored environment;

#### **Role Initialization:**

initial prompts to list the detailed instructions, help they understand their roles.

#### **Eye-Nerve Perception:**

Bridge VLM and LLM by iterative QA, the NERVE ask and summarize the dialogue to get a detailed language description

#### **Nerve-Brain Collaboration:**

Accept language information & action from Nerve

#### **Brain reasoning and control:**

choose action from 'move forward', 'turn left', 'turn right', 'stop' **Brain-Human Interaction:** 



Figure 1: The structure of LLM-Brain robot.

Table 1: The specifications for the prompt roles, which are employed in the embodied AI experiment. We use these system prompts to initialize each agent at first. All agents can understand their roles in LLM-Brain, and cooperate with each other to finish the embodied AI task.

Туре	System Instructions
Type I. Eye (to Nerve)	I take a picture of a room in the house. Ask me questions about the content of this image. Carefully asking me informative questions to maximize your information about this room from my image content. I need to navigate in this house completely relying on your text narration for the image later, so you are encouraged to ask for more information about the positions and relative positions of the objects in this room.
Type II. Nerve (to Eye):	Answer the given questions. If you are not sure about the answer, say you don't know honestly. Don't imagine any contents that are not in the image.
Type III. Brain (to Nerve):	Now summarize the information you get about this room space in a few sentences.
Human (to Brain):	Example 1: Explore this house as much as you can. Example 2: After your exploration, answer me some questions about this house.

### Experiments—Active Exploration in Habitat

Give instruction

 Explore this house as completely as possible' (prompt) to the brain before exploration



(a) Observation in text: This is a hallway with a window on the right side. There is no furniture in the room. There is <u>a door in</u> <u>the middle</u> of the room. The size and shape of the room are unknown. Next action : move forward



(b) Observation in text: This is a small room with yellow walls and wooden flooring. <u>There</u> is a door on the left side of the room and a lamp on the right side. There is a wreath hanging on the wall. Next action: turn left



(c) Observation in text: The room is square-shaped and has a large window and a mirror. The predominant colors in the room are white and black. There is a square-shaped table and a decorative chair in front of the window. Next action: turn right



(d) Observation in text: This is an indoor room with a hallway, a rectangular fireplace, and a door. The walls of the room are white and the overall lighting in the room is bright. There is no information about any other objects present in the room. Next action: turn left

Figure 2: The navigation decision based on observation. One agent describes the scene, one agent summarizes the description, and one agent selects an action to take based on the summary. We show the good turns (a,b) and also the bad turns (c,d) based on textual observation.

### Experiments—Embodied Question Answering

不能区分来自

不同时间步的

相同实例,计

Concept Ouestion: Does this house have a TV? Answer: Yes, this house has a TV. It is located in the living room , on the left side of the room. Question: What is the color of the bed? Answer: The bed in the small circular bedroom at place 1 has a red bedspread. The color of the bed frame itself is not mentioned. Counting Question: How many living rooms does this house have? Answer: This house has 4 living rooms. Ouestion: What rooms are there in the house? Answer: A small circular bedroom. A kitchen. A living room with an orange leather couch and a square table. A large room with glass doors. A living room with a dark-colored couch and a square table . A living room with an orange leather couch and a television . A combined living room and kitchen area A living room with a purple couch, a square table with a laptop, and a desk with a laptop. Relation Question: Is the glass door closer to the chairs or the bed? 数问题性能差。 Answer: The glass doors would be closer to the chairs in the same room than the bed in the small circular room. Ouestion: What is in front of the TV? Answer: There is a square table with a white lamp on top. There is also an orange leather couch on the right side of the room. But it's no clear that the couch is directly facing the TV. Open Question: Give me a short plan to relax. Answer: Start by brewing a cup of tea or coffee in the, using the available appliances. 1.Head to the living room with the dark-colored couch, and take a seat on the couch to enjoy your drink, 2. After finishing your drink, move to the living room with the television. Sit on the orange leather couch and watch your favorite TV show or movie. 3. If you prefer some quiet time, head to the living room at place 8 with the purple couch. Sit down with the laptop on the square table or the desk and browse the internet, read articles, or watch videos. 4. To wind down for the night, move to the small circular bedroom at place. Turn off the laptop on the desk, and get into the bed with the red bedspread for a comfortable night's sleep. (c) Question Answering

Figure 4: Qualitative results on experiments of embodied question answering.

### Grounded Decoding: Guiding Text Generation with Grounded Models for Robot Control

[1] Mai J, Chen J, Li B, et al. LLM as A Robotic Brain: Unifying Egocentric Memory and Control[J]. arXiv preprint arXiv:2304.09349, 2023.
[2] Huang W, Xia F, Shah D, et al. Grounded Decoding: Guiding Text Generation with Grounded Models for Robot Control[J]. arXiv preprint arXiv:2303.00855, 2023.

### Motivation

- · 在具身场景下应用LLM:
  - · 缺乏物理环境理解
  - 不能处理非语言观察
  - 忽略奖励、安全限制
- •从交互中学习到的language-condition机器人策略可以提供necessary grounding
  - ·没有足够数据学习,所以缺乏high-level的语义理解
- Grounded Decoding各取所长:
  - Achieve probabilistic filtering: decode an action sequence that both has high probability under the language model and a set of grounded model objectives.

# Compared with SayCan

- SayCan:
  - use a LLM and a value function to select robotic skills among a constrained set of primitives
  - 'Scoring mode' to get the probability of a skill being useful to a high-level instruction
  - only considers a fixed and enumerated set of primitives.
  - only considers grounding functions derived from RL-trained value functions for affordance founding functions
- Grounded Decoding:
  - jointly decodes the LLM and the grounded model at the token level
  - Explores many types of grounding functions to propose a broad family of algorithms.

### Method—Formulation

- LLM predicts p(W) of a text sequence W,  $W = w_{1:N} = (w_1, ..., w_N)$ 
  - $p(W) = \prod_{n=1}^{N} p_{LLM}(w_n | w_{1:n-1})$ , where  $p_{LLM}$  is predicted by a transformer
- GM models probabilities relating to the robot embodiment and environment
  - Model a probability of tokens  $p_G(w_{1:n}|s)$ , given state s
- Problem formulation:
  - Given an instruction in language  $l, w_{1:N}^* = \arg \max_{w_{1:N}, w_n \in \mathcal{W}} p_{GD}(w_{1:N}|s, l)$
- Assumptions:
  - *s* and *l* are conditionally independent, given  $w_{1:N}$ .
  - $p_{GD}(w_{1:N}|s,l) \propto \prod_{n=1}^{N} p_{LLM}(w_n|w_{1:n-1},l) p_G(w_{1:n}|s)$

Algorithm 1 GD w/ Greedy Search

1: Given: state s, instruction  $\ell$ , terminal set  $\mathcal{W}_{term}$ 2: Initialize:  $w = \{\}, n = 0$ 3: while  $w_n \notin \mathcal{W}_{term}$  do 4: n = n + 15:  $w_n = \arg \max_{w_n \in \mathcal{W}} p_{LLM}(w_n | w_{1:n-1}, \ell) p_G(w_{1:n} | s)$ 6: Return: w



**Figure 2:** Overview of **Grounded Decoding**. Given a *free-form* language instruction, a **language model** and **grounded models** jointly decide the next candidate token to be decoded by **combining** their respective likelihood. Language model proposes likely tokens that produce goal-directed and coherent long-horizon behaviors, while grounded models connect them to the physical scene, through a flexible composition of multiple objective functions, such as affordance, preferences, and safety, that can be independently obtained.

# Grounding Function

- Affordance Grounding Function (AF):
  - Determine what is possible in the scene.
  - Given a scene image and instruction, AF is the output of CLIPort primitive policy
- Safety Grounding Function (S):
  - A simple indicator to prevent the robot from interacting hazardous objects
- Preference Grounding Function (P):
  - A similar way with S (soft)

# Experiments

Control: a pretrained multitask language-conditioned policy, CLIPort

LLM: InstructGPT

GD generates grounded freeform actions, without the need of mapping the action to a repertoire of skills in SayCan

Grounding functions can be composed by:

 $p_G = \prod_{i=1}^n p_1$ 

			100% -							
Tabletop Rearrangement (Sim) User		Step 1: Pick up <u>Y</u> and place it on left side Step 2: Pick up O and place it on right of Y Step 3: Done								
	Spell as much of "you" as you can.		40% 20% - 0% -		L U me	о. <sup>ч</sup>				
	MiniGrid 2D Maze (Sim) User	Step 1: Go to the purple door and open it Step 2: Go to the green door and open it Step 3: Go to the <b>purple</b> door and open it Step 4: Go to the green door and open it	100% 80% 60%							
	Traverse the rooms to get to the goal.	Step 5: Go to the red door and open it Step 6: Go to the goal	20% 0%	20% - 0% - purple green	red oran yellow					
	<b>Kitchen Mobile Robot</b> (Real) User	Thought: I will bring you the [ <b>pepsi</b> ] and the [apple] Plan: Step 1: Find an apple Step 2: Pick up the apple	100% 80% 60% 40%							
	Bring me a soda that is not coke, and a fruit.	Step 3: Bring it to you Step 4: Put it down	20%	pepa gapatus	perture south perpeture. South					
Language Model Council Andels Combined Score										

**Figure 3:** Example rollouts and likelihood of representative tokens under Grounded Decoding objective in three distinct domains: simulated tabletop rearrangement (*top*), Minigrid 2D Maze (*middle*), and real-world kitchen mobile manipulation (*bottom*). Each domain uses different prompts, grounded models, and low-level primitives. The GD formulation is shared across the domains, decoding a pre-trained langauge model with respect to domain-specific grounded models to decompose a *open-ended* instruction into actionable steps.

# Experiments

- Letters (8 tasks): Rearranging alphabetical letters (e.g., "put the letters in alphabetical order from left to right").
- Blocks & Bowls (8 tasks): Rearranging or combining blocks and bowls (e.g., "put the blocks in the matching bowls").
- Box Packing (4 tasks): Sorting food items and utensils into boxes in accordance with safety constraints and user preferences (e.g., "Can you pack the picnic box for me?").

	CLIPort		+LLM	+GD	
	Short	Long	Ungrounded	Greedy	Beam
Seen Tasks					
Letters	7%	40%	20%	43%	57%
<b>Blocks &amp; Bowls</b>	2%	62%	35%	60%	77%
Box Packing*	15%	28%	11%	<b>79</b> %	78%
Unseen Tasks					
Letters	6%	10%	19%	37%	41%
Blocks & Bowls	6%	10%	28%	44%	<b>50%</b>

**Table 1:** Average success rate for each tabletop task category. \*Box Packing tasks are seen during training, but safety and preference requirements are only enforced during evaluation.