Details Make a Difference: Object State-Sensitive Neurorobotic Task Planning*

Xiaowen Sun**, Xufeng Zhao, Jae Hee Lee, Wenhao Lu,
Matthias Kerzel, and Stefan Wermter

Knowledge Technology, Department of Informatics, University of Hamburg
{xiaowen.sun, xufeng.zhao, jae.hee.lee, wenhao.lu,
matthias.kerzel, stefan.wermter}@uni-hamburg.de
www.knowledge-technology.info

Abstract. The state of an object reflects its current status or condition and is important for a robot's task planning and manipulation. However, detecting an object's state and generating a state-sensitive plan for robots is challenging. Recently, pre-trained Large Language Models (LLMs) and Vision-Language Models (VLMs) have shown impressive capabilities in generating plans. However, to the best of our knowledge, there is hardly any investigation on whether LLMs or VLMs can also generate object state-sensitive plans. To study this, we introduce an Object State-Sensitive Agent (OSSA), a task-planning agent empowered by pre-trained neural networks. We propose two methods for OSSA: (i) a modular model consisting of a pre-trained vision processing module (dense captioning model, DCM) and a natural language processing model (LLM), and (ii) a monolithic model consisting only of a VLM. To quantitatively evaluate the performances of the two methods, we use tabletop scenarios where the task is to clear the table. We contribute a multimodal benchmark dataset that takes object states into consideration. Our results show that both methods can be used for object state-sensitive tasks, but the monolithic approach outperforms the modular approach. The code for OSSA is available at https://github.com/Xiao-wen-Sun/OSSA

Keywords: Object State Identification · Artificial Intelligence · Robotics · Language Models · Multimodality

1 Introduction

Object attributes such as color, shape, and size play important roles in shaping the diverse states objects can exhibit, significantly impacting our daily interactions. Understanding and managing these states is crucial as overlooking them can result in unforeseen consequences. Humans intuitively rely on their understanding of object states and common sense to interact with everyday objects.

* Published in the Proceedings of the International Conference on Artificial Neural Networks, 2024.
** Corresponding author

However, integrating state-sensitive knowledge poses a significant challenge for robotic systems. Enumerating and seamlessly integrating this nuanced understanding into robotic systems remains a complex endeavor.

Recent approaches that leverage Large Language Models (LLMs) [39] have shown promising results in tasks that require human-level commonsense knowledge. Such approaches use LLMs' commonsense reasoning ability to interpret natural language goals [41], such as 'put one apple into the fridge.' According to those goals, LLM generates the suggested action plan, 'next actions: pick up the apple, move to the kitchen, open the fridge, ...' However, the object's physical state (e.g., intact apple, sliced apple, etc.) is crucial yet less considered in the task planning [26]. To the best of our knowledge, there is hardly any research that leverages a pre-trained neural network (e.g., LLM or VLM) to address the integration of object states into planning tasks for household robots. To address this gap, we introduce an **O**bject **S**tate-**S**ensitive **A**gent (**OSSA**), which utilizes commonsense reasoning of pre-trained neural networks for robot task planning.

We pose several real-world challenges in object state-sensitive agent (OSSA) task planning. First, in order to solve the tasks, the agent does not only involve identifying different objects in the scene but also distinguishes between their *states*. For example, in the 'clear the table' [19,9] task, a robot needs to be able to distinguish between *whole* and *sliced* fruit, and between *clean* and *dirty* plates. This is a challenge because existing state-of-the-art object detection models often fail at differentiating between objects in different states. A plan lacking state-sensitive awareness may lead to unexpected results.

Second, the agent needs to employ commonsense reasoning for taking *state-sensitive* actions that correspond to the object states of various scenarios instead of asking the users for an exhaustive design or an intervention. For example, whole fruit go directly into the fridge or to the cupboard, while sliced fruit can either go into the fridge or be discarded into the trash bin if they are regarded as leftovers. One way to solve the commonsense reasoning problem is using rule-based symbolic approaches [19], but by design, they do not generalize to situations outside their rule base, i.e., they cannot handle new objects and states. Commonsense reasoning with a data-driven model trained on a large dataset that generalizes well (e.g., a large language model [31]) is, therefore, a more viable choice.

Third, the robot needs to identify cases where common sense should not dominate. For example, a robot has to take into account the preferences of the user when handling specific objects in specific states [36]. In the table clearing example, different users might handle the leftover food differently, e.g., some might prefer to discard the leftovers, while others are more frugal and prefer to keep the leftovers for the next meal. The robot, therefore, needs to detect situations where different user preferences come into play, and has to ask the user for clarification instead of arbitrarily choosing one on its own. Existing approaches, however, do not take user preferences into account [28,24].

In this paper, we study the problem of state-sensitive instruction following. We investigate two different methods, (i) a modular model consisting of an ob-

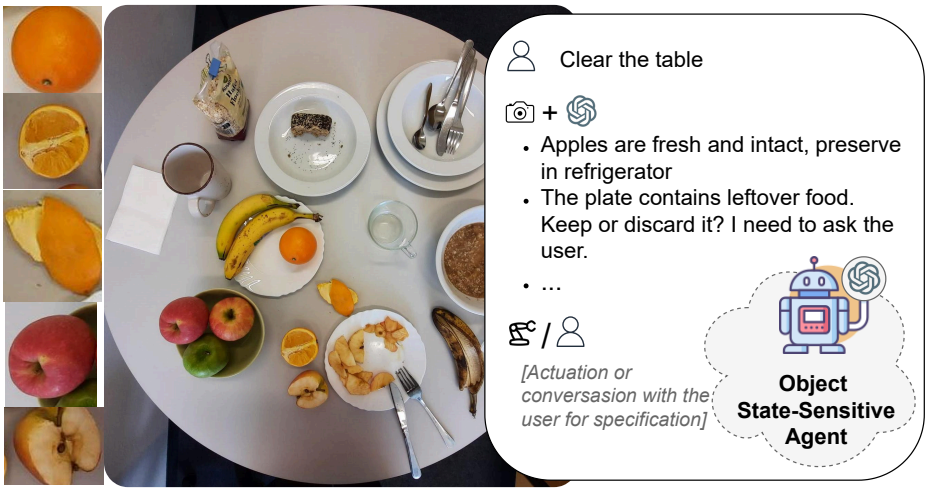


Fig. 1: The given scene contains various objects in various states. For example, orange, half-orange, and orange peel; clean napkin and dirty napkin; banana and banana peel. Based on commonsense knowledge, the agent sorts the objects (discard the banana peel in the trash bin; keep the bananas in the cupboard). However, the robot is not able to decide how to deal with the leftover food because different people may have different preferences regarding leftover food (e.g., half orange and half bread).

ject detection module and an LLM, and (ii) a monolithic VLM-only model to caption and reason universally. We use cluttered tabletop scenarios (in Fig. 1) as a specific application of our approaches, where the task is to clear the table based on the object's attributes, states, and user preferences. We summarize our contributions as follows:

1. We introduce an object state-sensitive agent (OSSA) that can perceive fine-grained scene information (objects and their states) and generate an appropriate object manipulation plan for the robot's low-level executor to fulfill a given task.
2. To explore a pre-trained neural network's capability for object state-sensitive task planning, we propose two methods for OSSA: a modular approach consisting of dense captioning model (DCM) and LLM modules and a monolithic approach consisting only of a VLM.
3. To evaluate the proposed methods, we formulate an instruction-following task for robots that focuses on the objects' attributes and states in a tabletop scenario.
4. We provide an open-source benchmark dataset containing various object states, which involves 40 scenarios and 184 objects in total.

2 Related Work

2.1 Vision and Language Models for Task Planning in Robotics

Recently, vision-language models (VLMs) [31,32] and large language models (LLMs) [39] have transformed robot task planning, with the integration of LLMs' commonsense knowledge into planning for embodied agents garnering growing interest [34,11,3,27,23,40,26,41].

However, these methods assume a singular state for objects in their plans, which may not suit the variability of the domestic settings we study. The most high-performance perception frameworks (e.g., OWL-ViT [17], OWL-V2 [18], YOLO [4], CLIP [22], etc.) that the existing work is using [42] are not trained to distinguish the object states, e.g., half apples, apple, dirty plate, and clean plate. Thence, we conclude that these approaches are not sensitive to object states.

One of the closely related works to ours is TidyBot [36], a system that enables a domestic cleaning robot to adapt to individual user preferences in managing objects, recognizing that each item may require a unique approach based on personal taste. They leverage the summarization capabilities of LLMs to get the user's preference for sorting the objects, e.g., 'put light-colored clothes in the drawer and dark-colored clothes in the closet'. In our work, we take advantage of commonsense knowledge from LLMs to sort the objects and detect when human preference needs to be involved in automation tasks.

In other research, VILA [10] utilizes GPT-4V(vision) as VLM to do task planning, which is also close to our work. However, the difference is that they focus on long-horizon robotic tasks, where the manipulation actions and goals can be derived from the user instruction. In our work, we focus on automation tasks where the robot manipulation goals stem from visual perception. Then our model performs reasoning and generates a manipulation plan, according to the visual perception.

2.2 Benchmarks for Task Planning in Household Robots

Gutman et al. [9] use the `clear the table` as an experiment task to explore the users' preference for different autonomy levels of an assistive robot. The results show that the participants prefer the highly autonomous assistant. Other research uses `table clearing task` to study the robot to understand and execute humans' natural language instructions [19]. However, overall, their work does not mention that the objects from the same category may be in a different state, which would change the robot's action.

From the computer vision research perspective, two datasets [13] focus on object state recognition in cooking, e.g., whole, peeled, floured, etc. The Object State Detection Dataset [7] involves the object states: open, close, empty, containing something liquid, containing something solid, plugged, unplugged, folded, and unfolded. However, it does not consider the leftover food in our daily life. From the robot kinematics research perspective, many benchmarks [2,43,12,29]

and methods have been proposed to enable robots to complete certain manipulation tasks, e.g., picking, wiping, dragging, pushing, pouring, and placing. We focus on automated task planning and research how to utilize these enabled robot skills to complete certain complex tasks. For example, the robot, according to various object states, generates plans for a robot low-level executor that receives a specification of how to 'pick' and where to 'place' the target.

3 Methodology

3.1 Architecture

We introduce an Object State-Sensitive Agent (OSSA) that can perceive fine-grained scene information (object name and state, etc.) and generate an appropriate object manipulation plan for the robot's low-level executor according to those visual perceptions. The pseudocode of the system architecture is provided in Algorithm 1. OSSA generates object manipulation plans when an utterance (U) and an image of the table (I) are given. A chat system is for ambiguous situations. If uncertainties arise (e.g., encountering a peeled pear on the table), the robot will ask the user for disambiguation. Lastly, a low-level executor will execute language-conditioned instructions, such as picking and placing for fulfilling the tasks. We assume that the low-level executor has these skills, which can be acquired via machine learning [21] approaches (e.g., reinforcement learning [15,1] and imitation learning [12]) or direct hard-coding.

3.2 Object Manipulation Plan Generation

To explore the best way to employ commonsense knowledge [41] and reasoning [11] capabilities of pre-trained neural networks (e.g., LLMs or VLMs) for object sensitive-state agent task planning (OSSA), we propose two categories of methods for object manipulation plan generation. The input of the method is a scene (I) and a user's utterance (U). We prompt LLMs to generate structured responses in a JSON-like format, 'object name': { 'color', 'size', 'shape', 'container', 'state', 'grasping type', 'destination', 'placing type' }. 'Color', 'size', and 'shape' are the object attributes that are visible in the scene. The object's physical 'state' is visible in the scene. However, reasoning the human-defined object state requires commonsense knowledge. 'Grasping type' is an action that informs the robot how to grasp the target at the initial place. 'Destination' is a place that informs the robot where the target should be stored. 'Placing type' is an action that informs the robot what it should do at the destination. Those three items are from commonsense knowledge reasoning. For object state detection, a computer vision system dense captioning [14] can localize salient regions in images and use natural language to describe them. The natural language description of a salient region includes the condition of the object in the region. Therefore, state-of-the-art dense captioning can be utilized in our study. However, the output of the dense captioning model does not include 'grasping type',

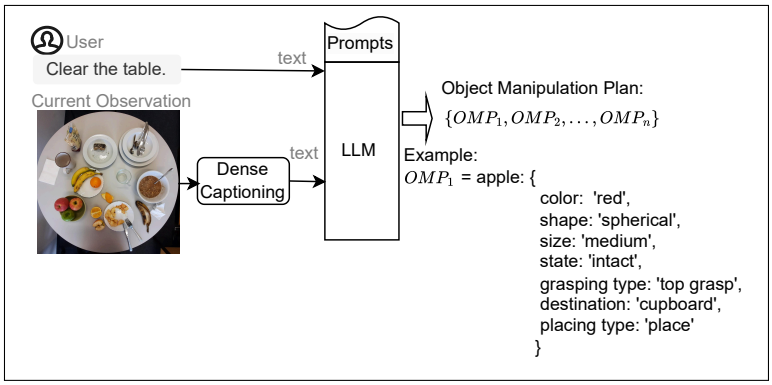
Algorithm 1 System architecture

```
r ← Robot.Initialize()
commands = NULL
while True do
    U ← r.GetUserUtterance()
    I ← r.GetImageofTable()
    {OMP_1, ..., OMP_n} ← r.OSSA(U, I)
    for OMP_i in {OMP_1, ..., OMP_n} do
        if OMP_i.destination is "uncertain" then
            AI_Response ← r.ChatSystem(OMP_i)
            U ← r.GetUserUtterance()
            OMP_i ← r.ChatSystem(U)                    # ChatSystem revise OMP_i
            commands.append(OMP_i)
        else
            commands.append(OMP_i)
        end if
    end for
    if commands not NULL then
        r.LowLevelExecutor(commands)
        commands = NULL
    end if
end while
```

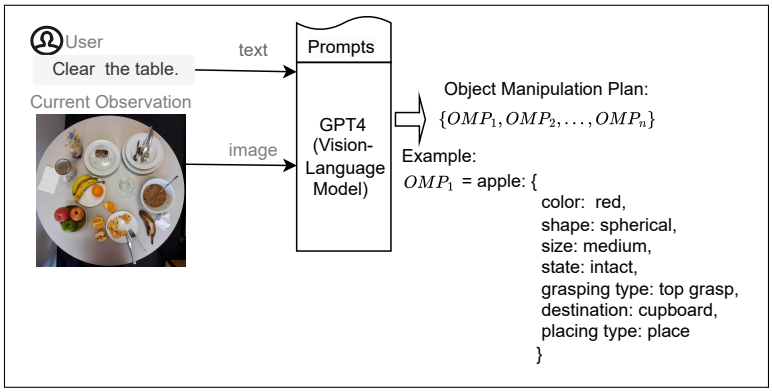
'placing type', and 'destination'; another module with commonsense knowledge and reasoning abilities is needed to generate them. GPT-4 [37,20] can not only be a visual perception model but also perform as a vision-language model. In this study, we utilize both of the abilities of GPT-4. We propose two methods: i) a modular model consisting of a vision processing module (dense captioning model) and natural language processing model (LLM), and (ii) a monolithic VLM-based model.

Modular Method A modular pipeline [3,40,23,27,26,11,36] combines two models, e.g., vision and language. First, a vision detection model (e.g., YOLO [4], ViLD [8], OWL-ViT [17], OWL-V2 [18], etc.) detects the objects in the scene. Second, a large language model processes the user's instruction and the output of the vision model. Those vision models are trained to detect the abstract object category. Therefore, we cannot use their methods directly.

The dense captioning model does not only localize salient regions in images but also uses natural language to describe them [14]. Instead of object detection models, we use a dense captioning model to get the description of salient regions of a scene as text. As Fig. 2a shows, a prompt LLM processes a user instruction and dense caption. The quality of the dense captions is an important factor influencing the model's performance. We choose the state-of-the-art dense captioning model GRiT [35]. Additionally, GPT-4V was also successfully used for image or video understanding [16,31,30]. In this work, we use both GPT-4V and GRiT to generate the dense caption for a given image.



(a) Modular Method: OSSA-LLM-DCM.



(b) Monolithic Method: OSSA-VLM

Fig. 2: Overview of our two proposed methods for OSSA: (a) OSSA-LLM-DCM represents the modular model that combines a prompt large language model (LLM) and a dense captioning model (DCM); (b) OSSA-VLM represents only a vision-language model (VLM).

Monolithic Method As shown in Fig. 2b, the user's instruction U and image of a table I are the input of a monolithic model. The model's output is object manipulation plans: $\{\text{OMP}_1, \text{OMP}_2, \ldots, \text{OMP}_n\}$. Yang et al. [37,20] are given preliminary explorations with GPT-4V(ision). GPT-4V has also shown remarkable results for game action prediction in the simulation environment [6,5]. The approach VILA by Hu et al. [10] unveils the capability of GPT-4V for long-horizon robotic planning to generate a sequence of actionable steps. The difference from those works is that we leverage GPT-4V to recognize the object state and generate a manipulation plan for an object according to its state.

Prompts for Object State Detection and Reasoning LLMs and VLMs are pre-trained on massive datasets from a diverse set of sources, which allows

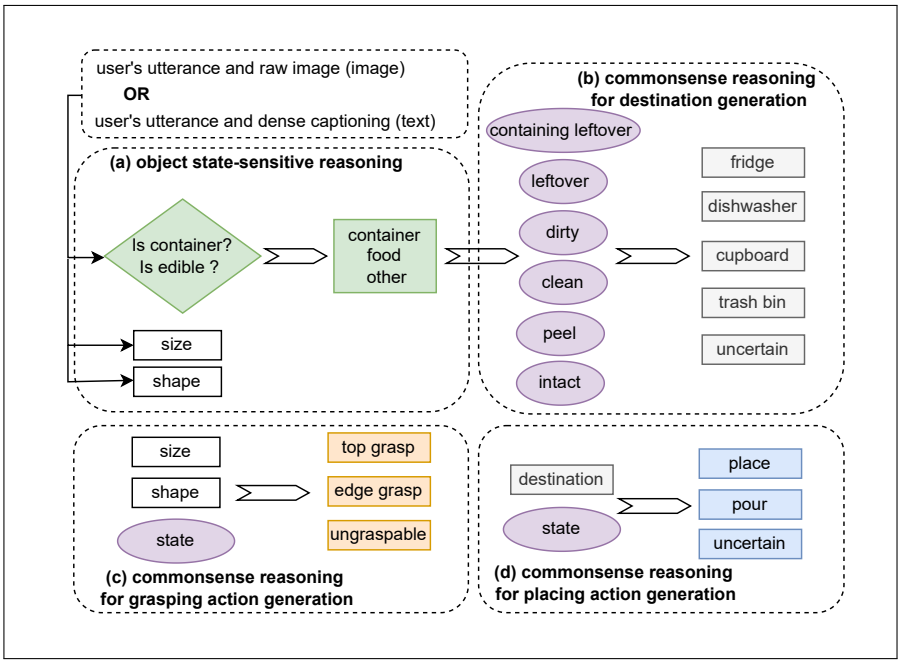


Fig. 3: Chain-of-thought for OSSA. (a) The pre-trained model (e.g., LLM or VLM) utilizes commonsense knowledge to reason about the object state; (b) according to the object's state and user's preference, the model generates a destination for the object; (c) according to the object's state, shape, and size, the model generates a grasping action for the object; (d) according to the object's state and destination, the model generates a placing action for the object.

them to acquire human-level commonsense knowledge for handling object states. To unleash their full potential, however, it is necessary to devise appropriate prompts and guide them; otherwise, their generation cannot be directly used for robot tasks [3,24]. In this study, we utilize an LLM or a VLM as a high-level planner for the robot's low-level executor. Fig. 3 shows a schematic chart of the chain-of-thought [33,38] that both models share, which guides them to recognize the objects' states and plan the objects' destinations, grasping types, and placing types.

4 Experiments

To quantitatively evaluate the performance of the two proposed methods, we formulate tabletop scenarios where the task is to clear the table (in Sec. 4.1). We contribute a multimodal benchmark dataset according to these scenarios that take object states into consideration (in Sec. 4.2). We then evaluate our proposed methods on the dataset (in Sec. 4.3).

4.1 Task Formulation

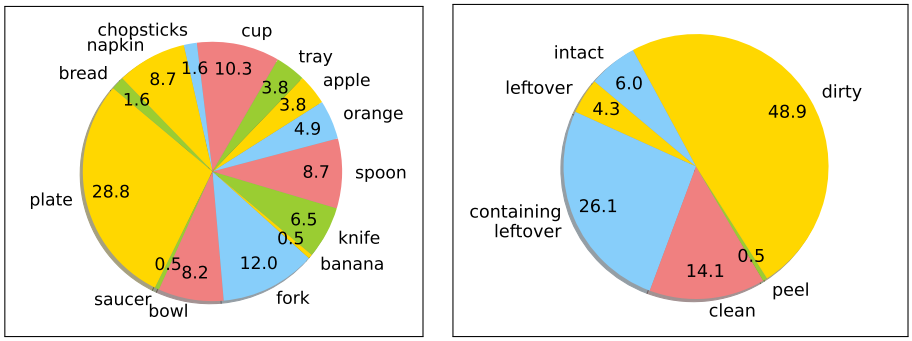
In our study, we identify two types of object states about leftovers: "containing leftover food", where containers like plates or bowls hold liquid or semi-fluid contents that are not directly manipulable by the robot (e.g., a bowl filled with leftover soup), and "leftover food", referring to food remains that have been sliced or peeled, which robots can handle.

According to these specified leftover types, we consider three different common scenarios: T1) The instruction is "clear the table" without specifying what to do with the leftover food. In this scenario, the robot is supposed to generate a manipulation plan for all the objects except those classified as "leftover food" or "containing leftover food". In this case, the expected behavior of the robot is to ask the user for handling specifications about the leftovers. T2) The instruction is "clear the table and keep all the leftover food". In this scenario, if the object state is "leftover food" or "containing leftover food", the robot should store it in the fridge. T3) The instruction is "clear the table and discard all the leftover food". In this scenario, if the object state is "leftover food", the robot is supposed to throw it into the trash bin. However, if the object state is "containing leftover food", the robot is supposed to grab the container (not the soup directly) and pour the contents into the trash bin before putting the container into the dishwasher.

4.2 Benchmark Dataset

To quantitatively evaluate the methods that we propose, we built a benchmark dataset, which is composed of 40 scenes involving 184 objects. First, we sourced 27 scenes from our daily lives. Second, to create balanced data, we use a diffusion model [25] to generate an additional 13 scenes. Fig. 4a shows all the objects that are involved in this dataset and the proportion among them. The same object may be in different states at different times. The object states that we consider, and their proportion are shown in Fig. 4b.

Annotation Rules The object manipulation plan format, outlined in Sec. 3.2, was employed for annotating the dataset. We asked two individuals to label the dataset according to set rules to reduce bias. In the end, they collaborated to resolve discrepancies and finalize the annotations.
Object name: use the object name as a JSON key. When more than one item from the same object category is in the scene, add the number behind the name (e.g., 'plate 1', 'plate 2'); **Color**: the object color (e.g., 'white', 'silver', 'orange', 'red'); **Size**: the object size, 'small', 'medium', 'big'; **Shape**: the object shape, 'elongated', 'irregular', 'oval', 'round', 'spherical', 'cylindrical', 'rectangle'; **Container**: use 'true' or 'false' to label the object as a container or not; **State**: If the object is a container, we label it with three states: 'clean', 'dirty', or 'containing leftover food'. If the object is not a container but edible, we label it with three states: 'intact', 'peel', or 'leftover food'. If the object is not a container and



(a) Distribution of Objects(%). (b) Distribution of Object States(%).

Fig. 4: Dataset Statistics

inedible (fork, knife, spoon), we label it with two states: 'dirty' or 'clean'; **Destination**: we use four places: 'trash bin', 'fridge', 'cupboard', and 'dishwasher'; **Grasping type**: we set two types of grasp action: 'top grasp' or 'edge grasp'; **Placing type**: In most cases, the robot places the object in the destination. But in the special case that the object is a container that contains leftover food, the robot should pour the leftover food into the trash bin.

Evaluation In this study, we aim to generate a manipulation plan for the objects to the robot's low-level executor. The main evaluation metric is accuracy. There are five parts of the output needed for low-level execution. We evaluate our proposed methods' performance in those five parts: 1) State Detection Accuracy (**StaA**), 2) Ambiguous Detection Accuracy (**AmbA**), 3) Destination Generation Accuracy (**DesA**), 4) Grasping Type Generation Accuracy (**GraA**), and 5) Placing Type Generation Accuracy (**PlaA**). Finally, we calculate one overall accuracy, representing how many objects are being predicted correctly, Completion Accuracy (**ComA**).

4.3 Results

First, we test the performance of our proposed methods on object state detection. For the modular method, we prompt the GPV-4V(ision)[1] as a dense captioning model to get the description of the image. GRiT is another dense captioning model that we used in our experiment. We prompt an LLM (GPT-4[2]) to process the image description (text) to abstract the object states. For the monolithic method, we prompt GPV-4V(ision) as a vision-language model to directly detect the object state. We prompt the pre-trained model in zero-shot or

[1] gpt-4-vision-preview
[2] gpt-4-0125-preview

Table 1: Object State Detection Average Accuracy(%)±Standard deviation

Method	Result
OSSA-LLM(Zero-shot)-GRiT	55.61±5.05
OSSA-LLM((Few-shot)-GRiT	55.69±4.76
OSSA-LLM((Zero-shot)-GPT-4V	73.47±1.39
OSSA-LLM((Few-shot)-GPT-4V	74.77±1.37
OSSA-VLM (Zero-shot)	**75.14**±0.37
OSSA-VLM (Few-shot)	**79.83**±1.00

few-shot settings, respectively. Therefore, we evaluate six variants: OSSA-LLM-GRiT with the zero-shot setting, OSSA-LLM-GRiT with the few-shot setting, OSSA-LLM-GPT-4V with the zero-shot setting, OSSA-LLM-GRT-4V with the few-shot setting, OSSA-VLM with the zero-shot setting, and OSSA-VLM with the few-shot setting.

As Tab. 1 shows, the monolithic method with few-shot prompts achieves the highest performance compared to the other five variants. Overall, the modular model that combines LLM with GRiT performs worse than the other four variants. However, the advantage of this model is the robot does not need an extra object detection model to determine the object's location. The model that combines LLM with GPT-4V is the most expensive in this experiment because it calls two pre-trained models for one plan generation. The performance is worse than OSSA-VLM with GPT-4V and does not supply the object's location. From the results of OSSA-VLM and OSSA-LLM-GPT-4V, we conclude that GPT-4 performs well as a vision-language model that plans multimodal tasks. The potential vision information will be lost after the image is converted to text descriptions. For example, spatial reasoning and object attribute understanding are no longer possible.

Second, besides detecting the object state, we prompt the pre-trained LLM and VLM to generate the object manipulation plan consisting of 'grasping type', 'destination', and 'placing type'. Based on the previous experiment's results, we conclude that GPT-4 functions more efficiently as a VLM in multimodal tasks. In this experiment, we test two methods: OSSA-LLM-GRiT and OSSA-VLM. We also prompt methods in zero-shot or few-shot settings respectively. Hence, we evaluate the four variants in three tasks which are defined in Sec. 4.1.

The object state detection accuracies shown in Tab. 1 and Tab. 2 are different. When asking for more reasoning items from the pre-trained models (e.g., LLM or VLM), the model's performance decreases. Similarly, in certain cases humans perform better when focusing on a single task rather than on multiple tasks. Overall, in those three tasks, the monolithic method OSSA-VLM performs best in ambiguity detection, destination generation, and completion rate. The few-shot prompts also enhance the performance of models in ambiguity detection and destination generation items. We notice outliers in the grasping and placing action generation. For destination generation (**DesA**), the few-shot monolithic method performs much better than the modular method in the three tasks. For

Table 2: Average Accuracy of Object Manipulation Plan Generation. OSSA-L(Z)-G represents zero-shot OSSA-LLM-GRiT, OSSA-L(F)-G represents few-shot OSSA-LLM-GRiT, OSSA-VLM(Z) represents zero-shot OSSA-VLM, OSSA-VLM(F) represents few-shot OSSA-VLM, State Detection Accuracy (StaA), Destination Generation Accuracy (DesA), Grasping Type Generation Accuracy (GraA), Placing Type Generation Accuracy (PlaA), Completion Accuracy (ComA), Ambiguous Detection Accuracy (AmbA), '-' means that the tasks T2 and T3 do not have ambiguity.

Task	Method	Average accuracy(%)±Standard deviation					
		StaA	AmbA	DesA	GraA	PlaA	ComA
T1	OSSA-L(Z)-G	51.09±0.94	65.78±12.54	50.37±2.62	85.83±1.44	90.74±2.72	21.74±1.09
	OSSA-L(F)-G	50.54±0.55	95.30±0.26	57.00±1.47	**92.10**±1.73	96.41±0.66	26.27±0.83
	OSSA-VLM(Z)	71.10±1.33	92.97±2.33	83.38±0.19	89.51±0.57	**97.70**±0.76	52.91±1.14
	OSSA-VLM(F)	**74.36**±1.78	**97.37**±0.07	**84.81**±1.29	83.90±1.74	93.64±0.84	**53.09**±1.12
T2	OSSA-L(Z)-G	50.18±1.13	–	59.20±2.52	88.45±0.38	90.31±3.72	22.65±0.83
	OSSA-L(F)-G	50.36±0.63	–	46.76±3.26	**92.83**±2.15	**97.84**±0.03	21.38±1.75
	OSSA-VLM(Z)	68.67±1.32	–	81.17±0.90	90.20±1.50	94.99±2.73	48.63±1.91
	OSSA-VLM(F)	**72.55**±0.97	–	**85.59**±1.01	83.72±1.64	95.48±1.34	**54.18**±1.11
T3	OSSA-L(Z)-G	50.72±0.83	–	57.54±2.44	84.26±2.70	92.85±1.26	22.83±1.44
	OSSA-L(F)-G	50.18±0.31	–	58.13±4.45	**92.06**±1.67	90.97±0.65	25.18±1.25
	OSSA-VLM(Z)	70.00±0.39	–	74.80±2.36	91.43±2.81	**94.80**±1.19	47.26±1.92
	OSSA-VLM(F)	**72.19**±0.87	–	**77.85**±2.08	84.88±0.87	91.19±1.87	**47.45**±0.83

grasping type generation (**GraA**), the modular model OSSA-LLM-GRiT with few-shot prompts performs best. We conclude that the pre-trained model reasons from the text better than from the image, which needs to consider the object's size and shape. For all the variants in the placing action generation (**PlaA**), the performance is above 90 %. From the completion accuracy (**ComA**), we can see that the few-shot monolithic method OSSA-VLM performs better than the other methods.

5 Conclusion

In this paper, we introduced an object state-sensitive agent (OSSA) that can perceive fine-grained scene information (object name and state, etc.) and generate an appropriate object manipulation plan for the robot's low-level executor according to those visual perceptions. We proposed two approaches: a modular approach consisting of vision (GRiT, or GPT-4V) and Language modules (GPT-4) and a monolithic approach consisting only of a VLM (GPT-4V). To quantitatively evaluate the two proposed approaches, we formulated an instruction-following task for robots in which the object state needs to be considered in a tabletop scenario. We demonstrated that VLM GPV-4V supplies more concrete information than the state-of-the-art dense caption model GRiT. Consequently, the monolithic approach using GPT-4V performed better than the pipeline-based modular approach consisting of the dense caption model GRiT and the language

model GPT-4. A limitation of the monolithic approach (GPT-4V) is, however, that it is not trained to generate bounding boxes of objects. An additional object detection model is needed to get the locations of objects.

In the future, we will develop a model that can distinguish between objects in different states and also localize their location. Furthermore, we will use our models in real scenarios with real robots, taking into account additional objectives such as cost and time for the creation and execution of object state-sensitive plans.

Acknowledgment

The authors gratefully acknowledge support from the China Scholarship Council (CSC) and the German Research Foundation DFG under project CML (TRR 169).

References

1. Beik Mohammadi, H., Zamani, M.A., Kerzel, M., Wermter, S.: Mixed-reality deep reinforcement learning for a reach-to-grasp task. In: International Conference on Artificial Neural Networks. pp. 611–623. Springer (2019)
2. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al.: RT-1: Robotics transformer for real-world control at scale. Robotics: Science and Systems (2023)
3. Brohan, A., Chebotar, Y., Finn, C., Hausman, K., Herzog, A., Ho, D., et al.: Do as I can, not as I say: Grounding language in robotic affordances. In: Conference on Robot Learning (CoRL). pp. 287–318. PMLR (2023)
4. Chen, A., Sharma, A., Levine, S., Finn, C.: You only live once: Single-life reinforcement learning. In: Advances in Neural Information Processing Systems. vol. 35, pp. 14784–14797 (2022)
5. Durante, Z., Huang, Q., Wake, N., Gong, R., Park, J.S., Sarkar, B., et al.: Agent AI: Surveying the horizons of multimodal interaction. arXiv preprint arXiv:2401.03568 (2024)
6. Gong, R., Huang, Q., Ma, X., Vo, H., Durante, Z., Noda, Y., et al.: MindAgent: Emergent gaming interaction. arXiv preprint arXiv:2309.09971 (2023)
7. Gouidis, F., Patkos, T., Argyros, A., Plexousakis, D.: Detecting object states vs detecting objects: A new dataset and a quantitative experimental study. In: International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. vol. 5, pp. 590–600 (2022)
8. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-vocabulary object detection via vision and language knowledge distillation. In: International Conference on Learning Representations (2022)
9. Gutman, D., Olatunji, S.A., Markfeld, N., Givati, S., Sarne-Fleischmann, V., Oron-Gilad, T., Edan, Y.: Evaluating levels of automation with different feedback modes in an assistive robotic table clearing task for eldercare. In: Applied Ergonomics. vol. 106, p. 103859 (2023)
10. Hu, Y., Lin, F., Zhang, T., Yi, L., Gao, Y.: Look Before You Leap: Unveiling the power of GPT-4V in robotic vision-language planning. arXiv preprint arXiv:2311.17842 (2023)

11. Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., et al.: Inner Monologue: Embodied reasoning through planning with language models. In: Conference on Robot Learning (CoRL). vol. 205, pp. 1769–1782. PMLR (2022)
12. Jang, E., Irpan, A., Khansari, M., Kappler, D., Ebert, F., Lynch, C., Levine, S., Finn, C.: BC-Z: Zero-shot task generalization with robotic imitation learning. In: Conference on Robot Learning (CoRL). pp. 991–1002. PMLR (2022)
13. Jelodar, A.B., Salekin, M.S., Sun, Y.: Identifying object states in cooking-related images. arXiv preprint arXiv:1805.06956 (2018)
14. Johnson, J., Karpathy, A., Fei-Fei, L.: DenseCap: Fully convolutional localization networks for dense captioning. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4565–4574 (2016)
15. Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., et al.: Scalable deep reinforcement learning for vision-based robotic manipulation. In: Conference on Robot Learning (CoRL). pp. 651–673. PMLR (2018)
16. Lin, K., Ahmed, F., Li, L., Lin, C.C., Azarnasab, E., Yang, Z., et al.: MM-VID: Advancing video understanding with GPT-4V (ision). arXiv preprint arXiv:2310.19773 (2023)
17. Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., et al.: Simple open-vocabulary object detection with vision transformers. arXiv preprint arXiv:2205.06230 (2022)
18. Minderer, M., Gritsenko, A., Houlsby, N.: Scaling open-vocabulary object detection. In: Advances in Neural Information Processing Systems. vol. 36 (2024)
19. Nyga, D., Roy, S., Paul, R., Park, D., Pomarlan, M., Beetz, M., Roy, N.: Grounding robot plans from natural language instructions with incomplete world knowledge. In: Conference on Robot Learning (CoRL). pp. 714–723. PMLR (2018)
20. OpenAI: GPT-4V(ision) system card (2023)
21. Özdemir, O., Kerzel, M., Weber, C., Lee, J.H., Wermter, S.: Language model-based paired variational autoencoders for robotic language learning. IEEE Transactions on Cognitive and Developmental Systems **15**(4), 1812–1824 (2023)
22. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763. PMLR (2021)
23. Rana, K., Haviland, J., Garg, S., Abou-Chakra, J., Reid, I., Suenderhauf, N.: SayPlan: Grounding large language models using 3d scene graphs for scalable task planning. In: Conference on Robot Learning (CoRL) (2023)
24. Ren, A.Z., Dixit, A., Bodrova, A., Singh, S., Tu, S., Brown, N., et al.: Robots that ask for help: Uncertainty alignment for large language model planners. Conference on Robot Learning (CoRL) (2023)
25. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 10684–10695 (2022)
26. Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., et al.: PROGPROMPT: Generating situated robot task plans using large language models. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 11523–11530. IEEE (2023)
27. Song, C.H., Wu, J., Washington, C., Sadler, B.M., Chao, W.L., Su, Y.: LLM-Planner: Few-shot grounded planning for embodied agents with large language models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2998–3009 (2023)

28. Sun, X., Weber, C., Kerzel, M., Weber, T., Li, M., Wermter, S.: Learning visually grounded human-robot dialog in a hybrid neural architecture. In: International Conference on Artificial Neural Networks. pp. 258–269. Springer (2022)
29. Valmeekam, K., Marquez, M., Olmo, A., Sreedharan, S., Kambhampati, S.: PlanBench: an extensible benchmark for evaluating large language models on planning and reasoning about change. In: Advances Neural Information Processing Systems. vol. 36 (2024)
30. Wake, N., Kanehira, A., Sasabuchi, K., Takamatsu, J., Ikeuchi, K.: GPT-4V (ision) for robotics: Multimodal task planning from human demonstration. arXiv preprint arXiv:2311.12015 (2023)
31. Wang, J., Wu, Z., Li, Y., Jiang, H., Shu, P., Shi, E., et al.: Large language models for robotics: Opportunities, challenges, and perspectives. arXiv preprint arXiv:2401.04334 (2024)
32. Wang, X., Chen, G., Qian, G., Gao, P., Wei, X.Y., Wang, Y., Tian, Y., Gao, W.: Large-scale multi-modal pre-trained models: A comprehensive survey. Machine Intelligence Research pp. 1–36 (2023)
33. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-Thought prompting elicits reasoning in large language models. In: Advances in Neural Information Processing Systems. vol. 35, pp. 24824–24837 (2022)
34. Wermter, S., Palm, G., Elshaw, M.I.: Biomimetic Neural Learning for Intelligent Robots. Intelligent Systems, Cognitive Robotics and Neuroscience. Springer (Jul 2005)
35. Wu, J., Wang, J., Yang, Z., Gan, Z., Liu, Z., Yuan, J., Wang, L.: GRiT: A generative region-to-text transformer for object understanding. arXiv preprint arXiv:2212.00280 (2022)
36. Wu, J., Antonova, R., Kan, A., Lepert, M., Zeng, A., Song, S., Bohg, J., Rusinkiewicz, S., Funkhouser, T.: Tidybot: Personalized robot assistance with large language models. In: Autonomous Robots. vol. 47, pp. 1087–1102. Springer (2023)
37. Yang, Z., Li, L., Lin, K., Wang, J., Lin, C.C., Liu, Z., Wang, L.: The Dawn of LMMs: Preliminary explorations with GPT-4V (ision). In: arXiv preprint arXiv:2309.17421. vol. 9, p. 1 (2023)
38. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., Narasimhan, K.: Tree of thoughts: Deliberate problem solving with large language models. In: Advances in Neural Information Processing Systems. vol. 36 (2024)
39. Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)
40. Zhao, X., Li, M., Weber, C., Hafez, M.B., Wermter, S.: Chat with the Environment: Interactive multimodal perception using large language models. In: Conference on Intelligent Robotsand Systems (IROS). pp. 3590–3596 (2023)
41. Zhao, Z., Lee, W.S., Hsu, D.: Large language models as commonsense knowledge for large-scale task planning. Advances in Neural Information Processing Systems **36** (2024)
42. Zhou, H., Yao, X., Meng, Y., Sun, S., BIng, Z., Huang, K., Knoll, A.: Language-conditioned learning for robotic manipulation: A survey. arXiv preprint arXiv:2312.10807 (2023)
43. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., et al.: RT-2: Vision-language-action models transfer web knowledge to robotic control. In: Conference on Robot Learning (CoRL). pp. 2165–2183. PMLR (2023)