Execution Traces

Execution Example: 01

Number of agents: 2 (1 human, 1 ad hoc agent using the proposed architecture)

Task routine: *Prepare Breakfast for a Weekday, Prepare Workstation, Make Coffee, Serve Snacks*

In this scenario the ad hoc agent used the LLM to anticipate the high-level future tasks in the environment, which were then validated and adapted based on prior domain-specific knowledge and experience. The prompt to the LLM was automatically generated using a pre-defined system message, two examples for few-shot prompting which are randomly selected from agent's prior experience and accompanied with Chain-of-Thoughts(CoT) explanations, and any information on current and completed tasks (as described in Section 3.3 of the paper). Below is the prompt created and sent to the LLM by the agent in this example:

> You are an intelligent household assistant. Anticipate future tasks based on previous days' data. You will get a list of tasks that can be executed in the household environment, day of the week, an example of a partially completed routine and the expected output. Complete the rest of the tasks for the partially completed routine only using the tasks from the task list, following the examples given.
> Task List: [Prepare breakfast weekday,Prepare coffee,Prepare home work-station, Prepare lunch, Pack bag, Clean kitchen, Prepare breakfast weekend, Prepare table for guests, Serve snacks, Prepare activities, Clean dishes]
>
> Example1: Complete the following routine on a weekend the human is having guests for lunch : []
> [Prepare breakfast weekend, Prepare table for guests, Prepare lunch, Clean dishes] As everyday on weekends it is important to first prepare breakfast. Since the guests are arriving then the agent must make table to serve lunch to the guests. It is important to start on lunch as soon as the table is ready. Finally the agent may not have time to clean the whole kitchen while having guests but it can clean atleast the dirty dishes from lunch to maintain the hygiene.
>
> Example2: Complete the following routine on a weekday the human is leaving for office: []
> [Prepare breakfast weekday, Pack bag, Prepare lunch, Clean kitchen] Since the human is leaving for the office, it's a busy morning focused on essentials. Breakfast is prepared first to start the day with energy. Next, packing the bag ensures they have everything needed for work. Lunch is then prepared in case they need it later. Finally, after they've left, the kitchen is cleaned to maintain a tidy space.
>
> Complete the following routine on a weekday the human is working from home and need a snack: []

The output from the LLM for the above query consisted of the list of anticipated future tasks as below.

> Prepare breakfast weekday, Prepare home work-station, Serve snacks, Clean kitchen
> Since the human is working from home, the first task is to prepare breakfast to fuel their day. Next, setting up the home work-station is essential for a productive work environment.

> As they need a snack, serving snacks follows to keep them energized. Finally, cleaning the kitchen ensures the space remains tidy and conducive to work.

This output was then sent to the validator, where it was verified by comparing with prior experience. The refined task list was,

[Prepare breakfast weekday, Prepare home work-station, Serve snacks]

While the first three tasks were often performed by the human together, based on prior experience, the final anticipated task — Clean kitchen, was rarely selected. Thus the validator removed it from the anticipated list. The selected task list was then combined with any currently assigned tasks, allowing the agent to jointly plan and execute actions within the domain to achieve its current goals.

Execution Example: 02

Number of agents: 3 (1 human, 2 ad hoc agents using the proposed architecture)

Task routine: *Prepare Breakfast for a Weekend, Prepare Table for Guests, Prepare Lunch, Clean Dishes*

The prompt to the LLM by the two ad hoc agents in the above scenario was as follows.

> You are an intelligent household assistant. Anticipate future tasks based on previous days' data. You will get a list of tasks that can be executed in the household environment, day of the week, an example of a partially completed routine and the expected output. Complete the rest of the tasks for the partially completed routine only using the tasks from the task list, following the examples given.
> Task List: [Prepare breakfast weekday,Prepare coffee,Prepare home work-station, Prepare lunch, Pack bag, Clean kitchen, Prepare breakfast weekend, Prepare table for guests, Serve snacks, Prepare activities, Clean dishes]
>
> Example1: Complete the following routine on a weekend the human is not having guests: [] [Prepare breakfast weekend, Clean kitchen, Prepare activities, Prepare lunch] Since it is a weekend, the routine starts with preparing breakfast. After breakfast, the kitchen needs to be cleaned to remove any mess from the meal. The next task involves preparing activities to enjoy the day. Finally, lunch should be prepared to ensure the human is nourished throughout the day.
>
> Example2: Complete the following routine on a weekday the human is leaving for office: [] [Prepare breakfast weekday, Pack bag, Prepare lunch, Clean kitchen] Since the human is leaving for the office, it's a busy morning focused on essentials. Breakfast is prepared first to start the day with energy. Next, packing the bag ensures they have everything needed for work. Lunch is then prepared in case they need it later. Finally, after they've left, the kitchen is cleaned to maintain a tidy space.
>
> Complete the following routine on a weekend the human is having guests for lunch: []

Since each agent prompted the LLM separately the two ad hoc agents received two different outputs.

The output received by the first ad hoc agent was:

> Prepare breakfast weekend, Prepare table for guests, Prepare lunch, Serve snacks

The task list refined by the validator after consulting prior knowledge was,

[Prepare breakfast weekend, Prepare table for guests, Prepare lunch]

Since the human usually did not require snacks after lunch, the validator consulted prior experience and removed the 'Serve snacks' task from the list.

The output received by the second ad hoc agent was:

> Prepare breakfast weekend, Prepare table for guests, Prepare lunch, Serve snacks, Clean dishes

Again, the validator refined the LLM's output by removing the 'Serve snacks' task. Additionally, the LLM correctly anticipates the 'Clean dishes' task for the second agent, thus it was added to the agents task list to be performed.

[Prepare breakfast weekend, Prepare table for guests, Prepare lunch, Clean dishes]

These refined task lists were then used by the two ad hoc agents separately in their planning to select actions and jointly achieve goals.

Execution Example: 03

Number of agents: 2 (1 human, 1 ad hoc agent using Base8 - LLM for low-level action planning)

Task routine: *Prepare Breakfast for a Weekday, Prepare Workstation, Make Coffee, Prepare Lunch*

In this scenario the ad hoc agent used the LLM to directly compute sequences of actions to be executed in the *VirtualHome* domain to achieve specific tasks. The prompt to the LLM consisted of a system message, details of actions available in *VirtualHome* and their intended purpose, some *Action Feasibility Rules*, information about adjacent places in the domain, current world state (location of agents, objects, and appliances, each appliance's state, objects held by the agents), description on task to be performed, immediate previous actions of the human and the ad hoc agent, any specific information to be considered on any given day (human working from home, weekday or the weekend, human is expecting guests or not), detailed example of selecting an action, and a query asking the LLM to generate the next action in the sequence for achieving the assigned goal (as described in Section 4.1 of the paper). Below is a prompt created by the agent:

> You are an intelligent household assistant collaborating with a human to perform daily tasks. Your task is to achieve the goal by selecting the next feasible action in a logical sequence. Briefly explain your reasoning and predict only one action at a time, returning the action as a Python list, e.g., [move(agent,location)]. If the goal is already achieved return 'goal_achieved'. You may choose not to take any action if it is necessary.
>
> Possible actions:
>
> move(agent,location): Move the agent to an adjacent location.
>
> grab(agent,object): Pick up an object.
>
> put(agent,object,surface): Place an object on a surface.

open(agent,appliance): Open an appliance.

close(agent,appliance): Close an appliance.

switchon(agent,appliance): Switch on an appliance.

switchoff(agent,appliance): Switch off an appliance.

Action Feasibility Rules:

Movement Limitation(Critical): You must only move to adjacent locations. Adjacent locations are explicitly defined in the next_to relationships below. If the destination is not adjacent, the move action is prohibited. Always check adjacency before predicting a move.

Object Location: You must be in the same location as an object to perform any action on it (e.g., grab or put).

Carrying Limit: You cannot hold more than two objects at a time. When holding two objects, actions like open, close, switchon, or switchoff require putting down at least one object first.

Appliance Safety: For safety, you should not open appliance doors when they are switched on.

If the human is holding an object, assume they will handle the tasks involving that object. You must not attempt to grab, or interact with the object in the humans possession. Instead, focus on completing other parts of the goal unrelated to that object.

You can only move between the following adjacent places:

next_to(counter_one, kitchentable)

next_to(counter_one, counter_three)

next_to(counter_three, kitchentable)

next_to(kitchentable,kitchen_smalltable)

next_to(kitchen_smalltable, bedroom_desk)

next_to(bedroom_desk, bedroom_coffeetable)

next_to(kitchen_smalltable, livingroom_desk)

next_to(livingroom_desk, livingroom_coffeetable)

Example Sequence for Rational Action:

If a task requires moving an item from one room to another, you should:

1. Move to the item's location.
2. Grab the item.

3. Move to the destination.
4. Place the item.

Current state:

human location: kitchentable, agent location: kitchentable, milk on kitchentable, apple on livingroom_coffeetable, plum on livingroom_coffeetable, mug on livingroom_coffeetable, cupcake on bedroom_coffeetable, candybar on bedroom_coffeetable, wine on bedroom_coffeetable, boardgame on bedroom_coffeetable, waterglass on livingroom_desk, computer on livingroom_desk, juice on bedroom_desk, plate on counter_one, cutlets on counter_one, coffeemaker on counter_three, coffeepot placed inside coffeemaker, book on kitchen_smalltable, chips on kitchen_smalltable, cereal on livingroom_desk, bananas on livingroom_coffeetable, breadslice on counter_three, dishwasher under counter_three, cellphone on livingroom_coffeetable, dishwasher door closed, dishwasher not switchedon, computer not switchedon, coffeemaker not switchedon, coffee not prepared.

The teams current goal is to Prepare breakfast on a weekday:

goal = on(cereal,kitchentable), on(breadslice,kitchentable), on(bananas,kitchentable)

Anticipate an action for you to perform given above state and constraints on a weekday the human is working from home. Ensure the Action Feasibility Rules: Movement Limitation(Critical), Object Location, Carrying Limit and Appliance Safety are strictly respected and predict the next logical action.

The output received by the ad hoc agent from the LLM for above query was:

The current goal is to prepare breakfast by placing cereal, breadslice, and bananas on the kitchentable.

Currently, the human is at the kitchentable, and the cereal is located on the livingroom_desk, while the breadslice is on counter_three, and the bananas are on the livingroom_coffeetable. The human can handle the items they are near, while the agent can assist by moving to grab items that the human cannot access directly.

To start the process, the agent should move to grab one of the objects. The closest item to the agent at kitchentable is the breadslice on counter_three.

Hence, the next logical action would be to:

1. Move to the counter_three to grab the breadslice.

So, I predict the next action as follows: [‘move(agent,counter_three)’]

Since the action selected by the LLM is feasible in this scenario, the ad hoc agent and human team perform the following actions:

$< human > [find] < coffeetable > \mid < agent > [find] < counter\_three >$

In the next three consecutive steps the LLM received similar prompts and selected to perform the following two actions: [‘grab(agent,breadslice)’] [‘move(agent,kitchentable)’] [‘put(agent,breadslice,kitchentable)’]

Thus the actions performance by human and agent were:

$< human > [grab] < bananas > \mid < agent > [grab] < breadslice >$
$< human > [find] < livingroom\_desk > \mid < agent > [find] < kitchentable >$
$< human > [grab] < cereal > \mid < agent > [putback] < breadslice >< kitchentable >$

However the next action anticipated by the LLM for the ad hoc agent was:

> In the current state, the human is holding both cereal and bananas, which means they cannot handle any additional objects. Since the goal involves switching on the computer and placing the book and cellphone on the livingroom desk, I need to focus on actions that can help achieve this goal. The computer is located at the livingroom desk, and I cannot carry any items for the human while the human is holding items.
>
> Since there's no action feasible for me to grab or put anything while the human is busy, one action I can perform immediately is to move to the livingroom desk to prepare for placing the items once the human is free.
>
> I will predict the next action based on this reasoning.
>
> Next action: ['move(agent,livingroom_desk)']

This action 'move(agent,livingroom_desk)' violated the 'Movement Limitation(Critical)' rule under 'Action Feasibility Rules', which required the agent to verify the adjacency of locations before attempting to move.

Since this action selected by the LLM was not feasible as the agent cannot move between places that are not adjacent, the agent followed the recovery mechanism and sent feedback to the LLM (as described in Section 4.1, we allowed up to three corrections per trial).

Following was the feedback that was added to the previous prompt and resent to the LLM for action selection.

> Your previous action move(agent,livingroom_desk) was not feasible since you can only move to adjacent locations. livingroom_desk and kitchentable are not next to each other. Please select a feasible action.
> Ensure the Action Feasibility Rules: Movement Limitation(Critical) is strictly respected.

This lead the LLM to provide the following output:

> Given the current state, the agent is at the kitchentable, and there are several items (including the book and cellphone) on the livingroom_coffeetable that need to be placed on the livingroom_desk. However, the agent cannot interact with the items in the human's possession (cereal) and cannot grab any items since they are not within the agent's immediate location.
>
> The next logical action is for the agent to move to a location that allows for future actions, such as grabbing the book, which is on the kitchen_smalltable. The kitchen_smalltable is adjacent to the kitchentable. Therefore, the agent should move to the kitchen_smalltable.
>
> The action I will predict is: ['move(agent,kitchen_smalltable)']

Since this actions was feasible the human and agent then continued their execution as follows: $< human > [find] < kitchen\_smalltable > | < agent > [find] < kitchen\_smalltable >$