Appendix

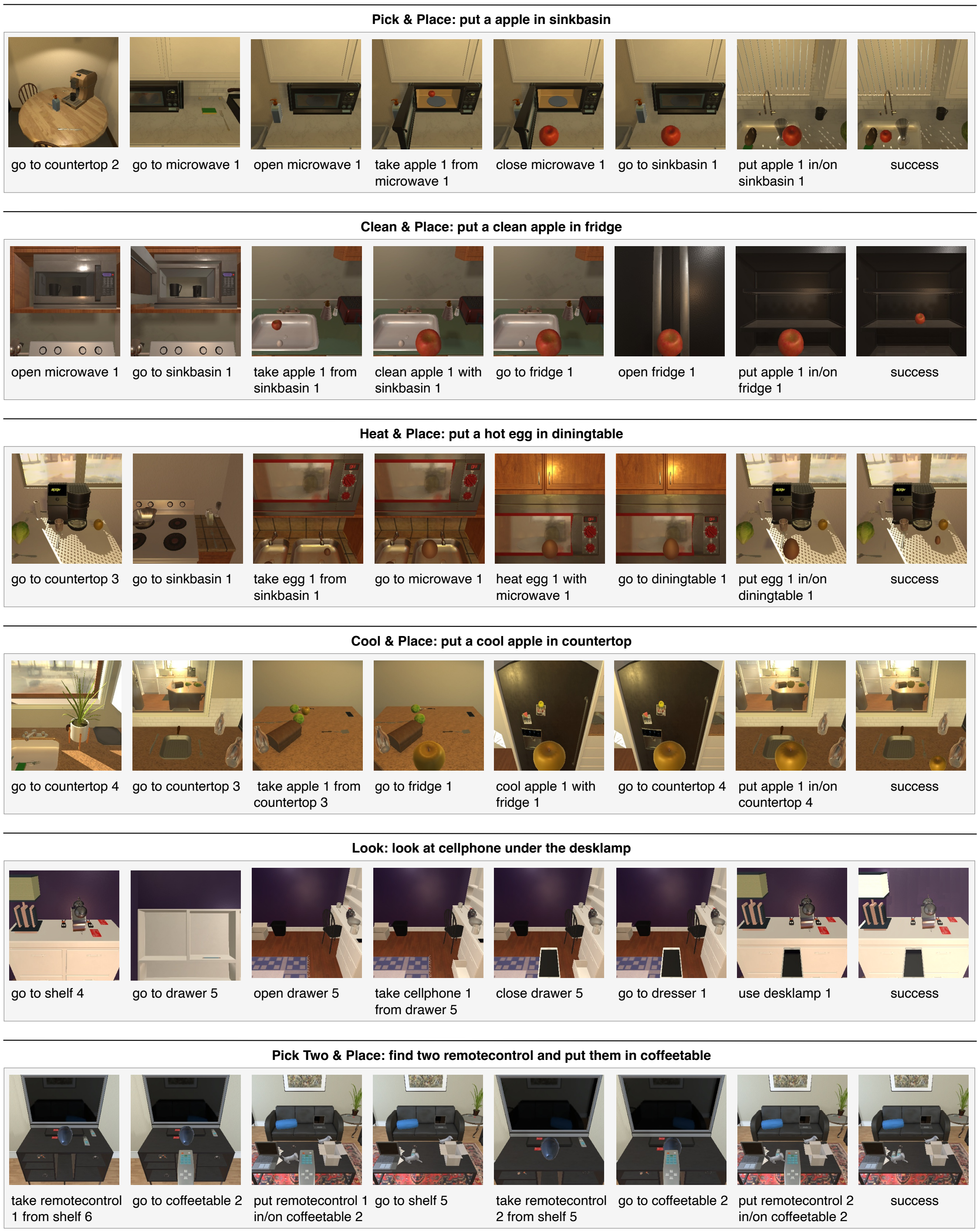

Figure 8: **Visualized task examples of ALFWorld [56].** This benchmark adopts diverse household scenes developed by the Ai2Thor [30] environment, in which all objects can be relocated to different positions based on placeable surface areas and class constraints, enabling the procedural generation of a large pool of new tasks via recombining different objects and goal positions.

6. Full Prompts for LLM Expert

In this section, we provide all LLM prompts for the training procedure (Alg. 1) of EMMA. We adopt the prompting technique developed by ReAct [73] but ignore the reasoning traces, i.e., "think" steps, when executing imitation learning between EMMA and the LLM actor. After each trial i, the retrospective feedback $\mathcal{P}_i$ generated by the LLM critic will be appended to long-term memory $\mathcal{P}$. In practice, we bound $\mathcal{P}$ by a maximum number of stored feedback (usually set to 1-3) to adhere to the max context length of the LLM.

An example of the prompt adopted by LLM actor for generating actions

```
{environment: You are in the middle of a room. Looking quickly around you, you see a cabinet
4, a cabinet 3, a cabinet 2, a cabinet 1, a countertop 1, a garbagecan 1, a handtowelholder 2, a
handtowelholder 1, a sinkbasin 2, a sinkbasin 1, a toilet 1, a toiletpaperhanger 1, and a towelholder 1.
task instruction: Your task is to: put some spraybottle on toilet.
> think: To solve the task, I need to find and take a sparybottle, then put it on toilet.
> think: First I need to find a spraybottle. A spraybottle is more likely to appear in cabinet (1-4),
countertop (1), toilet (1), sinkbasin (1-2), garbagecan (1). I can check one by one, starting with
cabinet 1.
> go to cabinet 1
On the cabinet 1, you see a cloth 1, a soapbar 1, a soapbottle 1.
> go to cabinet 2
The cabinet 2 is closed.
> open cabinet 2
You open the cabinet 2. The cabinet 2 is open. In it, you see a candle 1, and a spraybottle 2.
> think: Now I find a spraybottle (2). Next, I need to take it.
> take spraybottle 2 from cabinet 2
You pick up the spraybottle 2 from the cabinet 2.
> think: Now I take a spraybottle (2). Next, I need to put it in/on toilet 1.
> go to toilet 1
On the toilet 1, you see a soapbottle 2.
> put spraybottle 2 in/on toilet 1
You put the spraybottle 2 in/on the toilet 1.} few-shot examples × n

long-term memory P: [...]
target environment: ...
task instruction: ...
>
```

An example of the prompt adopted by LLM critic for generating retrospection

```
{previous trial: You are in the middle of a room. Looking quickly around you, you see a cabinet 10, a
cabinet 9, a cabinet 8, a cabinet 7, a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a
cabinet 1, a coffeemachine 1, a countertop 2, a countertop 1, a diningtable 2, a diningtable 1, a drawer
3, a drawer 2, a drawer 1, a fridge 1, a garbagecan 1, a microwave 1, a sinkbasin 1, a stoveburner 4, a
stoveburner 3, a stoveburner 2, a stoveburner 1, and a toaster 1.
Your task is to: heat some mug and put it in coffeemachine.
> go to countertop 1
On the countertop 1, you see a butterknife 1, a cup 1, a fork 2, and a mug 1.
> take mug 1 from countertop 1
You pick up the mug 1 from the countertop 1.
> go to stoveburner 1
On the stoveburner 1, you see a pan 2.
> examine stoveburner 1
On the stoveburner 1, you see a pan 2.
> examine stoveburner 1
On the stoveburner 1, you see a pan 2.
> examine stoveburner 1
On the stoveburner 1, you see a pan 2.
> examine stoveburner 1
On the stoveburner 1, you see a pan 2.
STATUS: FAIL
retrospection: You were stuck in a loop in which you continually examined stoveburner 1 instead of
heating mug 1 with stoveburner 1. You should have taken mug 1 from countertop 1, then heated it with
stoveburner 1, then put it in coffeemachine 1. It did not help to execute two identical actions in a row.
You will try to execute a different action if You am stuck in a loop again.} few-shot examples × n

current trial: ...
retrospection:
```

7. Parallel TextWorld

While the idea of parallel TextWorld is heavily inspired by previous work [55, 56], we have enhanced the TextWorld engine to create text-based equivalents of each visual environment for training and evaluating language-based agents. This enhancement involves utilizing a combination of the PDDL [1] and Fast Downward [22] to maintain and update the textual state of the simulated environments. Based on the metadata provided by the simulator, we represent a visual state as a list of attributes. These attributes detail the relationships among various entities in the environment, such as positions, goals, and objects. Note that all these attributes, variables, and rules are defined within the framework of PDDL.

8. Training Details

We provide hyperparameters used for training EMMA in Table 2. These hyperparameters are largely derived from those proposed for finetuning InstructBLIP model [11]. When training, we only update the parameters of linear projection layer while keeping other components frozen, as done during instruction tuning for many existing work [17, 76]. We use the AdamW optimizer [39] with a linear warmup of the learning rate, followed by a linear decay with a minimum learning rate of 0. Moreover, we remove the instruction input of Q-Former, which is used in InstructBLIP, and find this improves performance cross all experiments. Our implementation is heavily inspired by the LAVIS library [33] so the training and evaluation processes use the standard procedure provided by LAVIS.

Hyperparameter	**Value**
EMMA's Architecture	
LLM decoder	Vicuna-7b-v1.1 [74]
Image encoder	ViT-L [46]
Q-Former	$\text{BERT}_{\text{base}}$ [12]
Pretrained weights	InstructBLIP [11]
Number of query tokens	32
Q-Former text input	False
Max text length	1024
Image resolution	224
Behavior Cloning	
Finetuning epochs	6
Warmup steps	1000
Learning rate	10^{-5}
Batch size	128
AdamW β	$(0.9, 0.999)$
Weight decay	0.05
Drop path	0
Inference beam size	5
Imitation Learning	
Base model for LLM expert	text-davinci-003
Prompts for LLM expert	refer to Sec. 6
Number of trials	12
Episode length	30
Size of long-term memory	3
Learning rate	5×10^{-6}
Warmup steps	300
Batch size	16
Training epochs per trial	5
DPO β	0.1

Table 2: Hyperparameters of EMMA for ALFWorld experiments

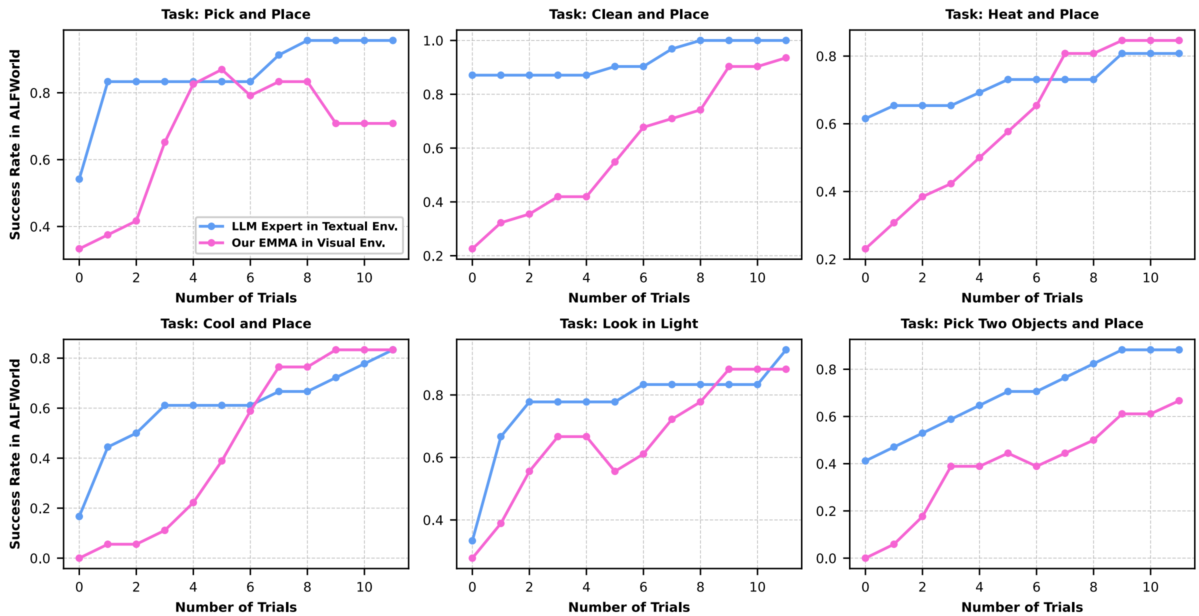

Figure 9: **Comparison of success rate between EMMA and the LLM expert.** As the number of trials increases, the gap between the two agents decreases, and EMMA even outperforms or matches the expert in some tasks (e.g., "Heat and Place" and "Cool and Place"), indicating the effectiveness of cross-modality imitation learning.

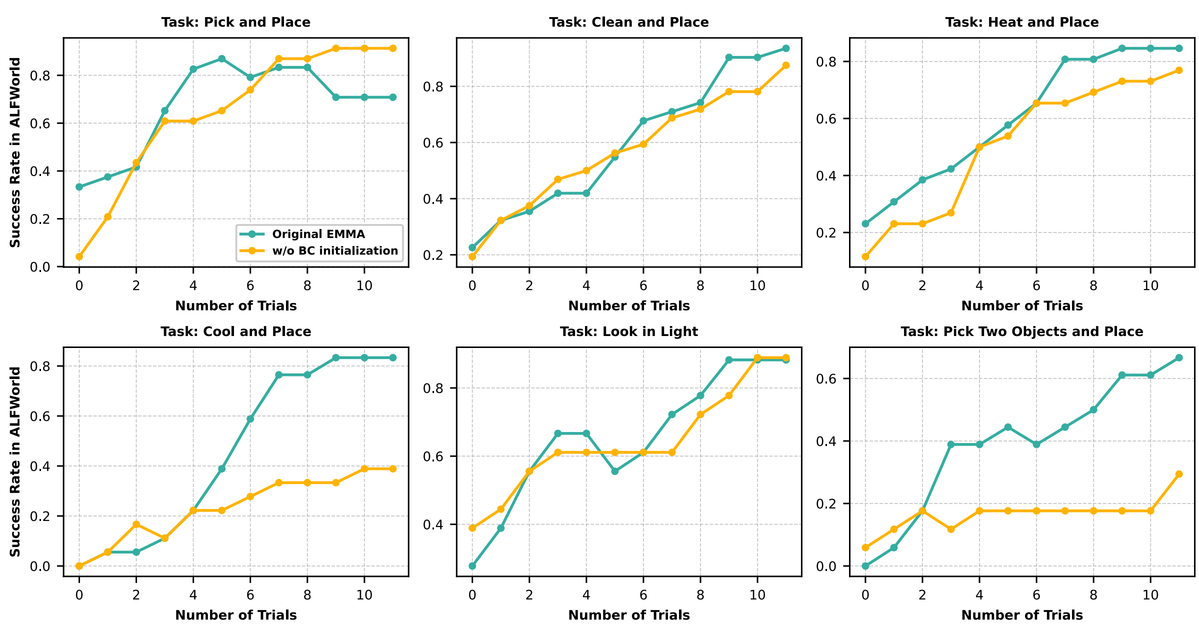

Figure 10: **Ablation study.** The performance of "EMMA w/o BC initialization" is consistently worse than the original EMMA.

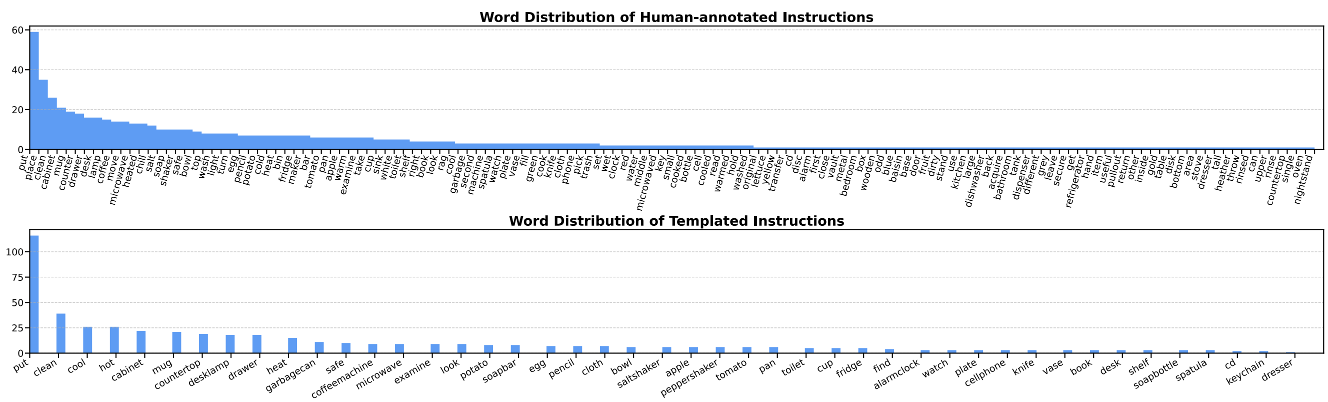

Figure 11: **Vocabulary Distributions.** Frequency distribution of all words for human-annotated and templated task instructions. The diversity of human-annotated instructions presents a significant challenge for the generalization abilities of agents.

9. Collection of Demonstration Dataset

Fine-tuning pretrained VLMs on a pre-collected demonstration dataset via behavior cloning is a critical step, enabling these models to comprehend and follow the unique grammar of ALFWorld as well as to develop a basic "gamesense". However, the number of task instructions in the original ALFWorld [55] is too limited to yield sufficient data for fine-tuning these large pretrained VLMs effectively. Hence, we propose an automated pipeline, which leverages *text-davinci-003* and a rule-based planner to generate a large amount of new instructions and their resulting expert demonstrations, respectively.

To generate a diverse set of new task instructions, we harness the in-context learning capabilities of LLM. Our procedure begins with extracting detailed descriptions from the ALFWorld benchmark for each environment, providing comprehensive information on the number and functional attributes of all items. Then, based on the types of room in these environments, we design different prompts that aim at inducing the LLM to generate task instructions aligned with the features of the target environment. An example of these prompts is detailed in Table 3. For each generated task instruction, we gather demonstrations $\{x_{\text{task}}, s_v^t, x_a^t\}_{t=0}^{T}$ using a rule-based planner devised by ALFWorld. It's important to note that this planner operates with an unfair advantage: it considers the environment as fully observable and has complete information of world dynamics, relying on metadata that is not accessible to the agent during training. In summary, our dataset comprises 15247 expert demonstration episodes, amounting to 178585 image-text pairs.

```
Q:
environment: You are in the middle of a room. Looking quickly around you, you see a cabinet, a countertop, a
cabinet, a countertop, a drawer, a drawer, a drawer, a stoveburner, a stoveburner, a drawer, a stoveburner, a
stoveburner, a cabinet, a cabinet, a microwave, a cabinet, a cabinet, a cabinet, a sink, a sinkbasin, a fridge, a
toaster, a coffeemachine, a cabinet, a drawer, a drawer, a drawer, a drawer, a shelf, a shelf, a countertop, a shelf,
a drawer, and a garbagecan.
object dictionary: all of operable objects are listed in the following dictionary with a consistent format {type
of operation: {object name: number of objects}}: {'pickupable': {'dishsponge': 3, 'apple': 2, 'butterknife':
3, 'peppershaker': 2, 'saltshaker': 3, 'bowl': 2, 'spatula': 2, 'pot': 3, 'winebottle': 3, 'statue': 2,
'creditcard': 3, 'plate': 2, 'pan': 2, 'kettle': 3, 'soapbottle': 3, 'potato': 3, 'fork': 2, 'bread': 2,
'knife': 3, 'glassbottle': 3, 'book': 1, 'tomato': 1, 'vase': 2, 'egg': 1, 'papertowelroll': 1, 'cup': 1,
'lettuce': 1, 'spoon': 1, 'mug': 1}, 'sliceable': {'apple': 2, 'potato': 3, 'bread': 2, 'tomato': 1, 'egg':
1, 'lettuce': 1}, 'receptacle': {'bowl': 2, 'pot': 3, 'plate': 2, 'pan': 2, 'stoveburner': 4, 'drawer': 9,
'countertop': 3, 'cabinet': 9, 'microwave': 1, 'shelf': 3, 'toaster': 1, 'garbagecan': 1, 'cup': 1, 'fridge':
1, 'coffeemachine': 1, 'sinkbasin': 1, 'mug': 1}, 'canFillWithLiquid': {'bowl': 2, 'pot': 3, 'winebottle': 3,
'kettle': 3, 'glassbottle': 3, 'houseplant': 1, 'cup': 1, 'mug': 1}, 'breakable': {'bowl': 2, 'winebottle': 3,
'statue': 2, 'plate': 2, 'glassbottle': 3, 'window': 1, 'vase': 2, 'egg': 1, 'cup': 1, 'mug': 1}, 'dirtyable':
{'bowl': 2, 'pot': 3, 'plate': 2, 'pan': 2, 'cup': 1, 'mug': 1}, 'openable': {'kettle': 3, 'drawer': 9,
'cabinet': 9, 'book': 1, 'microwave': 1, 'fridge': 1}, 'cookable': {'potato': 3}, 'toggleable': {'stoveknob':
4, 'microwave': 1, 'toaster': 1, 'coffeemachine': 1, 'lightswitch': 1, 'faucet': 1}}
generate 30 diverse tasks based on the environment description and object dictionary.
A:
Task 1: pick_clean_then_place_in_recep, put a clean pan in fridge
Task 2: pick_and_place_simple, put a apple in countertop
Task 3: pick_two_obj_and_place, put two apple in garbagecan
Task 4: pick_heat_then_place_in_recep, put a hot apple on plate
Task 5: pick_cool_then_place_in_recep, put a cool apple in countertop
Task 6: pick_and_place_simple, put a mug in coffeemachine
Task 7: pick_two_obj_and_place, put two creditcard in plate
Task 8: pick_clean_then_place_in_recep, put a clean mug in coffeemachine
Task 9: pick_heat_then_place_in_recep, put a hot mug on plate
Task 10: pick_cool_then_place_in_recep, put a cool mug in plate
Task 11: pick_and_place_simple, put a peppershaker in cabinet
Task 12: pick_two_obj_and_place, put two peppershaker in shelf
Task 13: pick_clean_then_place_in_recep, put a clean fork in pot
Task 14: pick_heat_then_place_in_recep, put a hot fork on bowl
Task 15: pick_cool_then_place_in_recep, put a cool fork in plate
Task 16: pick_and_place_simple, put a statue in countertop
Task 17: pick_two_obj_and_place, put two bowl in microwave
Task 18: pick_clean_then_place_in_recep, put a clean potato in fridge
Task 19: pick_heat_then_place_in_recep, put a hot potato on plate
Task 20: pick_cool_then_place_in_recep, put a cool potato in pot
Task 21: pick_and_place_simple, put a egg in countertop
Task 22: pick_two_obj_and_place, put two bread in microwave
Task 23: pick_clean_then_place_in_recep, put a clean egg in garbagecan
Task 24: pick_heat_then_place_in_recep, put a hot egg on bowl
Task 25: pick_cool_then_place_in_recep, put a cool egg in pan
Task 26: pick_and_place_simple, put a pan in stoveburner
Task 27: pick_two_obj_and_place, put two pot in stoveburner
Task 28: pick_clean_then_place_in_recep, put a clean tomato in coffeemachine
Task 29: pick_heat_then_place_in_recep, put a hot tomato on plate
Task 30: pick_cool_then_place_in_recep, put a cool tomato in plate

Q:
environment: ...
object dictionary: ...
generate 30 diverse tasks based on the environment description and object dictionary.
A:
...LLM-generated task instructions...
```

Table 3: An example of the prompt for generating new task instructions in the kitchen