

Supplementary Material of “P-RAG: Progressive Retrieval Augmented Generation for Planning on Embodied Everyday Tasks”

Anonymous Authors

In this supplementary file, we introduce the detail of the prompt setting in our framework, and provide more experiments on the proposed P-RAG, including more iteration correlation analysis, more visual results, and extra retrieval experiments. Besides, we also provide 46 planning videos in the ALFRED and MINI-BEHAVIOR datasets with details introduced in the last section of this file.

1 PROMPT SETTING

The prompt design in P-RAG is structured including the following four parts:

- **Prior Information Prompt.** Prior Information Prompt is the prior information provided to P-RAG about the environment, such as the specific task requiring the LLM to play the role of a smart home assistant robot.
- **Action Prompt.** Action Prompt is used to provide explanations to P-RAG about the function of each action name. We use the PDDL (Planning Domain Definition Language) format to organize actions, enabling a more precise description of each action.
- **Reference Information Prompt.** Reference Information Prompt is the integration of retrieved historical trajectories provided to P-RAG as reference for decision-making.
- **Action Formatting Prompt.** Action Formatting Prompt guides the LLM to generate not only plausible actions but also to adhere to specific formatting requirements. In our experiment, actions are generated in the form of a list, such as “[open the microwave]”.

(# Prior Information Prompt for MINI-BEHAVIOR)

Imagine you are an agent in a 2D grid game and need to finish a task mission. There are objects and furniture on the grid. The grid have *three* dimensions. Furniture like {furniture} cannot be pick_up. The objects (include furniture) with their states are {objects_state}

(# Prior Information Prompt for ALFRED)

You are a helpful assistant that can plan household tasks.

(# Action Prompt for MINI-BEHAVIOR)

```
1.(:action pick_up
:parameters (?target_obj)
:precondition ((target_obj is not furniture) and (not carrying
anything))
:effect (carrying target_obj)
You CAN ONLY pick up one thing meanwhile
so ['pick_up target_0' , 'pick_up target_1'] is not allowed
—
2.(:action drop
:parameters (?relationship ?target_furniture)
:precondition (and (carrying something) and (cell behind
agent is empty in the dimension) and (relationship is one of
['on','below']))
:effect ((not carrying anything) and(object carrying ?relation-
ship target_furniture))
e.g.
drop(on target_furniture) means you are picking up ob-
ject,and want to put it on top of the target_furniture,
drop(below target_furniture) means you are picking up ob-
ject,and want to put it below the target_furniture
...
```

(# Action Prompt for ALFRED)

“Base on the history” you have done for this task, you need and only need to choose which action to do for next one step. “Nothing happens” means you do the same action as the last step or the action is invalid. You can only choose from the following admissible commands (any other commands will be invalid): {admissible_commands}

(# Reference Information Prompt for MINI-BEHAVIOR)

Looks like the task didn’t finish as expected. Let’s figure out what to do next!

We have a Python dictionary that represents the current state of our environment. It’s organized in two ways:

1. object_1/object_2/relationship: Value - This tells us how ‘object_1’ is related to ‘object_2’. For instance, ‘rag_0/shelf_0/onTop: True’ means that ‘rag_0’ is on top of ‘shelf_0’.

2. object_1/attribute: Value - This tells us about a certain attribute of ‘object_1’. For instance, ‘bucket_0/onfloor: True’ means that ‘bucket_0’ is on the floor.

Here’s what our environment looks like right now:

{state}

I’ve found a similar situation and the action taken in that case(maybe empty). You can use this as a reference or come up with your own solution:

{example}

Unpublished working draft. Not for distribution.

Permission to make digital or hard copies of all or part of this work for personal or internal use, or for the internal or personal use of specific clients, is granted by ACM for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM MM, 2024, Melbourne, Australia

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

(# Reference Information Prompt for ALFRED)

There are some tasks that are similar to the current task other agent has done before. Note the object in the same square brackets may occur together. You cannot directly use the action in the similar task, but you can use the history of the similar task to help you make next step decision. Here are the top {k} similar tasks, histories, scene graphs and finished status: {top_k_results}

(# Action Formatting Prompt for MINI-BEHAVIOR)

Please format the actions you give into a Python list of strings. Each action should be described in a string, following the examples provided. Ensure that your output does not include any extraneous text or formatting. 'target_obj' should be replaced with the actual object name. Example: ['pick_up the target_obj', 'drop the target_obj'] DO NOT output redundant text

(# Action Formatting Prompt for ALFRED)

If you do not see the target you require, feel free to choose above action to explore the scene. You can only see part of the scene at a place. Please give me a plan with action in [] like ### ['go to target_n'] OR ['take target_n from source_n'] ### which I can execute with python function "eval()" to be a list of strings.

2 CORRELATION BETWEEN P-RAG AND OTHER SETTINGS

		P-RAG		GPT-4		P-RAG (Iter.)	
		True	False	True	False	True	False
P-RAG	True	-	-	70.45%	29.5%	73.3%	26.7%
	False	-	-	5.9%	94.1%	6.8%	93.2%
GPT-4	True	72.1%	27.9%	-	-	74.4%	25.6%
	False	6.3%	93.7%	-	-	6.8%	93.2%
P-RAG (Iter.)	True	70.2%	29.8%	69.6%	30.1%	-	-
	False	5.9%	94.1%	5.4%	94.6%	-	-

Table 1: Relation Between P-RAG, GPT-4 Baseline and P-RAG (Iter.) evaluation on Valid Seen of ALFRED. Each group of data consists of four cells, with each cell representing a proportion. The proportions indicate the ratio of True/False evaluations for the method assessment corresponding to the completion status (True/False) of the task in the row to the method assessment (True/False) in the column, as shown in Formula 1.

Table 1, 2, and 3 display the correlation of success rates among different P-RAG and other settings within the same task dataset. Firstly, we present the correlation between successful tasks among P-RAG, GPT-4 baseline, and P-RAG (iter.) on the ALFRED Valid Seen dataset in Table 1. Each cell in the table can be calculated according to the following formula:

$$R = \frac{\sum_{i \in [1, N]} I(D_{column, i} = S_{column}) I(D_{row, i} = S_{row})}{\sum_{i \in [1, N]} I(D_{row, i} = S_{row})}, \quad (1)$$

where $I(\cdot)$ is indicator function, N is the total number of the tasks, $D_{column, i}$, $D_{row, i}$ stand for the done state of the i -th task with column and row method, S_{column} and S_{row} denote True or False state in the corresponding row and column, respectively. The P-RAG method refers to evaluation on the ALFRED Valid Seen dataset after three iterations on Train100. P-RAG (Iter.) stands for the method evaluated on the ALFRED Valid Seen dataset after one iteration on Valid Seen dataset. The experimental results indicate that compared to P-RAG, P-RAG (Iter.) can successfully complete more tasks in both successful and failed tasks relative to the GPT-4 baseline. Secondly, the data presented in Table 2 suggests that the increase in success rates with different iterations of P-RAG (Iter.) primarily stems from tasks that P-RAG failed to complete, rather than tasks that P-RAG itself successfully completed. Thirdly, the format of Table 3 is a transposed version of Table 2. The results from the table demonstrate that P-RAG can fully cover the tasks successfully completed by the GPT-4 baseline and also complete some of the tasks failed by the GPT-4 baseline. Subsequent data illustrates that the difference in successful tasks between P-RAG and P-RAG (Iter.) increases as the number of iterations increases.

3 ANALYSIS OF VISUAL RESULTS

We provide three visual cases to demonstrate the decision-making ability of P-RAG in Fig. 1 Fig. 2 and Fig. 3. As shown in Fig1, P-RAG needs to complete "Cool a mug in fridge". The agent successfully understands the goal instruction and decomposes the instructions into different stages with steps as followings:

- (1) go to cabinet 2,
- (2) open cabinet 2,
- (3) take mug 1 from cabinet 2,
- (4) go to fridge 1,
- (5) open fridge 1,
- (6) put mug 1 in/on fridge 1,
- (7) close fridge 1,
- (8) open fridge 1,
- (9) take mug 1 from fridge 1,
- (10) go to coffeemachine 1,
- (11) put the mug 1 in/on the coffeemachine 1.

Fig. 2 shows another case of trajectory provided by P-RAG. Given the goal instruction of "Place spray bottles on the back of a toilet", agent successfully grabs a spray bottle from the shelf and places it at the back of a toilet. Impressively, it understands the plural property of the word "bottles" in the instruction, and then proceeds to grab another spray bottle and place it at the back of the toilet as before with steps as followings:

- (1) go to shelf 1,
- (2) take spraybottle 1 from shelf 1,
- (3) go to toilet 1,
- (4) put spraybottle 1 in/on toilet 1,
- (5) go to shelf 2,
- (6) go to shelf 3,
- (7) take spraybottle 2 from shelf 3,
- (8) go to toilet 1,
- (9) put spraybottle 2 in/on toilet 1.

We also provide an example of P-RAG's failure to accomplish, shown in Fig. 3. The agent needs to complete the task "Put two

		GPT-4		P-RAG (1 Iter.)		P-RAG (2 Iter.)		P-RAG (3 Iter.)	
		True	False	True	False	True	False	True	False
P-RAG	True	50.0%	50.0%	50.0%	50.0%	58.3%	41.7%	50.0%	50.0 %
	False	0.0%	100%	6.9%	93.1%	13.9%	86.1%	16.7%	83.3%

Table 2: Relation Between P-RAG, GPT-4 Baseline and P-RAG (Iter.) evaluation on Valid Unseen of ALFRED. Each group of data consists of four cells, with each cell representing a proportion. The proportions indicate the ratio of True/False evaluations for the method assessment corresponding to the completion status (True/False) of the task in the row to the method assessment (True/False) in the column.

		GPT-4		P-RAG (1 Iter.)		P-RAG (2 Iter.)		P-RAG (3 Iter.)	
		True	False	True	False	True	False	True	False
P-RAG	True	100%	8.1%	54.5%	8.2%	41.2%	7.5%	33.3%	9.1 %
	False	0.0%	91.9%	45.5%	91.8%	58.8%	92.5%	66.7%	90.9%

Table 3: Relation Between P-RAG, GPT-4 Baseline and P-RAG (Iter.) evaluation on Valid Unseen of ALFRED. Each group of data consists of four cells, with each cell representing a proportion. The proportions indicate the ratio of True/False evaluations for the method assessment corresponding to the completion status (True/False) of the task in the column to the method assessment (True/False) in the row.

rolls of toilet paper on the back of a toilet”. Initially, it correctly grabs a roll of toilet paper and places it on the back of a toilet. However, when attempting to handle the next roll of toilet paper, it mistakenly grabs the one already placed on the back of the toilet, leading to subsequent invalid actions with all steps :

- (1) go to cabinet 1,
- (2) open cabinet 1,
- (3) take toiletpaper 1 from cabinet 1,
- (4) go to cabinet 2,
- (5) open cabinet 2,
- (6) go to cabinet 3,
- (7) open cabinet 3,
- (8) go to toilet 1,
- (9) put toiletpaper 1 in/on toilet 1,
- (10) go to cabinet 3,
- (11) take toiletpaper 1 from cabinet 3,
- (12) go to toilet 1,
- (13) put toiletpaper 1 in/on toilet 1,
- (14) go to cabinet 1,
- (15) close cabinet 1,
- (16) close cabinet 1,
- (17) close cabinet 3,
- (18) go to cabinet 2,
- (19) close cabinet 2
- (20) go to toilet 1.

4 RETRIEVAL RESULT

In our retrieval process, we return top K relevant historical trajectories to LLM, and set $K = 3$. We display part of the retrieved results as followings.

Query task name: Put in the heated apple in the trash bin
Query observation: You arrive at loc 11. The microwave 1 is closed
The retrieved top 3 tasks:
 1) Put in the heated apple in the trash bin.
 2) Put the heated slice apple in the trash bin.
 3) Put a warmed up apple in the trash can.

Query task name: to cook a sliced tomato to throw it in the trash.
Query task observation: You arrive at loc 40. On the garbage-can 1, you see a plate 1, and a ladle 1.
The retrieved top 3 tasks are:
 1) Slice tomato to be placed in a pan with a knife to be moved to the counter by the stove.
 2) Heat up a slice of tomato in the microwave to put in the fridge.
 3) Cut a tomato on the counter.

Query task name: put a cup and knife on a counter top”
Query observation: You open the cabinet 1. The cabinet 1 is open. In it, you see a soapbottle 1, a plate 1, a glassbottle 1, a peppershaker 1, and a spoon 1.”
The retrieved top 3 tasks are:
 1) put a cup and knife on a counter top.
 2) place a cup with a knife in it on the kitchen counter space.
 3) Move a knife and a green cup to the counter.

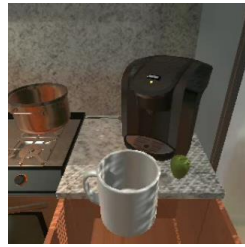
5 VIDEO EXPLANATION

We provide a total of 46 videos across two datasets, with 41 videos in ALFRED and 5 videos in MINI-BEHAVIOR. Each frame in the ALFRED videos represents the state after executing an action displayed in text. We collect 19 trajectories for Valid Seen and 22 trajectories for Valid Unseen. The videos in MINI-BEHAVIOR depict textual descriptions of actions, which will actually be decomposed into smaller sub-actions such as “turn left”, “turn right” and “forward”. All the videos conclude with the text “DONE” to signify the successful completion of the task.

Cool a mug in fridge



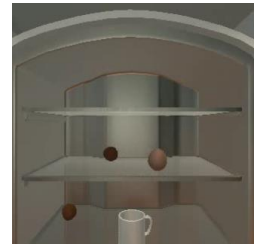
1. Go to cabinet 2



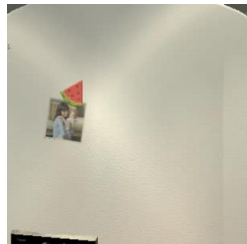
2. Take mug 1



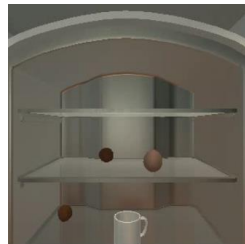
3. Go to fridge 1



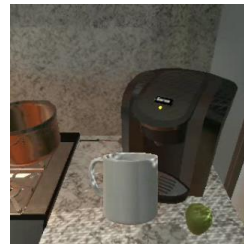
4. Put mug1 in fridge 1



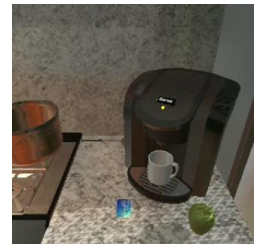
5. Close fridge 1



6. Open fridge 1



7. Take mug 1



8. Put the mug 1

Figure 1: Visualization of trajectory in the ALFRED Valid Unseen dataset with P-RAG. We visualize the trajectory sampling of the task “Cool a mug in fridge”. Note that this trajectory consists of a total of 11 steps, and we display the key 8 steps.

Place spray bottles on the back of a toilet



1. Go to shelf 1



2. Take spraybottle 1



3. Go to toilet 1



4. Put spraybottle 1



5. Go to shelf 3



6. Take spraybottle 2



7. Go to toilet 1



8. Put the spraybottle 2

Figure 2: Visualization of trajectory in the ALFRED Valid Seen dataset with P-RAG. We visualize the trajectory sampling of the task “Place spray bottles on the back of a toilet”. Note that this trajectory consists of a total of 9 steps, and we display the key 8 steps.

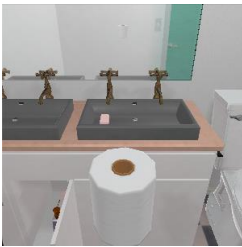
Put two rolls of toilet paper on the back of a toilet



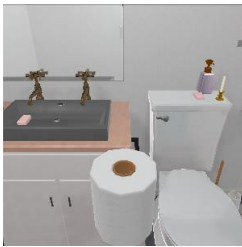
1. Go to cabinet 1



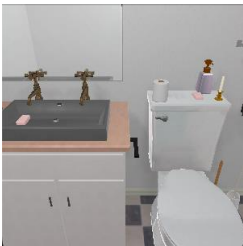
2. Open cabinet 1



3. Take toiletpaper 1



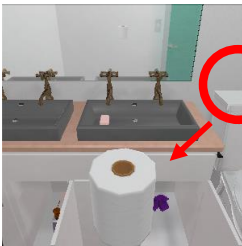
4. Go to toilet 1



5. Put toiletpaper 1



6. Go to cabinet 1



7. Take toiletpaper 1

Failed

Figure 3: Visualization of failed trajectory in the ALFRED Valid Seen dataset with P-RAG. We visualize the trajectory sampling of the task “Put two rolls of toilet paper on the back of a toilet”. Note that this trajectory consists of a total of 20 steps, and we display the key 7 steps. As indicated by the red circle, the agent mistakenly identifies the already placed toilet paper as the second one to be placed, picking it up again and putting it at the back of the toilet.