Anticipate & Collab: Supplementary Material

Contents

1 Tasks

1.1 Regular Tasks

We created a dataset of high-level tasks in a household environment.

- prepare breakfast (options = healthy_bf, junk)
- prepare lunch (options = healthy_l, junk)
- prepare dinner (options = healthy_d, junk)
- prepare medicines
- wash the dishes
- do the laundry
- serve the food
- prepare clothes (options = casual, formal, gym, party)
- charge the electronic devices
- prepare the office bag
- clean the room (room options = livingroom, kitchen, bathroom, storeroom) (object options = vacuum cleaner, mop)
- set up the office table
- throw away leftover food
- put remaining food in the fridge
- decorate the place
- serve a drink
- dust electronic devices
- extinguish the fire

1.2 Anticipation Example

The LLM model Claude is used in this example. The example illustrated below shows an input of task list, user preference and the scene description where the prompt shows that the user is performing the task of preparing the office bag and LLM needs to figure out what should be the next 4 anticipated tasks. The output of the LLM shows a Chain-of-thought reasoning along with the set of tasks highlighting what it has anticipated and why.

```
# The following tasks are possible in the household
tasks_sample_space = ['prepare food', 'serve the food', 'serve a drink',
    'serve a drink (coke)', 'serve a drink (ice_tea)', 'serve a drink
    (sugary_fruit_juice)', 'serve a drink (milk)', 'serve a drink (coffee)',
    'serve a drink (tea)', 'prepare medicines', 'wash the dishes', 'do the
    laundry', 'prepare office clothes', 'prepare casual clothes', 'prepare gym
    clothes', 'charge the electronic devices', 'prepare the office bag',
    'clean the room (kitchen)', 'clean the room (bathroom)', 'clean the room
    (bedroom)', 'clean the room (livingroom)', 'set up the office table',
    'throw away leftover food', 'put remaining food in the fridge', 'decorate
    the place', 'dust electronic devices', 'extinguish the fire']

# The following tasks were done by **User 1** previously:
user_tasks = {'user 1': {'observation 1': ['prepare office bag', 'prepare
    office clothes', 'charge electronic devices', 'prepare food', 'serve the
    food'], 'observation 2': ['prepare casual clothes', 'prepare food', 'serve
    the food', 'prepare medicines', 'put remaining food in the fridge'],
    'observation 3': ['wash the dishes', 'clean the room (kitchen)', 'dust the
    electronic surfaces', 'clean the room (livingroom)'], 'description':
    {'observation 1': 'The user is a disciplined employee, who wakes up early
    morning to prepare for office. They prepare their office bag, office
    clothes, charge their electronic devices, followed by preparation of a
    healthy breakfast. Office clothes can only be prepared if they are clean,
    else the user does laundry before preparation.', 'observation 2': "In the
    evening, the user prepares their casual clothes (if they're clean). Then
    they prepare and serve a healthy dinner, and take their medicines. After
    that, they either store the leftover food back in the fridge, or throw it
    away.", 'observation 3': 'Before going to sleep, the user makes sure that
    the house is clean. They clean each room that is dirty, clean the dirty
    dishes, and dust the electronic devices.'}}}

It is morning time, the user has to go to the office.
 You see the user perform the task:
 *prepare the office bag*
 What do you anticipate to be the next 4 tasks?
 Requirement: The office clothes are dirty
Anticipate the next 4 tasks for the day.
Answer only as a valid python dictionary, with keys: 'tasks' and
    'chain-of-thought'. Keep tasks from the sample space.
```

The LLM gives the following output:

```
{   'chain-of-thought': "Since the user has prepared the office bag, we can
    anticipate the next tasks based on the morning routine described in
    'observation 1'. The next task would typically be to prepare office
    clothes, but the requirement states that the office clothes are dirty.
    Therefore, the user will first need to do the laundry before preparing the
    office clothes. After that, they will prepare the office clothes, charge
    their electronic devices, and then prepare and serve their breakfast.",
```

```
'tasks': ['do the laundry', 'prepare office clothes', 'charge the
    electronic devices', 'prepare food']}
```

1.3 Resource Availability

In case of situations where the LLM is prompted to complete a task of cooking *pasta* for dinner on a specific day but the resource is not available, then LLM figures out and provides another option of cooking *Chicken and rice* based on the availability of the item. We illustrate the example below

1.3.1 Input Prompt:

```
prompt = f'''
# The following tasks are possible in the household
tasks_sample_space = {task_sample_space}

# The following tasks were done in the last week:
task_last_week = {task_dict}

# For this week, these are the resources available:
Objects available: {objects}

# Input: Today is Thursday, Based on the resources available, anticipate the
    tasks for the day
'''
```

1.3.2 Output LLM Prompt:

```
{
"chain of thought": "Following last Thursday's routine, Checking the resource
    availability. Last Thursday, boiled egg was made for breakfast. Boiled egg
    is available. Last Thursday, mixed vegetable curry was made for lunch.
    Mixed vegetable curry is available. Preparing mixed vegetable curry for
    lunch. Last Thursday, pasta was made for dinner. Pasta is NOT
    AVAILABLE. Preparing chicken and rice for dinner.",
"tasks": [
    "Clean the office",
    "Make the bed",
    "Prepare tea",
    "Prepare breakfast (boiled egg)",
    "Wash the dishes",
    "Prepare lunch (mixed vegetable curry)",
    "Set up the office table",
    "Prepare dinner (chicken and rice)",
    "Prepare medicines"
]}
```

2 List of Actions and Associated Costs Defined in Domain Description

Actions	Cost
human_moves_br	Depends on human_dur
agent_moves_br	Depends on agent_dur
agent_pickup	20
agent_putdowns	20
human_picks	20
human_putdowns	20
human_picks_S	15
human_putdowns_S	15
agent_putdown	20
human_putdown	30
human_putdown_s	2
agent_picksUp_object	10
human_picksUp_object	20
agent_putsdown_object	10
human_putsdown_object	20
open	10
human_picksUp_freceptacle	5
human_putsdown_freceptacle	5
human_picksUp_receptacle	20
human_putsdown_receptacle	20
agent_picksUp_receptacle	10
agent_putsdown_receptacle	10
human_switches_on	15
agent_switches_on	10
human_switches_off	15
agent_switches_off	10
agent_switch_on	10
agent_switch_off	10
human_switch_on	10
human_switch_off	10
agent_cleans	25
human_cleans	75
agent_slice	15
human_slice	30
make_fruit_salad	80
human_cooks	50
agent_cooks	80
human_boils	30
agent_boils	15
human_roast	25
agent_roast	60
human_prepare_eggs	25
agent_prepare_eggs	80
agent_prepares_pizza_base	60

Actions	Cost
human_prepares_pizza_base	20
agent_bake_pizza	20
human_bake_pizza	60
agent_serves_pizza	20
human_serves_pizza	40
agent_serves_food	20
human_serves_food	40
agent_serves_fruit	20
human_serves_fruit	40
agent_serves_vegetable	20
human_serves_vegetable	40
bakeacake	90
agent_serves_baked	20
agent_serves_drink	20
human_serves_drink	10
remaining_food_cleaned	15
agent_washingdishes	30
human_washingdishes	10
agent_extinguish_fire	50
human_extinguish_fire	70
agent_washingclothes	40
human_washingclothes	60
agent_ironclothes	40
human_ironclothes	50
agent_foldclothes	30
human_foldclothes	50
laundry_done	10
agent_holds_vc_hose	10
human_holds_vc_hose	10
agent_starts_cleaning_	45
human_starts_cleaning_	50
agent_passes_to_human	5
agent_cleans_electronics	60
human_cleans_electronics	20
all_electronic_item_cleaned	0
house_cleaned	0
office_table_ready	0
agent_cleans_recepetacle	20
human_cleans_receptacle	25
prepare_office_bag	0
clothes_prepared	10
attached_for_charging	5
agent_provides_	5
party_starts	0

Table 1: Table describing actions with their corresponding costs defined in the domain description.

The Table 1 outlines the actions along with their respective costs as specified in the domain description file. The cost for movements between two points is determined by the time required for a human or agent to move from one point to the other. The costs associated with the action descriptions are normalized to a scale from 0 to 100.

3 HRI collaborative plan

Here, we describe an example where human-robot collaborative plan was generated. We show an example from the experimental setting in which the human is faster than the agent and thus has a lower movement cost than the agent. To reach the below goal state:

```
(:goal
    (and
     (salad_prepared bowl_1 sliced_apple sliced_avocado sliced_banana)
     (fruit_served fruit_salad bowl_2 working_table livingRoom))
)
```

while trying to optimize the "cost" metric, we obtain the following plan with a cost of 833:

```
(human_moves_br dining_table shelf kitchen kitchen)
(human_picksup_freceptacle bowl_1 shelf kitchen)
(human_picksup_freceptacle bowl_2 shelf kitchen)
(human_moves_br shelf fridge kitchen kitchen)
(human_picks apple fridge kitchen)
(human_picks avocado fridge kitchen)
(human_picks banana fridge kitchen)
(agent_switches_on faucet sink kitchen)
(human_moves_br fridge sink kitchen kitchen)
(human_putdowns apple sink kitchen)
(agent_cleans apple)
(human_picks apple sink kitchen)
(human_putdowns avocado sink kitchen)
(agent_cleans avocado)
(human_picks avocado sink kitchen)
(human_putdowns banana sink kitchen)
(agent_cleans banana)
(human_picks banana sink kitchen)
(agent_moves_br sink countertop kitchen kitchen)
(human_moves_br sink countertop kitchen kitchen)
(human_putsdown_freceptacle bowl_1 countertop kitchen)
(human_putdowns apple countertop kitchen)
(human_putdowns avocado countertop kitchen)
(human_putdowns banana countertop kitchen)
(agent_slice apple sliced_apple)
(agent_slice avocado sliced_avocado)
(agent_slice banana sliced_banana)
(agent_pickup sliced_apple countertop kitchen)
(agent_putdown sliced_apple bowl_1 countertop kitchen)
(agent_pickup sliced_avocado countertop kitchen)
(agent_putdown sliced_avocado bowl_1 countertop kitchen)
(agent_pickup sliced_banana countertop kitchen)
(agent_putdown sliced_banana bowl_1 countertop kitchen)
(make_fruit_salad bowl_1 sliced_apple sliced_avocado sliced_banana fruit_salad)
(human_picks fruit_salad countertop kitchen)
(human_moves_br countertop working_table kitchen livingroom)
(human_putsdown_freceptacle bowl_2 working_table livingroom)
(human_putdowns fruit_salad working_table livingroom)
(human_serves_fruit fruit_salad bowl_2 working_table livingroom)
```

4 PDDL plans

Below, we will showcase the *Adaptation* scenario and the scenario when there is *No-Adpatation*. For this experiment following goal state is used:

```
(:goal
  (and
  (served_drink milk glass_1 working_table livingroom)
  (prepared_clothes office_clothes ironing_board livingroom)
  )
)
```

1. **Adaptation**

   - ***Plan initially generated by the agent***[1]

```
(human_moves_br dining_table shelf kitchen kitchen)
(human_picksup_freceptacle glass_1 shelf kitchen)
(agent_pickup office_clothes closet livingroom)
(agent_moves_br closet ironing_board livingroom livingroom)
(agent_putdowns office_clothes ironing_board livingroom)
(clothes_prepared office_clothes ironing_board livingroom)
(human_moves_br shelf fridge kitchen kitchen)
(human_picks milk fridge kitchen)
(human_moves_br fridge oven kitchen kitchen)
(human_moves_br oven working_table kitchen livingroom)
(human_putsdown_freceptacle glass_1 working_table livingroom)
(human_putdowns milk working_table livingroom)
(human_serves_drink milk glass_1 working_table livingroom)
```

   - ***Plan that contains human's unexpected actions***

```
(human_moves_br dining_table shelf kitchen kitchen)
(human_picksup_freceptacle glass_1 shelf kitchen)
(agent_pickup office_clothes closet livingroom)
(agent_moves_br closet ironing_board livingroom livingroom)
(agent_putdowns office_clothes ironing_board livingroom)
(clothes_prepared office_clothes ironing_board livingroom)
(human_moves_br shelf fridge kitchen kitchen)
(human_picks apple fridge kitchen) \\ UNEXPECTED ACTION
(human_moves_br fridge oven kitchen kitchen)
(human_moves_br oven working_table kitchen livingroom)
(human_putsdown_freceptacle glass_1 working_table livingroom)
(human_putdowns apple working_table livingroom) \\ UNEXPECTED ACTION
(human_putdowns milk working_table livingroom) \\ NOT APPLICABLE
(human_serves_drink milk glass_1 working_table livingroom) \\ NOT APPLICABLE
```

[1] This plan comprises actions for the agent and those that the agent expects the human will perform.

- ***Agent adapts to human's unexpected actions***

```
(human_moves_br dining_table shelf kitchen kitchen)
(human_picksup_freceptacle glass_1 shelf kitchen)
(agent_pickup office_clothes closet livingroom)
(agent_moves_br closet ironing_board livingroom livingroom)
(agent_putdowns office_clothes ironing_board livingroom)
(clothes_prepared office_clothes ironing_board livingroom)
(human_moves_br shelf fridge kitchen kitchen)
(human_picks apple fridge kitchen)
(human_moves_br fridge oven kitchen kitchen)
(human_moves_br oven working_table kitchen livingroom)
(human_putsdown_freceptacle glass_1 working_table livingroom)
(human_putdowns apple working_table livingroom)
(human_putdowns milk working_table livingroom) - NOT_APPLICABLE
(human_serves_drink milk glass_1 working_table livingroom) - NOT_APPLICABLE
(agent_moves_br ironing_board fridge livingroom kitchen)
(agent_pickup milk fridge kitchen)
(agent_moves_br fridge oven kitchen kitchen)
(agent_moves_br oven working_table kitchen livingroom)
(agent_putdowns milk working_table livingroom)
(human_serves_drink milk glass_1 working_table livingroom)
```

2. **No-Adaptation**

- ***When there is no adaptation to human's unexpected behavior***

```
(human_moves_br dining_table shelf kitchen kitchen)
(human_picksup_freceptacle glass_1 shelf kitchen)
(agent_pickup office_clothes closet livingroom)
(agent_moves_br closet ironing_board livingroom livingroom)
(agent_putdowns office_clothes ironing_board livingroom)
(clothes_prepared office_clothes ironing_board livingroom)
(human_moves_br shelf fridge kitchen kitchen)
(human_picks apple fridge kitchen)
(human_moves_br fridge oven kitchen kitchen)
(human_moves_br oven working_table kitchen livingroom)
(human_putsdown_freceptacle glass_1 working_table livingroom)
(human_putdowns apple working_table livingroom)
(human_putdowns milk working_table livingroom) \\ NOT_APPLICABLE
(human_serves_drink milk glass_1 working_table livingroom) \\ NOT_APPLICABLE
```

  - Required Preconditions for the given goal state:

```
- stuff_at office_clothes ironing_board livingroom
- clothes_prepared office_clothes ironing_board livingroom
- receptacle_at glass_1 working_table livingroom
- served_drink milk glass_1 working_table livingroom
```

  - Preconditions met in No-Adaptation case:

```
- stuff_at office_clothes ironing_board livingroom
- clothes_prepared office_clothes ironing_board livingroom

Success % = 50%  - Success rate decreases in case of no-adpatation.
```

5 Simulation Details

Figure 1: Our Domain Environment

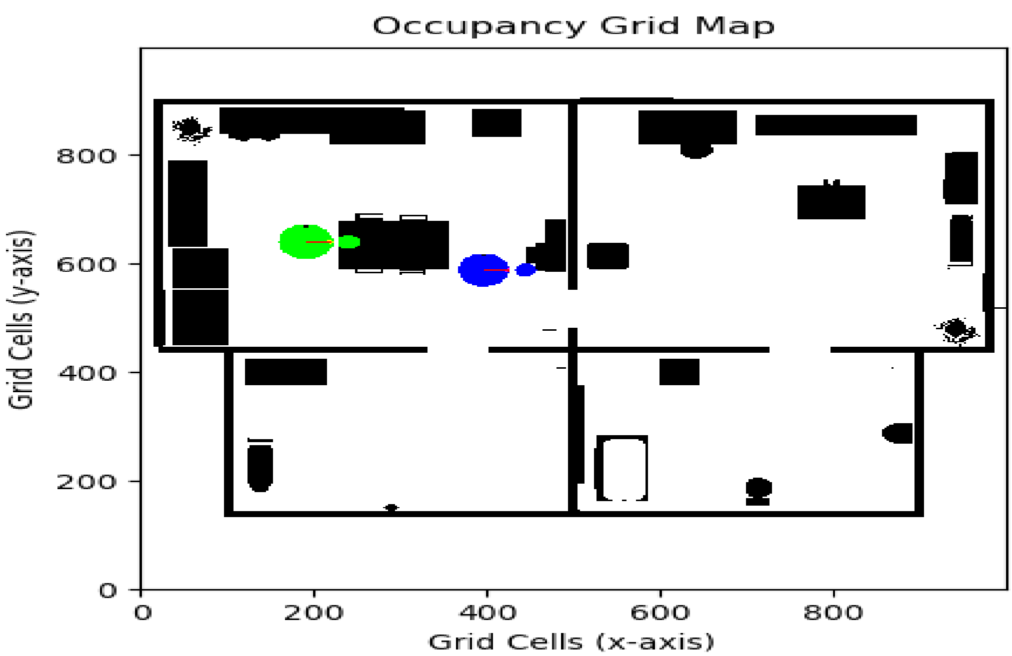

Figure 2: The tracking of human (represented in blue) and agent (represented in green) in the simulation environment

Fig. 1 is our customly made household environment in the Coppeliasim Simulator. Actions from the generated plan are fed to the actors via the Coppeliasim API, with action outcomes encoded in local child scripts provided by the simulator. Fig. 2 illustrates how the agent and the human are tracked in the simulation. This helps us in integrating the Collision avoidance stack.

6 Results

	ROBOT									
		Init →	Livingroom		Storeroom		Bathroom		Kitchen	
	Heuristics	Init ↓	ζ_{HFAS}	ζ_{AFHS}	ζ_{HFAS}	ζ_{AFHS}	ζ_{HFAS}	ζ_{AFHS}	ζ_{HFAS}	ζ_{AFHS}
HUMAN	seq-sat-fdss-2018 (90 sec)	Livingroom	1.143	1.171	1.09	**0.972**	1.058	1.134	1.063	1.009
		Storeroom	1.155	1.066	1.238	1.135	1.149	1.149	1.167	1.185
		Bathroom	1.196	1.108	1.162	1.053	1.203	1.126	1.23	1.163
		Kitchen	1.202	1.105	1.169	1.025	1.139	1.045	1.176	1.112
	seq-sat-fd-autotune-1 (90 sec)	Livingroom	1.039	**0.92**	1.016	**0.909**	0.906	**0.9**	0.806	**0.914**
		Storeroom	1.024	**0.987**	1.089	1.005	1.052	**0.997**	1.055	1.021
		Bathroom	1.013	**0.948**	1.043	**0.949**	1.067	**0.946**	1.025	**0.936**
		Kitchen	1.038	**0.982**	1.05	**0.981**	1.043	**0.968**	1.091	1.026

Table 2: Computed ζ as the ratio of execution time with collaboration over the execution time without collaboration. Each value in table is average over 50 trials (with two of more tasks in the goal state) for each of 16 possible initial locations of the agent and the human, and for each of two settings: HFAS and AFHS. Results indicate a clear benefit of human-agent collaboration in terms of reduction in execution time, thus supporting **H3**.

The Table 2 is an extension to the Hypothesis 3 in the paper. In short, we evaluate the execution cost between collaboration and no-collaboration scenario and claim that the collaboration between the human and the agent reduces the execution cost and leads to faster completion of tasks. We used different configurations provided by the *Fast-Downward* system namely *lama*, *seq-sat-fdss-2018* and *seq-sat-fd-autotune-1*. We observe that *lama* outperforms other aliases and here in the above table we showcase the results from different configurations.