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.

Method

- 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

Method

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

Type I. Eye VLM: 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.

Method

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

Type I. Eye VLM: 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.*

Method

- 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;*

Method

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:

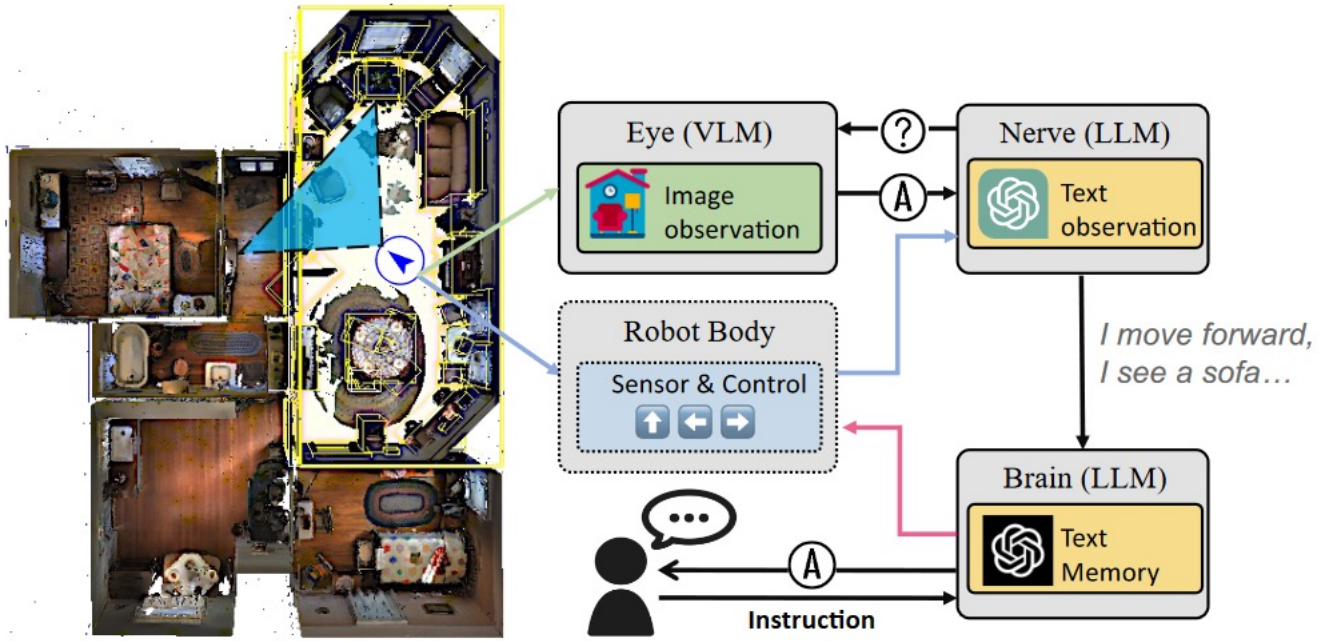


Figure 1: **The structure of LLM-Brain robot.**

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.

Type	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

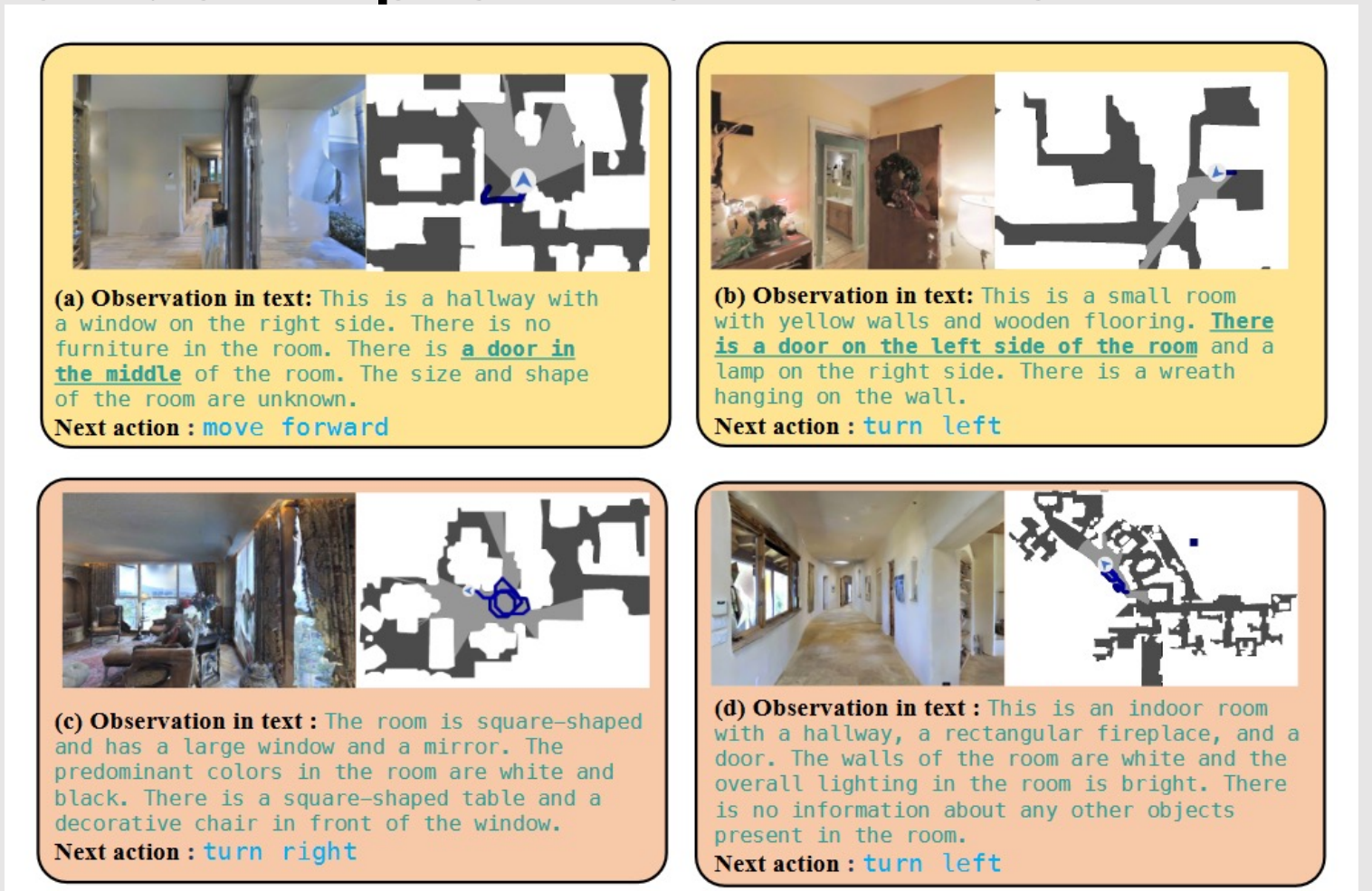
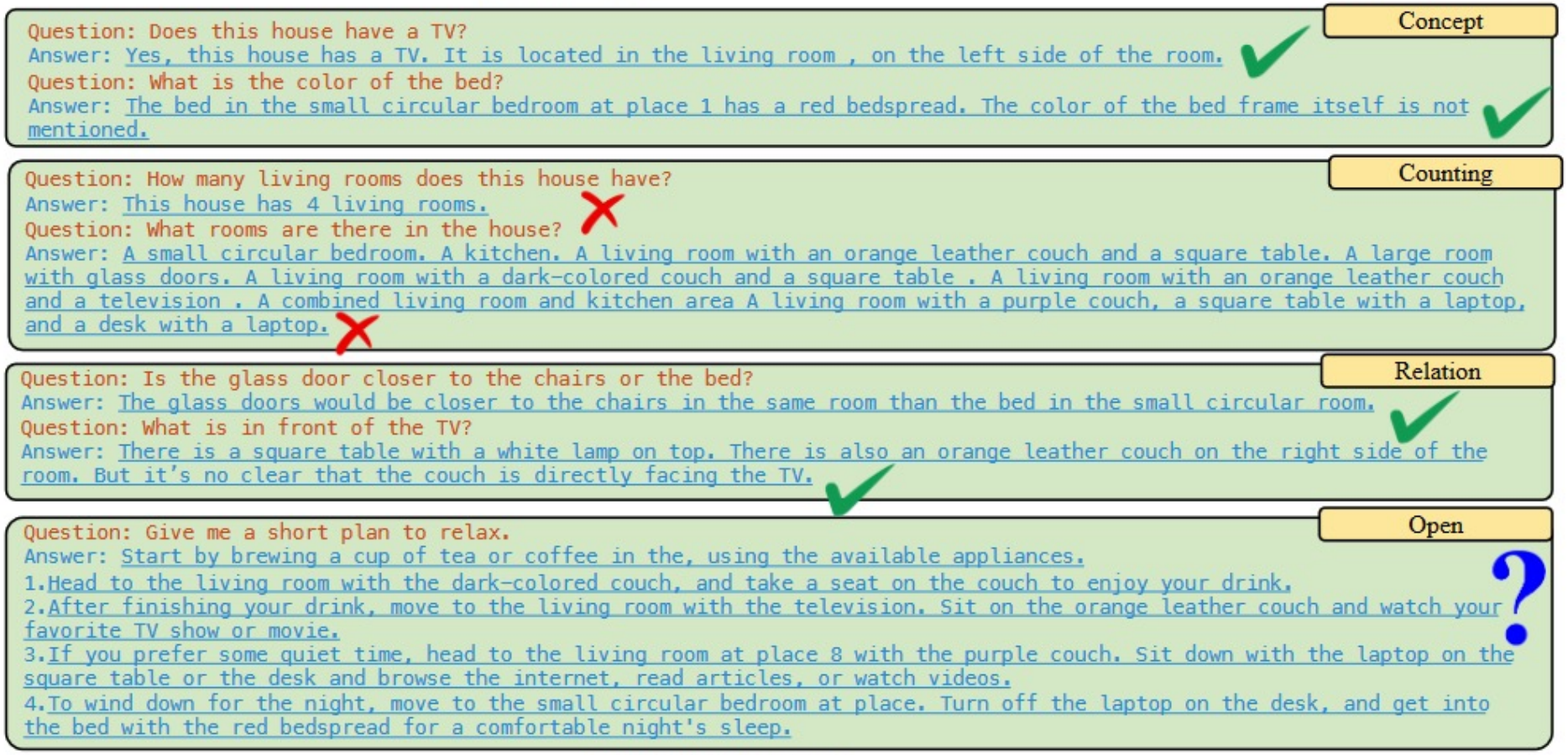


Figure 2: **The navigation decision based on observation**. One agent describes the scene, one agent summarizes the description, and one agent selects an action to take based on the summary. We show the good turns (a,b) and also the bad turns (c,d) based on textual observation.

Experiments—Embodied Question Answering

不能区分来自不同时间步的相同实例，计数问题性能差。



(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, $W = w_{1:N} = (w_1, \dots w_N)$
  - $p(W) = \prod_{n=1}^{N} p_{LLM}(w_n | w_{1:n-1})$, 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

Given: state s, instruction ℓ, terminal set $\mathcal{W}_{\text{term}}$
Initialize: $w = \{\}$, $n = 0$
while $w_n \notin \mathcal{W}_{\text{term}}$ **do**
  $n = n + 1$
  $w_n = \arg\max_{w_n \in \mathcal{W}} p_{\text{LLM}}(w_n | w_{1:n-1}, \ell)\, p_{\text{G}}(w_{1:n}|s)$
Return: w

Method

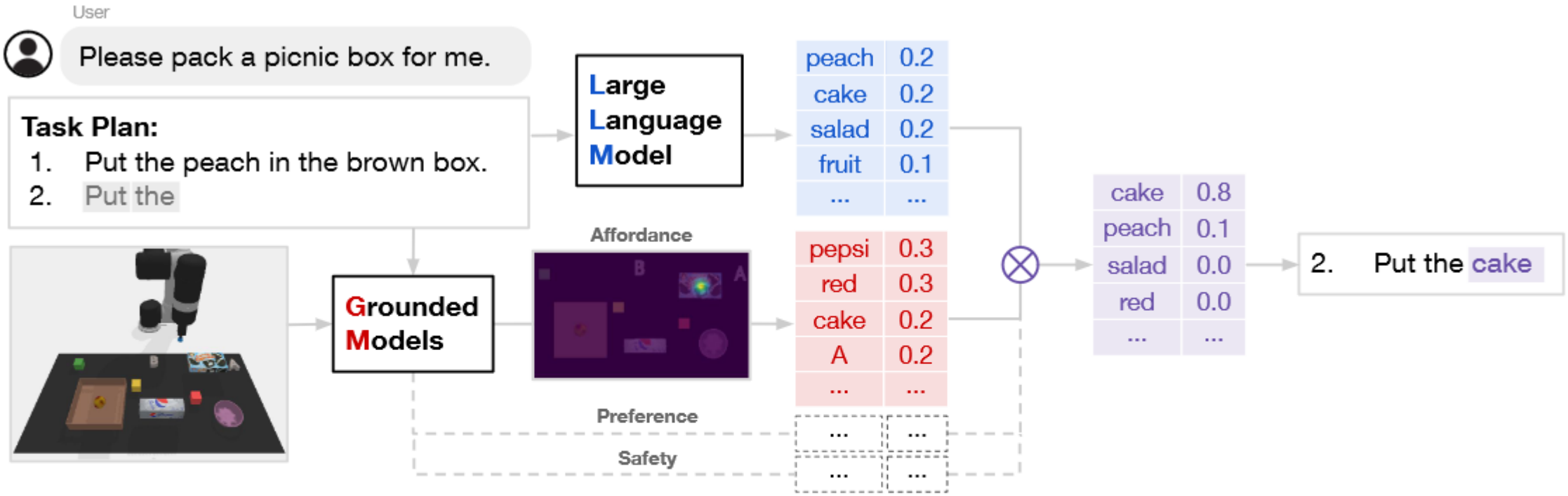


Figure 2: Overview of **Grounded Decoding**. Given a *free-form* language instruction, a language model and grounded models jointly decide the next candidate token to be decoded by combining their respective likelihood. Language model proposes likely tokens that produce goal-directed and coherent long-horizon 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 multi-task language-conditioned policy, CLIPort

LLM: InstructGPT

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

Grounding functions can be composed by:

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

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%**
Blocks & Bowls	2%	62%	35%	60%	**77%**
Box Packing*	15%	28%	11%	**79 %**	78%
Unseen Tasks					
Letters	6%	10%	19%	37%	**41%**
Blocks & Bowls	6%	10%	28%	44%	**50%**

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.