

SYNERGAI: Perception Alignment for Human-Robot Collaboration

Yixin Chen*, Guoxi Zhang*, Yaowei Zhang*, Hongming Xu*, Peiyuan Zhi, Qing Li†, Siyuan Huang†
<https://synerg-ai.github.io/>

Abstract— Recently, large language models (LLMs) have shown strong potential in facilitating human-robotic interaction and collaboration. However, existing LLM-based systems often overlook the misalignment between human and robot perceptions, which hinders their effective communication and real-world robot deployment. To address this issue, we introduce SYNERGAI, a unified system designed to achieve both perceptual alignment and human-robot collaboration. At its core, SYNERGAI employs 3D Scene Graph (3DSG) as its explicit and innate representation. This enables the system to leverage LLM to break down complex tasks and allocate appropriate tools in intermediate steps to extract relevant information from the 3DSG, modify its structure, or generate responses. Importantly, SYNERGAI incorporates an automatic mechanism that enables perceptual misalignment correction with users by updating its 3DSG with online interaction. SYNERGAI achieves comparable performance with the data-driven models in ScanQA in a zero-shot manner. Through comprehensive experiments across 10 real-world scenes, SYNERGAI demonstrates its effectiveness in establishing common ground with humans, realizing a success rate of 61.9% in alignment tasks. It also significantly improves the success rate from 3.7% to 45.68% on novel tasks by transferring the knowledge acquired during alignment.

I. INTRODUCTION

Imagine a household robot assisting humans in their homes, receiving an instruction “*help me grab my coffee mug from the kitchen*”, as depicted in Fig. 1. While this task seems routine, complications arise if the robot lacks knowledge of where the kitchen is or, more importantly, cannot discern the coffee mug among the various containers in the 3D environment. Achieving an accurate understanding of such scenes, including their semantic attributes, spatial arrangements, and personalized references like “one’s coffee mug,” is essential for the robot to effectively employ its planning, navigation, and manipulation skills. Effective communication between humans and robots, whether simple or complex, relies on establishing a common ground to facilitate human-robot collaborations, with perception serving as its foundational milestone. Despite the recent popularity of applying LLM in robotics [1]–[5], they overlook this fundamental aspect.

To achieve effective human-robot collaboration, a robotic system must meet several key requirements: (1) the ability to construct a compact representation of the 3D environment; (2) the versatility to handle various tasks under natural-language instructions in a zero-shot manner; and (3) the flexibility to quickly align with users through natural interactions, all within a coherent system.

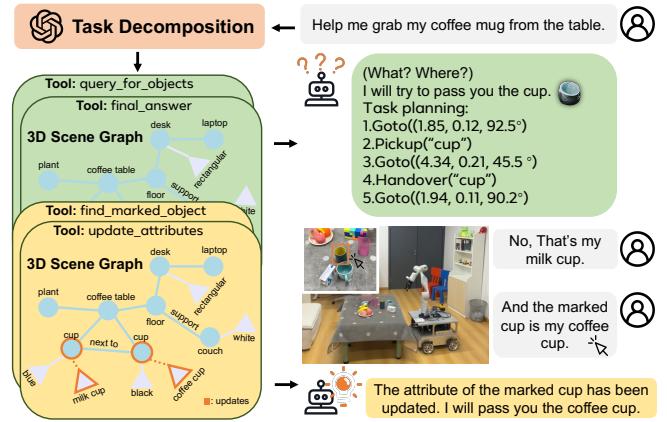


Fig. 1: **Overview of SYNERGAI.** Leveraging 3D Scene Graph (3DSG) as its representation, SYNERGAI decomposes complex tasks with LLMs and takes actions with our designed tools in intermediate steps. It interacts with humans through natural language and non-verbal mouse-clicking, facilitating human-robot collaboration and perceptual alignment by automatically modifying the data stored in 3DSG.

However, existing systems fall short of these requirements: they are not yet capable of perfectly perceiving real-world scenes and are not customized for individual users, making it difficult to learn personalized concepts such as naming conventions or preferences shaped by diverse cultures and lifestyles. Thus it is essential for the robot to quickly correct perceptual misalignment with users to enhance human-machine cooperation in completing real-world tasks.

In this paper, we propose SYNERGAI, a system that meets the aforementioned criteria. Our framework takes a collection of posed images as input, reconstructs the 3D scene and constructs a 3D Scene Graph (3DSG) [6]–[9] as its data structure. The 3DSG encapsulates hierarchical topology and key information necessary for 3D reasoning, including object categories, attributes, states, and spatial relationships.

Taking advantage of the explicit and explainable advantage of 3DSG, we leverage LLM to break down complex tasks—such as question answering, task planning, and captioning—into intermediate steps, and allocate appropriate tools for completing these steps. The toolset is designed to extract relevant knowledge from the 3DSG, make modifications, or generate responses based on the user input, supporting a wide range of open-world reasoning tasks. Additionally, we have incorporated an alignment mechanism that autonomously recognizes users’ intention of alignment and triggers a process to update the 3DSG on the fly. For higher efficiency in

* Equal Contribution.

† Corresponding Authors.

State Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI)

human-robot interaction, we have developed a user-friendly graphical interface that allows users to freely interact with the scene by dragging, zooming-in/out, changing views, marking objects and asking free-form questions at will.

We conduct extensive experiments to demonstrate SYNERGAI’s capabilities in human-robot collaboration and alignment. Results show that, even in a zero-shot manner, it achieves comparable performance with data-driven methods on ScanQA [10]. More importantly, we perform real-world alignment experiments of varying difficulties between SYNERGAI and humans to systematically assess its ability to establish common ground with humans. Our model realizes a success rate of 61.9% for the alignment tasks while providing a smooth interaction experience (64.87% per-step user satisfaction rate). It further demonstrates the capability to transfer the acquired knowledge to novel tasks by improving their success rate from 3.70% to 46.58%.

II. RELATED WORK

Human-Robot Alignment Significant attention has recently been focused on the Human-machine alignment [11], [12], especially for aligning LLMs with human intentions and values [13], [14] with Reinforcement Learning from Human Feedback (RLHF) [15]–[17] or supervised fine-tuning (SFT) [18]. In robotics, human-robot alignment centers on improving their coordination in real-world scenarios. Previous efforts have been devoted to enhancing effective human-robot interaction through dialogue, including generating help requests [19], seeking oracle in planning [20]–[22], following embodied instructions [23], and resolving the uncertainty of LLM-based planners [24]. Despite their progress in establishing natural language communications with humans [3], [25], [26], one limitation is the presumption that the robot and humans have reached a common ground, overlooking the fact that the robotic perception capabilities remain far from perfect to date. In this paper, we propose a systematic framework that enables the robot to align with humans both preemptively and during collaborative tasks. The alignment is facilitated through natural interactions, *i.e.*, natural language or a virtual interface, thereby endowing robots with correct perception aligned with individual human perspectives.

LLM in Robotics Previous research has effectively leveraged pre-trained LLMs’ in-context learning abilities for embodied agents to generate actionable task plans [1]–[3], [27]–[29], recover from failure [4], [5], [30]–[32], perform low-level control [33], or specify reward functions [34]–[37]. In order to enable these language models to perceive physical environments, visual information is either decoded with grounded models [38] or directly treated as input by multi-modal language model (MMLM) [39]–[41]. However, the power of these foundation models is often limited in separate stages of training or inference, leaving the potential for humans to *teach* the robots unexplored. Thus our framework is proposed to facilitate the robots’ potential to evolve in understanding the 3D world through interactions with humans. We utilize the structured representation of 3DSG [6]–[9] and its compatibility with LLM to boost

human-robot coordination, a critical aspect for deploying personalized humanoids in real-world scenarios.

LLM in 3D Scene Understanding The popularity of LLMs has recently spurred the development of 3D scene understanding in various tasks, *e.g.*, object referral [42]–[44], captioning [45], [46], vision-language-navigation [47], [48] and reasoning [10], [49]. The signature efforts like 3D-LLM [50], Chat-3D [51], LEO [52] and 3DMIT [53] investigate alternatives to incorporate the multi-modal inputs, *e.g.*, 3D point clouds, images, and texts, into a pre-trained LLM and further fine-tune the model on more data for downstream tasks. In this paper, we devise a framework capable of performing 3D reasoning tasks important for human-robot collaboration, *i.e.*, embodied question answering, task planning, and captioning, all in a zero-shot manner. The agent utilizes 3DSG as explicit representations and harnesses the extensive reasoning capabilities of foundation models to interact with humans and accomplish comprehensive tasks.

III. METHOD

In this section, we present 3D scene reconstruction and the construction of 3DSG in [Section III-A](#). We then illustrate the system design in [Section III-B](#) with available tools in [Table I](#).

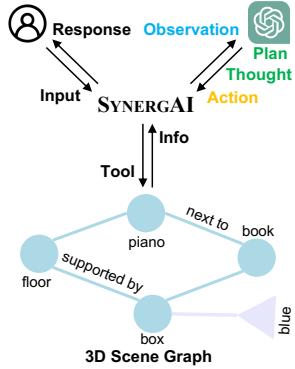
A. 3D Scene Reconstruction and 3D Scene Graph (3DSG)

From posed RGBD images, the 3D mesh of a scene can be reconstructed through either depth fusion and marching cubes algorithm [54] following ScanNet [55], or via neural rendering with state-of-the-art methods [56], [57]. Subsequently, we obtain the object instances and their semantic labels by employing 3D instance segmentation [58], 2D image classification [40] and multi-view association [59].

3DSG have recently emerged as an effective world representation for robotics [9], [60]–[63], capturing a hierarchically organized semantic graph representation of an environment with the versatility to encode the object states and spatial relationships. The data structure is suitable for parsing and we convert the graph to the LLM inputs similar to JSON serialization (see example observations in [Fig. 2](#)). The 3DSG is defined as a hierarchical graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v \in \mathcal{V}$ represents one distinct 3D object instance, denoted by its position, size, and attributes. The edges \mathcal{E} represent spatial relationships between nodes.

To construct the scene graph \mathcal{G} , we first instantiate the nodes with the instance predictions from the reconstructed mesh and assign object classes with their corresponding semantic labels. To acquire the semantic-rich object states and attributes, we utilize 3D object segmentation to identify its occurrence in the multi-view images through rendering. The images are then cropped with the rendered bounding boxes and processed through BLIP2 [40] to generate information about the object’s color, texture, shape, material and affordance. Following prior work [8], [43], [64], we utilize the positional and size information of objects within the scene to model the following spatial relations:

- **Horizontal proximity** reflects the distance of horizontal object placements, *e.g.*, close or far.



What is the item above the blue box?

The item above the blue box is a book.

Prompt: You are an assistant agent in a room. When the user ask for information or ask you to learn something, using your tools to collect the required information and propose data modification. Make a rough plan first. Each step of this plan corresponds to a single action. At each step, select an action, analyze your observation, and determine your next action based on the observations you received so far, the user's request, and your original plan.

Plan:

1. Use `query_for_items` to find items related to "blue box" to identify the blue box.
2. Use `query_for_relations` to find items that are above the identified blue box.

...

Action: ````{"action": "query_for_items", "action_input": {"query": "blue box"} } ````

Step 0

Observation: "The position of the box (id: 21) is [-0.25, -2.24, 0.23]. The box (id: 21) has attributes: ['cuboid']."

Thought: Given the identification of the blue box with id 21, the next step is to find items that are above this blue box.

Action: ````{"action": "query_for_relations", "action_input": {"item_id_list": ["21"]}} ````

Step 1

Observation: "The position of the box (id: 21) is [-0.25, -2.24, 0.23]. The box (id: 21) has attributes: ['cuboid']. The box (id: 21) is support book (id: 49). The box (id: 21) is resting on floor (id: -3). "

Thought: The observation indicates that the book (id: 49) is supported by the box (id: 21), which means the book is above the blue box. This is the item we were looking for based on the user's query. Now, I will proceed to post-process this information to prepare for the final answer.

Action: ````{"action": "post_process", "action_input": {"item_id_list": ["49"]}} ````

Step 2

Fig. 2: **The design of SYNERGAI and an example interaction.** SYNERGAI represents 3D scene with 3DSGs and leverages LLMs to respond to user inputs. It is first prompted to generate a *plan*, which effectively decomposes the input task into sub-tasks to be solved in a sequential process. At each step, SYNERGAI selects a *tool* as its *action* based on the *observation*, which contains the results of the previous actions. In this example, the system identifies the correct object of relationship “on the blue box”, but incorrectly recognizes it as a book, where *perception misalignment* happens.

- **Vertical proximity** encompasses both in-contact relationships (*e.g.*, support, inside, embed), and non-contact ones (*e.g.*, above, below).
- **Allocentric** relations describe the directional ones like left, right, in front of, *etc.*, which depend contextually on the robots’ viewing direction. Our framework *dynamically* updates the allocentric relationships based on its current position and viewing angle.

We traverse all object nodes to compute spatial relations, which undergo auto-verification to rectify incorrect ones.

B. System Design

Given the 3DSG as the scene representation, we aim to develop a robot system that can communicate with humans, perform 3D reasoning tasks, and align with human perception by leveraging the power of LLM. However, this presents two major challenges: 1) The intricate nature of the tasks and the 3D scenes makes it difficult to directly utilize the full 3DSG for complex reasoning, even for LLM. 2) Language-based interaction alone is insufficient for efficiently referencing objects in the presence of erroneous labels and relations, which is essential to accomplish the alignment tasks. The following explains how the design of SYNERGAI addresses these two challenges.

Task Decomposition We begin by noting that it’s more efficient to tackle complex tasks step-by-step, similar to Chain-of-Thought (CoT) [65], and most intermediate reasoning steps exhibit locality—they are solvable once the relevant information is retrieved from the 3D scenes. Therefore, our idea is to prompt the system to decompose the complex tasks

into intermediate steps and progressively gather relevant sub-graphs from the 3DSG to tackle them. As illustrated in Fig. 2, upon receiving user input, SYNERGAI invokes a sequential process with LLM, where at each step the system receives an *observation*, generates a *thought* and selects an *action*. The actions are calls to a set of APIs called *tools*, and the *thought* is the reasoning process and rationale behind choosing the next action. The *observation* contains the summarized information in the previous action, *e.g.*, the sub-graph retrieved from 3DSG. This process continues unless the agent gathers enough information and selects the termination tool *final_answer*, which finishes the reasoning process and returns the final response to the user.

Note that the agent is prompted to compose a *plan* that outlines the task decomposition at the first step in Fig. 2. By composing such a plan, the agent is guided to recognize the user’s intent, thereby reducing its workload in later steps. Moreover, by conditioning on this plan, the agent can proceed in a top-down fashion and select actions accordingly. Note that the actual executed actions may deviate from the original plan: the agent is capable of modifying its plan during thoughts when new observation reveals mistakes. Such flexibility turns out to be essential for complex 3D reasoning tasks where they can be resolved in different ways.

Observation Starting from the second step, SYNERGAI generates an observation that summarizes the information retrieved from the 3DSG by the last action or indicates errors occurred in calling tools. The agent utilizes these observations to generate thoughts and decide its next action. The retrieved information, *i.e.*, sub-graph, contains a list of strings of objects of interest, including their names, ids, attributes,

TABLE I: **The toolset for SYNERGAI.** They are designed as Python APIs, with the top five tools for 3D reasoning, the following four for alignment, and the last two for generating responses to the user.

Tool	Input	Return	Description
query_for_objects	String, \mathcal{G}	List[Object]	Collect the objects mentioned in a user input.
query_for_relations	List[Object], \mathcal{G}	List[Relation]	Collect the relations associated with a list of objects.
find_marked_object	Click, \mathcal{G}	Object	Collect the information of the object marked by the user.
calculate_mid_point	List[Point], \mathcal{G}	Point	Calculate the midpoint of a list of Points.
find_object_closest	Point, \mathcal{G}	Object	Collect the object closest to a point.
update_name	List[String], List[Object], \mathcal{G}	\mathcal{G} , List[Object]	Update the labels of a list of objects.
update_attributes	Object, List[String], \mathcal{G}	\mathcal{G} , List[Relation]	Update the attributes of an object.
add_relation	Object, Object, Relation, \mathcal{G}	\mathcal{G} , List[Relation]	Add a relation between two objects.
delete_relation	Object, Object, Relation, \mathcal{G}	\mathcal{G} , List[Relation]	Remove a relation between two objects.
post_process	List[Objects]	List[Objects]	Return the relevant information for the graphical user interface (GUI).
final_answer	String	String	Return the final response for the input.

and relations with other objects. We convert them into sentences compatible with LLMs using templates. For example, we use “The {object.name} (id: {object.id}) has attributes: {object.attributes}.” as the template for rendering the attribute-related information.

Human-Robot Interaction As previously mentioned, object references are essential for achieving alignment as they provide a common ground for communication. However, under erroneous perception, users may struggle to reference objects with incorrect labels and attributes in 3D environments via pure language-based interactions. Motivated by the higher efficiency of non-verbal cues in object reference than pure language in human communication [66]–[68], we address this challenge by implementing a GUI, which includes the reconstructed 3D scene, the segmentation, and the 3D Scene Graph. It further allows users to mark objects by clicking in the 3D scene or the nodes in the 3DSG. This non-verbal interaction is robust against semantic errors and can function if an object is properly segmented. Users can thus refer to an item as “the marked object” in their inputs after clicking on it, which significantly reduces their workload in human-robot alignment.

Tools The final piece of SYNERGAI is the tools available for each intermediate step, which are a set of Python APIs summarized in Table I. The tools are designed to extract relevant information from the 3DSG, modify its structure, or generate responses based on the user input. Our designed tools can be categorized into three purposes, 3D reasoning, alignment, and response generation, but during the sequential task-solving process, SYNERGAI automatically identifies the user intent and selects tools accordingly based on the user inputs. This means SYNERGAI can decide if it is required to perform 3D reasoning or align with humans based on the users’ natural language instructions, without the need for mode shifting or explicit task specifications.

Implementation We develop SYNERGAI based on the LangChain [69] framework, with GPT-4 turbo as the underlying LLM backend. LangChain manages the sequential process of *observation*, *thought*, and *action*, including generating step-wise prompts, parsing the agent’s output, and executing the tools. The step-wise prompt combines a template with the latest observation, doc-strings of tools, and

all historical observations, actions, and thoughts.

IV. EXPERIMENTS

A. Human-Robot Collaboration in Zero-shot 3D Reasoning

Dataset and Metrics To extensively evaluate SYNERGAI’s capability in high-level human-robot collaboration with language-guided interactions, we evaluate its zero-shot 3D reasoning task performance, with qualitative demonstrations in object captioning, scene captioning, question-answering and task planning in Fig. 3. To quantitatively evaluate its performance, we select the one that best reflects its reasoning capability, question-answering (QA) task. We utilize the ScanQA [10] benchmark to test our system. Since we evaluate our system in a zero-shot manner *without alignment*, we directly test its performance on its validation set, encompassing a total of 4675 scene QA tasks from 71 3D scenes in ScanNet [55]. Following common practice [10], we assess the QA performance using Exact Match (EM), CIDEr, BLEU-1, METEOR, and ROUGE.

Performance Analysis From Fig. 3, we can see SYNERGAI is capable of accomplishing various 3D reasoning tasks in a zero-shot manner. Table II demonstrate the quantitative results on ScanQA, where our model achieves comparable performance to the methods *fine-tuned* on ScanQA, yet fails short on the metric EM. The discrepancy can be attributed to the fact that our system generates answers by leveraging the power of LLM, which aligns more closely with human preference. For instance, given the question “*What is sitting on top of the toilet tank lid?*”, our response is “*A towel is sitting on top of the toilet tank lid.*”, whereas the ground-truth response is simply “*towel*”. The differentiation in format significantly affects metrics like EM.

B. Human-Robot Alignment

Setup We systematically assess SYNERGAI’s capability in achieving perceptual alignment with humans spanning 10 real-world scenes sourced from the ScanNet [55] dataset. The tests include two phases, *i.e.*, **alignment tasks** and **knowledge transfer**. In the first phase, we devise 42 alignment tasks targeted at the perception errors related to object naming, shape, material, and spatial relations. The tasks are designed in the form of question-answering and

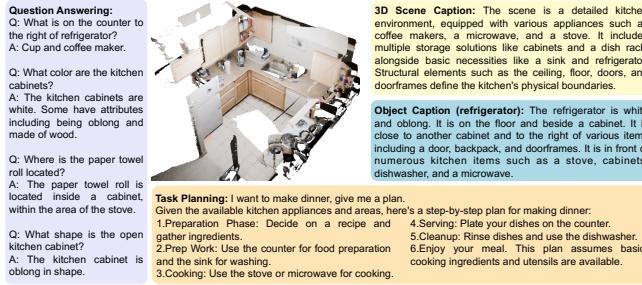


Fig. 3: Qualitative results of 3D Reasoning Tasks.

TABLE II: Zero-shot Performance on ScanQA. Our model reaches comparable performance with fine-tuned baselines.

Model	EM	CIDEr	BLUE-1	METEOR	ROUGE
Fine-tuned					
VoteNet+MCAN [10]	17.3	54.7	28.0	11.4	29.8
ScanRefer+MCAN [10]	18.6	55.4	26.9	11.5	30.0
ScanQA [10]	21.0	64.9	30.2	13.1	33.3
Flamingo (MultiView) [50]	18.8	55.0	25.6	11.3	31.1
BLIP2 (MultiView) [50]	13.6	45.7	29.7	11.3	26.6
3D-LLM (Flamingo) [50]	20.4	59.2	30.3	12.2	32.3
Zero-shot					
Ours	11.4	57.9	30.1	12.9	30.4

the objective is to correct the system’s perception through interaction, ensuring that the system’s final responses ultimately align with humans. Based on the number of objects involved with perception misalignment, the tasks are divided into *EASY* ($\text{misalignment}=1$, 25 tasks) to *HARD* ($\text{misalignment}>1$, 17 tasks). In the second phase, we design 27 novel tasks to measure if the knowledge acquired from the alignment can be transferred.

We engage the participation of 10 human subjects for the alignment experiment. They first undergo a preparatory session to become acquainted with our system under instructions, following which each participant is assigned tasks across 3 scenes. During the alignment experiments, subjects are tasked to inspect and correct the perception of the agent within the GUI using both natural language and mouse-click interactions. Participants are required to provide a binary rating for the system’s response to each interaction and assess the success of the alignment after each task.

Metrics We evaluate the success rate of the alignment task with both human (SR_{Human}) and LLM (SR_{LLM}) judgments. The answer accuracy of our system before the alignment is evaluated by LLM as a baseline, denoted as SR_{Init} . The ratio of reasonable responses ($RR_{Interaction}$) from our system during the interactions is judged by the human subjects. For task difficulty, we report the average number of interactions required to complete per task ($\#Inter./Task$) and the average number of actions that the system executes to generate responses in one interaction ($\#Action/Inter.$). The query ratio (QR) is reported as the ratio of tokens required to encode the extracted relevant sub-graphs and the full 3DSG. The knowledge transfer tasks are also measured by the success rate of question-answering. We provide the mean and 95% confidence intervals for all the metrics.

Results Table III and Fig. 5 present the major results for the alignment experiments, and Table IV demonstrates the

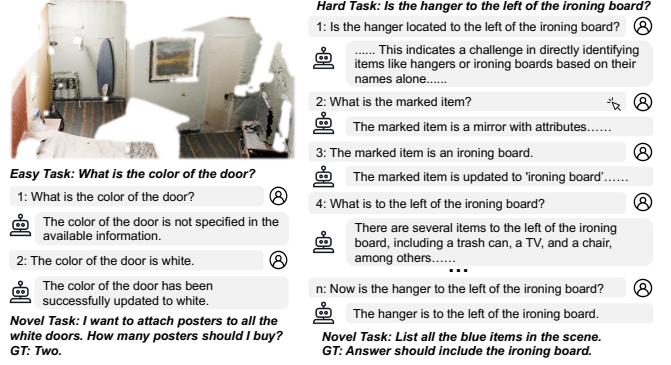


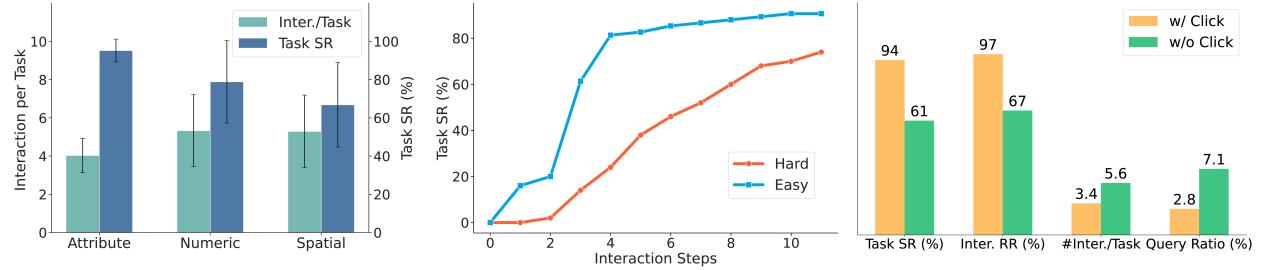
Fig. 4: Examples of human-robot alignment. Humans solve the *EASY* task within shorter interaction steps compared with the *HARD* task, where the user checks and corrects the label of the ironing board by clicking (the 2nd & 3rd user inputs). *Novel* tasks are designed such that knowledge from the alignment is required for their completion.

quantitative results for the knowledge transfer experiments. Below we summarize the key observations.

- **Our system is capable of achieving perceptual alignment with humans**, which is validated by the task success rate from both human (SR_{Human}) and machine (SR_{LLM}) evaluations. Meanwhile, it provides reasonable responses at each step of the interactions from the interaction satisfactory ratio.
- **The task difficulty significantly impacts both success rates and efforts for alignment.** The SR_{Human} , SR_{LLM} and $RR_{Interaction}$ all present a notable gap between the *EASY* and *HARD* tasks. Efforts for alignment ($\#Inter./Task$) are twice for the *HARD* tasks, while the $\#Action/Inter.$ remains similar. This suggests a human tendency to decompose complex alignment to simpler tasks that impose a constant load for LLM-based agent. The *Query Ratio* is also higher for *HARD* tasks as they involve more objects in the 3DSG. Fig. 5a shows an interaction step increase from tasks related to unary semantics, *i.e.*, object attributes, to those involving n-ary information such as numeric counting or spatial relations, with success ratios consistently decreasing.
- **The object reference plays a pivotal role in perception alignment.** We conduct an ablation study to verify the significance of mouse-clicking interactions. Results in Fig. 5c show a notable decline in the task success rate and an increase in the steps required for alignment without mouse clicking. This is attributed to the extra steps needed to reference the relevant objects, constituting a common ground for communication. Such findings align with our motivation for designing the GUI to facilitate efficient object marking, and underscore the necessity to develop improved human-robot interaction interfaces for future VR/AR and robot applications.
- **Our model can transfer the knowledge acquired in the alignment to novel tasks.** We design a baseline model that leverages in-context prompting [70], which takes the **ground-truth** knowledge for the alignment

TABLE III: Quantitative results of human-robot alignment. “SR” denotes the success rate for the alignment tasks, “RR” for the rate of reasonable responses and “QR” for the query ratio of the 3DSG.

	Alignment Task				Task Difficulty		
	SR _{Init} (%)	SR _{Human} (%)	SR _{LLM} (%)	RR _{Interaction} (%)	#Inter./Task	#Action/Inter.	QR (%)
EASY	8.00	91.18±9.89	72.00±14.14	74.06±9.19	3.38±0.68	3.23±0.17	2.55±0.78
HARD	0.0	72.48±14.2	47.06±20.15	51.36±9.49	6.45±1.07	3.30±0.24	3.86±1.23
OVERALL	4.76	83.61±8.36	61.90±11.83	64.87±7.31	4.65±0.74	3.26±0.13	3.08±0.68



(a) Success rate and #Iterations by task. (b) Task success rate vs. #interactions.

(c) Ablation on mouse clicking.

Fig. 5: Statistics of alignment experiments. (a) The success rate decreases for more complex tasks with increasing interaction steps required to achieve alignment. (b) The trend of task success rate as the interaction step increases. (c) The user interface impacts users’ ability to reference objects and hinders the alignment performance when mouse clicks are not used.

TABLE IV: Knowledge Transfer to Novel Tasks. Results are reported in Success Rate (%) as measured by LLM.

	Init.	In-Context Prompting [70]	Ours
EASY	2.56	23.08	43.59
HARD	4.76	28.57	47.62
OVERALL	3.70	25.93	45.68

tasks as additional inputs in the prompt, to answer the questions in the novel tasks. On the contrary, our model directly takes the updated 3DSG from the alignment phase, though not perfect. As shown in Table IV, our improved results prove the effectiveness of explicitly updating the 3DSG during alignment, which leads to better systematic generalizability for novel tasks.

C. Discussion and Limitation

In this paper, we focus on evaluating the applicability of LLM-based agents to function **in real-world settings**, emphasizing their robustness to operate effectively under imperfect perceptions rather than showcasing their strengths **in ideal settings**. We believe this is fundamental for the practical use of LLM-based agents in the real world, *e.g.*, comprehensive reasoning and task planning by callable execution module as action primitives as in Fig. 1.

We identify the following limitations of our system. First, despite the remarkable reasoning capabilities of LLMs, their tendency towards hallucination may lead to plans and responses that deviate from the 3DSG. Second, we design tools for our system to ensure their functionality completeness for existing tasks. This indicates our framework may not be able to perform certain “novel” operations, such as grouping the

nodes in the 3DSG. Third, the performance of our system is limited to the 3D reconstruction and segmentation methods to construct 3DSG, and it currently operates at the object level. Future efforts could focus on expanding its capabilities to encompass more levels of perception and understanding. Finally, while SYNERGAI’s zero-shot reasoning capability can generalize to novel scenes, its improved representation after alignment only works in one specific scene. Leveraging its interaction with humans to enhance its adaptability and scalability to novel scenes is one important future direction.

V. CONCLUSION

In this paper, we present SYNERGAI, a novel framework designed to achieve human-robot collaboration and bridge their perceptual gap. SYNERGAI leverages 3D reconstruction to create 3D Scene Graph as innate representations, and decomposes complex tasks and solve them sequentially through natural interactions with humans. Experimental results validate SYNERGAI’s capability in zero-shot 3D reasoning, achieving competitive performance on the ScanQA benchmark without alignment. Furthermore, our alignment experiments highlight its proficiency in achieving human-robot alignment across varying levels of task difficulty with high user satisfaction ratio and transferability to novel tasks. We hope our efforts and insights could facilitate the deployment of LLM-based robot systems in real-world scenarios.

ACKNOWLEDGEMENT

The authors would like to thank Huangyue Yu (BIGAI) for 3D scene graph generation, and Ziyu Zhu (Tsinghua University, BIGAI) for 3D semantic segmentation. The authors would also like to thank the subjects for their efforts during the alignment human study.

REFERENCES

- [1] T. Silver, V. Hariprasad, R. S. Shuttleworth, N. Kumar, T. Lozano-Pérez, and L. P. Kaelbling, “Pddl planning with pretrained large language models,” in *NeurIPS 2022 foundation models for decision making workshop*, 2022. [1](#) [2](#)
- [2] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, “Llm-planner: Few-shot grounded planning for embodied agents with large language models,” in *International Conference on Computer Vision (ICCV)*, 2023. [1](#) [2](#)
- [3] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, *et al.*, “Do as i can, not as i say: Grounding language in robotic affordances,” in *Conference on Robot Learning (CoRL)*. PMLR, 2023. [1](#) [2](#)
- [4] Y. Ding, X. Zhang, S. Amiri, N. Cao, H. Yang, A. Kaminski, C. Esselink, and S. Zhang, “Integrating action knowledge and llms for task planning and situation handling in open worlds,” *Autonomous Robots*, vol. 47, 2023. [1](#) [2](#)
- [5] Z. Liu, A. Bahety, and S. Song, “Reflect: Summarizing robot experiences for failure explanation and correction,” in *Conference on Robot Learning (CoRL)*. PMLR, 2023. [1](#) [2](#)
- [6] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese, “3d scene graph: A structure for unified semantics, 3d space, and camera,” in *International Conference on Computer Vision (ICCV)*, 2019. [1](#) [2](#)
- [7] Y. Chen, S. Huang, T. Yuan, S. Qi, Y. Zhu, and S.-C. Zhu, “Holistic++ scene understanding: Single-view 3d holistic scene parsing and human pose estimation with human-object interaction and physical commonsense,” in *International Conference on Computer Vision (ICCV)*, 2019. [1](#) [2](#)
- [8] J. Wald, H. Dhamo, N. Navab, and F. Tombari, “Learning 3d semantic scene graphs from 3d indoor reconstructions,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [1](#) [2](#)
- [9] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, “Kimera: From slam to spatial perception with 3d dynamic scene graphs,” *International Journal of Robotics Research (IJRR)*, 2021. [1](#) [2](#)
- [10] D. Azuma, T. Miyanishi, S. Kurita, and M. Kawanabe, “Scanqa: 3d question answering for spatial scene understanding,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [2](#) [4](#) [5](#)
- [11] J.-M. Hoc, “From human–machine interaction to human–machine cooperation,” *Ergonomics*, 2000. [2](#)
- [12] B. Christian, *The alignment problem: How can machines learn human values?* Atlantic Books, 2021. [2](#)
- [13] J. Ji, T. Qiu, B. Chen, B. Zhang, H. Lou, K. Wang, Y. Duan, Z. He, J. Zhou, Z. Zhang, *et al.*, “Ai alignment: A comprehensive survey,” *arXiv preprint arXiv:2310.19852*, 2023. [2](#)
- [14] S. Casper, X. Davies, C. Shi, T. K. Gilbert, J. Scheurer, J. Rando, R. Freedman, T. Korbak, D. Lindner, P. Freire, *et al.*, “Open problems and fundamental limitations of reinforcement learning from human feedback,” *Transactions on Machine Learning Research*, 2023. [2](#)
- [15] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, “Deep reinforcement learning from human preferences,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. [2](#)
- [16] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, “Training language models to follow instructions with human feedback,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. [2](#)
- [17] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, “Direct preference optimization: Your language model is secretly a reward model,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. [2](#)
- [18] J. Ji, B. Chen, H. Lou, D. Hong, B. Zhang, X. Pan, J. Dai, and Y. Yang, “Aligner: Achieving efficient alignment through weak-to-strong correction,” *arXiv preprint arXiv:2402.02416*, 2024. [2](#)
- [19] S. Tellex, R. Knepper, A. Li, D. Rus, and N. Roy, “Asking for help using inverse semantics,” *Robotics: Science and Systems Foundation*, 2014. [2](#)
- [20] K. Nguyen and H. Daumé III, “Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning,” in *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019. [2](#)
- [21] J. Thomason, M. Murray, M. Cakmak, and L. Zettlemoyer, “Vision-and-dialog navigation,” in *Conference on Robot Learning (CoRL)*. PMLR, 2020. [2](#)
- [22] A. Padmakumar, J. Thomason, A. Shrivastava, P. Lange, A. Narayan-Chen, S. Gella, R. Piramuthu, G. Tur, and D. Hakkani-Tur, “Teach: Task-driven embodied agents that chat,” in *AAAI Conference on Artificial Intelligence (AAAI)*, 2022. [2](#)
- [23] X. Gao, Q. Gao, R. Gong, K. Lin, G. Thattai, and G. S. Sukhatme, “Dialfred: Dialogue-enabled agents for embodied instruction following,” in *IEEE Robotics and Automation Letters (RA-L)*. IEEE, 2022. [2](#)
- [24] A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, J. Varley, *et al.*, “Robots that ask for help: Uncertainty alignment for large language model planners,” in *Conference on Robot Learning (CoRL)*. PMLR, 2023. [2](#)
- [25] S. Yao, H. Chen, J. Yang, and K. Narasimhan, “Webshop: Towards scalable real-world web interaction with grounded language agents,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. [2](#)
- [26] X. Deng, Y. Gu, B. Zheng, S. Chen, S. Stevens, B. Wang, H. Sun, and Y. Su, “Mind2web: Towards a generalist agent for the web,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. [2](#)
- [27] A. Zeng, M. Attarian, B. Ichter, K. Choromanski, A. Wong, S. Welker, F. Tombari, A. Purohit, M. Ryoo, V. Sindhwani, *et al.*, “Socratic models: Composing zero-shot multimodal reasoning with language,” *arXiv preprint arXiv:2204.00598*, 2022. [2](#)
- [28] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, and P. Stone, “Llm+ p: Empowering large language models with optimal planning proficiency,” *arXiv preprint arXiv:2304.11477*, 2023. [2](#)
- [29] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, *et al.*, “Inner monologue: Embodied reasoning through planning with language models,” in *Conference on Robot Learning (CoRL)*. PMLR, 2023. [2](#)
- [30] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, “Voyager: An open-ended embodied agent with large language models,” in *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023. [2](#)
- [31] Z. Wang, S. Cai, G. Chen, A. Liu, X. S. Ma, and Y. Liang, “Describe, explain, plan and select: interactive planning with llms enables open-world multi-task agents,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2024. [2](#)
- [32] P. Zhi, Z. Zhang, M. Han, Z. Zhang, Z. Li, Z. Jiao, B. Jia, and S. Huang, “Closed-loop open-vocabulary mobile manipulation with gpt-4v,” *arXiv preprint arXiv:2404.10220*, 2024. [2](#)
- [33] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, “Code as policies: Language model programs for embodied control,” in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2023. [2](#)
- [34] A. Tam, N. Rabinowitz, A. Lampinen, N. A. Roy, S. Chan, D. Strouse, J. Wang, A. Banino, and F. Hill, “Semantic exploration from language abstractions and pretrained representations,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. [2](#)
- [35] M. Kwon, S. M. Xie, K. Bullard, and D. Sadigh, “Reward design with language models,” *arXiv preprint arXiv:2303.00001*, 2023. [2](#)
- [36] Y. Du, O. Watkins, Z. Wang, C. Colas, T. Darrell, P. Abbeel, A. Gupta, and J. Andreas, “Guiding pretraining in reinforcement learning with large language models,” in *International Conference on Machine Learning (ICML)*. PMLR, 2023. [2](#)
- [37] H. Hu and D. Sadigh, “Language instructed reinforcement learning for human-ai coordination,” in *International Conference on Machine Learning (ICML)*. PMLR, 2023. [2](#)
- [38] W. Huang, F. Xia, D. Shah, D. Driess, A. Zeng, Y. Lu, P. Florence, I. Mordatch, S. Levine, K. Hausman, *et al.*, “Grounded decoding: Guiding text generation with grounded models for robot control,” *arXiv preprint arXiv:2303.00855*, 2023. [2](#)
- [39] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, *et al.*, “Palm-e: An embodied multimodal language model,” in *International Conference on Machine Learning (ICML)*. PMLR, 2023. [2](#)
- [40] J. Li, D. Li, S. Savarese, and S. Hoi, “BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models,” in *International Conference on Machine Learning (ICML)*, 2023. [2](#) [9](#)
- [41] R. OpenAI, “Gpt-4 technical report.” *arXiv preprint arXiv:2303.08774*, 2023. [2](#)
- [42] D. Z. Chen, A. X. Chang, and M. Nießner, “Scanrefer: 3d object

- localization in rgb-d scans using natural language,” in *European Conference on Computer Vision (ECCV)*, 2020. 2
- [43] P. Achlioptas, A. Abdelreheem, F. Xia, M. Elhoseiny, and L. Guibas, “Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes,” in *European Conference on Computer Vision (ECCV)*, 2020. 2
- [44] Z. Zhu, Z. Zhang, X. Ma, X. Niu, Y. Chen, B. Jia, Z. Deng, S. Huang, and Q. Li, “Unifying 3d vision-language understanding via promptable queries,” in *European Conference on Computer Vision (ECCV)*. Springer, 2024, pp. 188–206. 2
- [45] Z. Chen, A. Gholami, M. Nießner, and A. X. Chang, “Scan2cap: Context-aware dense captioning in rgb-d scans,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2
- [46] Z. Yuan, X. Yan, Y. Liao, Y. Guo, G. Li, S. Cui, and Z. Li, “X-trans2cap: Cross-modal knowledge transfer using transformer for 3d dense captioning,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2
- [47] C.-Y. Ma, J. Lu, Z. Wu, G. AlRegib, Z. Kira, R. Socher, and C. Xiong, “Self-monitoring navigation agent via auxiliary progress estimation,” in *International Conference on Learning Representations (ICLR)*, 2019. 2
- [48] Y. Hong, Q. Wu, Y. Qi, C. Rodriguez-Opozo, and S. Gould, “Vln bert: A recurrent vision-and-language bert for navigation,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2
- [49] X. Ma, S. Yong, Z. Zheng, Q. Li, Y. Liang, S.-C. Zhu, and S. Huang, “Sqa3d: Situated question answering in 3d scenes,” in *International Conference on Learning Representations (ICLR)*, 2023. 2
- [50] Y. Hong, H. Zhen, P. Chen, S. Zheng, Y. Du, Z. Chen, and C. Gan, “3d-lm: Injecting the 3d world into large language models,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2, 5
- [51] Z. Wang, H. Huang, Y. Zhao, Z. Zhang, and Z. Zhao, “Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes,” *arXiv preprint arXiv:2308.08769*, 2023. 2
- [52] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang, “An embodied generalist agent in 3d world,” *arXiv preprint arXiv:2311.12871*, 2023. 2
- [53] Z. Li, C. Zhang, X. Wang, R. Ren, Y. Xu, R. Ma, and X. Liu, “3dmft: 3d multi-modal instruction tuning for scene understanding,” *arXiv preprint arXiv:2401.03201*, 2024. 2
- [54] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3d surface construction algorithm,” *ACM Transactions on Graphics (TOG)*, 1987. 2, 9, 10
- [55] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, “Scannet: Richly-annotated 3d reconstructions of indoor scenes,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 4, 9, 10
- [56] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger, “Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 9, 10
- [57] J. Ni, Y. Chen, B. Jing, N. Jiang, B. Wang, B. Dai, P. Li, Y. Zhu, S.-C. Zhu, and S. Huang, “Phyrecon: Physically plausible neural scene reconstruction,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2
- [58] Z. Zhu, X. Ma, Y. Chen, Z. Deng, S. Huang, and Q. Li, “3d-vista: Pre-trained transformer for 3d vision and text alignment,” in *International Conference on Computer Vision (ICCV)*, 2023. 2, 9, 10
- [59] Q. Gu, A. Kuwajerwala, S. Morin, K. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, C. Gan, C. de Melo, J. Tenenbaum, A. Torralba, F. Shkurti, and L. Paull, “Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning,” in *International Conference on Robotics and Automation (ICRA)*, 2023. 2, 9
- [60] A. Kurenkov, R. Martín-Martín, J. Ichnowski, K. Goldberg, and S. Savarese, “Semantic and geometric modeling with neural message passing in 3d scene graphs for hierarchical mechanical search,” in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2021. 2
- [61] N. Hughes, Y. Chang, and L. Carlone, “Hydra: A real-time spatial perception system for 3d scene graph construction and optimization,” *arXiv preprint arXiv:2201.13360*, 2022. 2
- [62] C. Agia, K. M. Jatavallabhula, M. Khodeir, O. Miksic, V. Vineet, M. Mukadam, L. Paull, and F. Shkurti, “Taskography: Evaluating robot task planning over large 3d scene graphs,” in *Conference on Robot Learning (CoRL)*. PMLR, 2022. 2
- [63] Z. Ravichandran, L. Peng, N. Hughes, J. D. Griffith, and L. Carlone, “Hierarchical representations and explicit memory: Learning effective navigation policies on 3d scene graphs using graph neural networks,” in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2022. 2
- [64] B. Jia, Y. Chen, H. Yu, Y. Wang, X. Niu, T. Liu, Q. Li, and S. Huang, “Sceneverse: Scaling 3d vision-language learning for grounded scene understanding,” *arXiv preprint arXiv:2401.09340*, 2024. 2, 9
- [65] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., “Chain-of-thought prompting elicits reasoning in large language models,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3
- [66] D. McNeill, *How language began: Gesture and speech in human evolution*. Cambridge University Press, 2012. 4
- [67] M. A. Arribalzaga, K. Liebal, and S. Pika, “Primate vocalization, gesture, and the evolution of human language,” *Current anthropology*, 2008. 4
- [68] Y. Chen, Q. Li, D. Kong, Y. L. Kei, S.-C. Zhu, T. Gao, Y. Zhu, and S. Huang, “Yourefit: Embodied reference understanding with language and gesture,” in *International Conference on Computer Vision (ICCV)*, 2021. 4
- [69] H. Chase, “Langchain,” 2022. [Online]. Available: <https://github.com/hwchase17/langchain> 4
- [70] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 5, 6
- [71] I. Corporation, “Intel realsense technology,” <https://www.intelrealsense.com/>. [Online]. Available: <https://www.intelrealsense.com/> 9
- [72] M. Corporation, “Microsoft kinect,” <https://developer.microsoft.com/en-us/windows/kinect/>. [Online]. Available: <https://developer.microsoft.com/en-us/windows/kinect/> 9
- [73] Y. Mao, Y. Zhang, H. Jiang, A. X. Chang, and M. Savva, “Multiscan: Scalable rgbd scanning for 3d environments with articulated objects,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 9
- [74] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., “Learning transferable visual models from natural language supervision,” in *International Conference on Machine Learning (ICML)*, 2021. 9

SYNERGAI: PERCEPTION ALIGNMENT FOR HUMAN-ROBOT COLLABORATION

APPENDIX

I. DETAILS OF SYNERGAI

A. 3D Scene Reconstruction

a) *Reconstruction and Segmentation*: From a sequence of posed RGBD images, we can reconstruct the 3D scene with several solutions. For one, the depth frames can be fused into a Truncated Signed Distance Field (TSDF) volume using the camera trajectory, and the surface mesh is reconstructed using the marching cubes algorithm [54] following ScanNet [55]. With the recent development of Neural Radiance Fields (NeRFs), we can also optimize the neural implicit representation of the 3D scene via signed distance fields (SDF) by volume rendering, with MonoSDF [56] being a signature method. In the reconstructed scene, instance segmentation is necessary to obtain information about objects within the scene. We utilize the 3D-VisTA [58] method to segment and extract positional and size information of the 3D objects.

The 3D scene can also be represented by point clouds, where the points can be accumulated from the depth image like ConceptGraphs [59]. In this way, the semantic labels can be attained by merging the image-wise prediction from 2D foundation models from multi-views.

b) *Data Collection*: In real-world scenarios, comprehensive scene data is essential, including RGBD data, and camera intrinsic and extrinsic parameters, to achieve decent reconstruction results. To obtain this data, we employ RGBD cameras, like RealSense [71] or Kinect [72] from existing robot setups, or from captures from iPhone ARKit packages following Multiscan [73].



Fig. A.1: **Qualitative results of 3D reconstruction and segmentation.** MonoSDF [56] and MultiScan [73] reconstruct the 3D scenes from posed RGBD images and we apply 3D-VisTA [58] to obtain segmentation. ConceptGraphs [59] progressively reconstruct the 3D point clouds and assign semantic labels to them. The figure shows that different methods reveal limitations and failures in both reconstruction or segmentation.

B. 3D Scene Graph (3DSG)

Following prior work [64], our 3D scene graph is defined as a hierarchical graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the nodes \mathcal{V} comprises $\mathcal{V}_1 \cup \mathcal{V}_2 \cup \dots \cup \mathcal{V}_K$, with \mathcal{V}_k representing the set of nodes at a particular hierarchical level. Each node v represents one distinct 3D object instance and the edges \mathcal{E} represent spatial relationships between nodes. The relationships that our 3D scene graph captures are shown in Table A.1. The hierarchies are determined by the support relationship; for instance, objects supported by the floor constitute \mathcal{V}_0 , while objects supported by the table will form \mathcal{V}_1 , etc.. Note that edges originating from one node $v \in \mathcal{V}_k$ may only terminate in nearby hierarchies $\mathcal{V}_k \cup \mathcal{V}_{k+1} \cup \mathcal{V}_{k+1}$. In other words, edges in the scene graph exclusively connect nodes within the same hierarchical level, or one level higher or lower.

We instantiate the graph nodes with the instance segmentation from the point cloud and parameterize each node with object centroid $p_i \in \mathbb{R}^3$ and size of the bounding box. Next, we traverse all the nodes to determine their spatial relationships. In addition, we utilize an automatic verification procedure to validate the scene graph, further improving the quality of the scene graph we constructed. One of the verification operations involves manually maintaining a mapping between objects and relationship descriptions based on common sense. For example, people usually use “mounted on” to describe the relation between TV and wall, rather than “hanging on.”

To get detailed attributes of an object’s visual and physical properties, we utilize the object captioning pipeline outlined as follows. Given the multi-view images, we use the point cloud of the object to get the visible points in the images through rendering. The image is then cropped with the rendered bounding box and processed through BLIP2 [40] to generate information about the object color, shape, material and affordance, etc. For the attributes from every image, we calculate its CLIP [74] similarity score between the text and the cropped image and select the top 10 with the highest CLIP score and minimal occlusion. The selected attributes are fed into a LLM to obtain a coherent summary for the object. In this process, we explicitly instruct the language model to identify and correct the potential errors.

C. System Design

Task Decomposition Our system decomposes the complex tasks into intermediate steps and allocates tools to complete them. We prompt the system for task decomposition, which are demonstrated from our prompt template in Fig. A.3a and Fig. A.3b. The first three lines in Fig. A.3a correspond to the prologue of this prompt that instructs the agent to rely on the tools for responding to a user’s inputs. The prologue is followed by four in-context examples of plans. Note that in these plans we not only specify the tools to be used but also the reasons for selecting these tools.

TABLE A.1: Relationships in 3D Scene Graph (3DSG).

Category	Relation	
Vertical Proximity	support	embed
	hanging on	inside
	mounted on	affixed on
	below	above
Horizontal Proximity	higher than	lower than
	near	far
Allocentric	besides	next to
	left	right
	behind	is in front of

Fig. A.3b includes an answer format section that illustrates the syntax for calling tools and the sequential process of resolving a user input and several rules.

Observation For LLMs to comprehend information in 3DSGs, we render the information retrieved from actions using templates. Specifically, observations are organized on the basis of objects. Observation for each object can be rendered from three templates. The first one is for position and size: “The position of the {object.name} (id: {object.id}) is {object.position}.”. The second one is for attributes: “The {object.name} (id: {object.id}) has attributes: {object.attributes}.”. The third one is for relations: “The {object.name} (id: {object.id}) is {relation} {name_id_list}.”, where “{name_id_list}” is a list of strings in the format of “{name} (id: {id})” generated for objects that have the relation of interest to the object. To save up tokens, the third template is used only for tools that reason about relations, such as `query_for_relations`. When a tool retrieves multiple objects, we iterate through the objects, render strings, and then concatenate the strings to form the observation for the tool.

Human-Robot Interaction As mentioned in Section III-B, to facilitate human-robot alignment in the presence of perception errors, we design a GUI that allows users to freely interact with the scene by dragging, zooming-in/out, changing views, marking objects and asking free-form questions at will. Furthermore, it provides the potential to combine language-based interaction with object clicking for object referencing. Fig. A.2 shows an example, in which the user inspects and corrects the color of a door. In the GUI, the user selects the object by clicking on a point. After a point is clicked, we compute a 3D ray using the point (i.e. a point in the viewing plane) and the user’s current viewing angle. We take the first object whose meshes intersect with this 3D ray as the marked object and display its bounding box in the GUI, which is the yellow box around the door in Fig. A.2.

Tools The tools are a set of Python functions designed for interacting with 3DSGs and the users. When generating the overall step-wise prompt, LangChain extracts the docstrings of tools and injects them into the prompt, so the docstrings play a vital role in implementing tool usage. We use

a fixed format for the doc-strings and show an example in Fig. A.4. The doc-string starts with a brief description of the tool’s functionality (lines 3 and 4). The “Hints” section specifies expected behaviors. For example, the 1th hint says that the ambiguity of objects should be resolved by the user. If one changes this hint to “....., call this tool for each of the candidates”, then all of the objects related to the user input will be altered. The following three hints dictate constraints on the input argument, `new_attributes_list`. In this case, we expect the LLM to update one or more attributes of an object but keep the rest of the attributes unchanged. The “Argument” and “Return” sections are introduced to facilitate tool calling.



Fig. A.2: A screenshot for our interface. A user can select the scene to interact using the drop-down menu located at the upper left. The left part consists of the reconstructed view, the local 3DSG for the object of interest (bottom left), and object segmentation (bottom middle). The user can chat with our system using the input box located in the middle right. It can also select an object by clicking it in the reconstructed view or the 3DSG. In this example, the user marks the door.

II. EXPERIMENTS

A. Setup

We utilize the reconstructed meshes provided by the dataset for our experiments on the ScanNet [55], which are reconstructed by depth fusion and marching cubes [54]. For real-world robot execution, we utilize MonoSDF [56] to obtain the 3D scene reconstruction. We utilize 3D-VisTA [58] to get the instance segmentation for all the experiments.

B. Alignment Tasks

We list all the alignment tasks from ScanNet in Table A.2. Based on the number of objects involved with perception misalignment, the tasks are divided into *EASY* ($\text{misalignment}=1$) to *HARD* ($\text{misalignment}>1$).

We engage the participation of 10 human subjects for our alignment experiment, each assigned tasks across 3 scenes. For the ablation study, the subjects are allowed to use the GUI but are instructed to refrain from using mouse clicks. We randomly select 3 participants from the 10 and assign each 2 scenes randomly from the task pool to report the quantitative evaluations.

You are an assistant agent in a room and you have to respond to input from a user. When the user ask for information of items in the room, collect the required information using your tools. When the user ask you to learn something, use your tools to do so. When the user ask you for advices or actionable plans, make up one with your own knowledge and ground the plan to the room using your tools.

Make a rough plan first. Each step of this plan corresponds to a single action. State the plan before your first action.

[start of example plan]

User Input: Find the bed.

Plan:

1. Use query_related_objects to extract candidate items that could correspond to the bed.
2. If you can identify the bed, then include information of the bed in your response. Otherwise, in your response, ask the user for clarification.
3. Use post_process.
4. Use Final Answer to return your response to the user.

User Input: What is to the left of the marked item?

Plan:

1. Use find_marked_object to locate the marked item.
2. Use query_for_relations to gather information of spatial relationship between the marked item and other items, then determine the all items that are to the left of them marked item.
3. Use post_process.
4. Use Final Answer to return your response to the user.

User Input: What is in the middle of the bed and the desk?

Plan:

1. Extract the coordinates of the bed and the desk by using query_for_items_info.
2. Use calculate_mid_point to compute the mid point of the coordinates of the bed and the desk.
2. Use find_object_closest to find the item that is closest to the mid point of the bed and the desk.
3. Use post_process.
4. Use Final Answer to return your response to the user.

User Input: I want to fall asleep in a warm and dark environment. What should I do?

Plan:

1. Use query_related_objects to find all items that are related to [fall asleep warm dark]. Compose a imaginary plan as if you could operate the related items.
2. Use post_process.
3. Use Final Answer to return your imaginary plan to the user. Remember to include the [id] and [location] of related items in your plan.

[end of example plan]

(a) The first part of our prompt template.

At each step, select an action, analyze your observation, and determine your next action based on the observations you received so far, the user's request, and your original plan. Follow this format.

[start of answer format]
Action: `'''{"action": [selected action], "action_input": [action input]},'''`
Observation:
Thought: [thought]
Action: `'''{"action": [selected action], "action_input": [action input]},'''`
Observation:
... (repeat Thought/Observation N times)
Thought: [thought]
Action: `'''{"action": post_process, "action_input": [action input]},'''`
Observation:
Thought: [thought]
Action: `'''{"action": Final Answer, "action_input": [final response]},'''`
...
[end of answer format]

Rules:

1. Be sure to include an Action in your response.
 2. For [selected action], use one tool from [{tool_names}].
 3. For [action input], refer to the descriptions of the tools.
 4. When choosing Final Answer, do not format [final response] as a dict. Use a sentence in natural language for [final response].
 5. Be sure to choose Final Answer at the last step.
 6. For [thought], include your reasoning for choosing the next step.
 7. State your plan before the first action.
- ```

(b) The second part of our prompt template.

Fig. A.3: Our prompt template. This template will be combined with doc-strings of tools, latest observations, and historical information.

```

1 def update_object_attributes(
2     new_attributes_list: List[str], object_id: str, **kwargs) -> dict:
3     """Update the attributes of a specific object. Use this tool when the user
4     ask you to change some of the attributes of an object.
5
6     Hints:
7         1. If there are multiple candidates for object_id, do not call this tool
8             and ask the user for clarification.
9         2. The attributes of an object is a list of strings that specify the color,
10            the texture or other information of an object. Before using this tool, use
11            query_for_objects to get the current attribute list of the object under
12            consideration.
13         3. To prepare the input argument new_attributes_list, start with the
14            current attributes list. Replace the corresponding elements in the current
15            attributes list with new values specified by the user. For example, if the
16            current attributes list is ["blue", "wooden", "rectangular"] and the user
17            asks you to change the color to red and the shape to triangular, then the
18            new attributes list should be ["red", "wooden", "triangular"].
19         4. Do not alter the values of the attributes not mentioned by the user.
20
21     Arguments:
22         new_attributes_list: list, the new list of attributes to be assigned to the
23             object, eg. ["blue", "wooden"]
24         object_id: str, the id of the object to be updated, eg. "1".
25
26     Return:
27         observation: str, information about the object under consideration.
28         results: list, a list that contains the object under consideration.
29     """

```

Fig. A.4: An example for the doc-strings of tools.

TABLE A.2: Evaluation tasks designed for the ScanNet dataset. In our experiments, we also provide participants with images for objects involved in each task, so that they can refer to the objects by clicking.

Scene	Task ID	Category	Difficulty	Task	Expected Answer
scene0011_00	t1	Attribute	Easy	What is the tv made from?	Plastic.
	t2	Spatial	Hard	What is the item that is above the stove?	Stove hood.
	t3	Numeric	Hard	How many tables are there in the room?	Two.
scene0050_00	t1	Attribute	Easy	Is the blue box triangular in shape?	No.
	t2	Spatial	Easy	What is the item above the blue box?	Toolbox.
	t3	Spatial	Hard	What is the item's name for sitting and to the right of desk and behind the door?	Sofa.
	t4	Spatial	Hard	What is the item above the desk and beside the laptop?	Printer.
	t5	Spatial	Hard	What is the item in front of the piano?	Piano bench.
	t6	Numeric	Easy	How many chairs are there in the room?	One.
scene0169_00	t1	Attribute	Easy	What is the color of the plastic trash can?	Gray.
	t2	Attribute	Hard	What is the partition made from?	Glass.
	t3	Attribute	Hard	Is the red backpack oblong in shape?	No.
scene0342_00	t1	Attribute	Easy	What is the color of the screen?	Black.
	t2	Attribute	Easy	What is the color of the desk that is lower than the screen?	Brown.
	t3	Spatial	Hard	Is the red backpack placed on the black table?	Yes.
	t4	Spatial	Easy	Is the red backpack hung on the wall?	No.
scene0355_00	t1	Attribute	Easy	What is the microwave oven made from?	Stainless steel.
	t2	Attribute	Hard	Is the armchair round in shape?	No.
	t3	Attribute	Easy	Is the armchair made from wood?	No.
	t4	Numeric	Hard	How many tables are there in the room?	Two.
	t5	Numeric	Hard	How many chairs are there in the room?	Eight.
scene0356_00	t1	Attribute	Easy	What is the color of the dresser?	Shallow yellow.
	t2	Attribute	Easy	Is there any white recycling bin in the room?	Yes.
	t3	Attribute	Easy	Is the black chair rectangular in shape?	No.
	t4	Attribute	Easy	Are the shelf and the desk of the same color?	Yes.
	t5	Numeric	Hard	How many doors are there in the room?	Two.
scene0389_00	t1	Attribute	Easy	What is the color of the door?	White.
	t2	Spatial	Hard	Is the hanger located to the left of the ironing board?	Yes.
	t3	Spatial	Hard	What is to the left of the cabinet on which a TV is resting?	Include a refrigerator.
	t4	Attribute	Easy	Are the two doors in the same color?	Yes.
scene0406_00	t1	Attribute	Easy	What is the shape of the door?	Rectangular / oblong.
	t2	Attribute	Easy	What is the white sink made from?	Ceramic.
	t3	Attribute	Easy	Are the sink and the bathtub made from the same materials?	Yes.
scene0427_00	t1	Spatial	Hard	What is behind the trash bin?	A glass partition and a door frame.
	t2	Numeric	Easy	How many tables are there in the room?	One.
	t3	Numeric	Hard	How many chairs are there in the room?	Four.
	t4	Numeric	Hard	How many doors are there in the room?	One.
scene0144_00	t1	Attribute	Easy	What is the white rectangular dresser made from?	Wood or metal.
	t2	Attribute	Easy	What is the monitor made from?	Plastic.
	t3	Spatial	Easy	What is item on top of nightstand?	Printer.
	t4	Spatial	Easy	What is item supporting the monitor?	Small desk.
	t5	Numeric	Easy	How many dressers are there?	Two.