Anticipate & Act: Supplementary Material

Contents

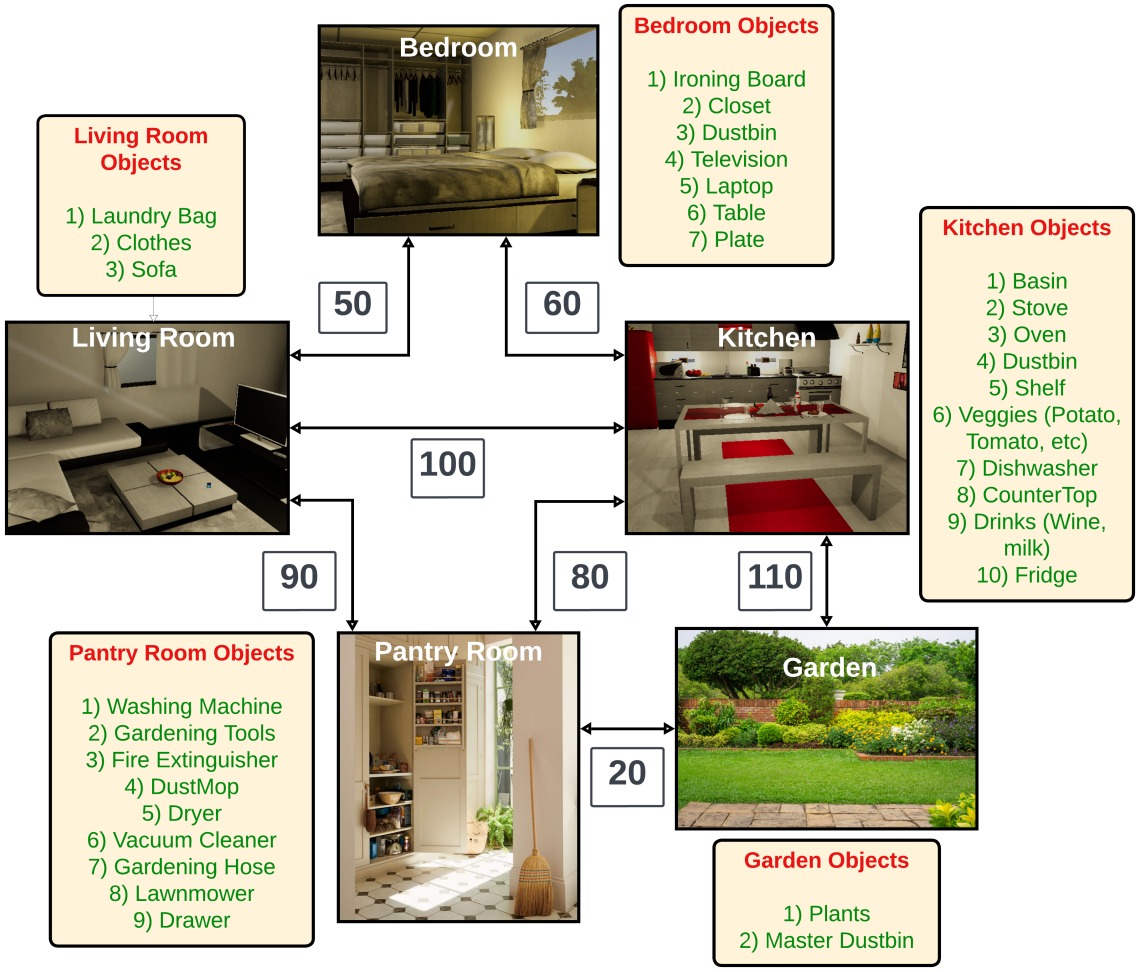

Figure 1: Our Domain Environment

PDDL Domain – Time of actions

Action	Time (sec)	Action	Time (sec)
PickUp	5	Collect clothes	10
Switch_On	1	Open (washing machine)	15
Switch_off	1	Cut the grass	100
PutDown	5	Washing Clothes	200
Clean edibles (washing apple, etc)	8		
Slice	20	Water plants	20
Cook	120	Dry clothes	120
Washing Dishes	20	Iron clothes	60
Bake	120	Fold clothes	10
Serve	8		
Throw away remaining food	3		

Time for moving the agent:

From	To	Time (sec)
livingRoom	Pantry	90
livingRoom	Kitchen	100
livingRoom	Garden	110
livingRoom	Bedroom	50
Pantry	Kitchen	80
Pantry	Garden	20
Pantry	Bedroom	140
Kitchen	Garden	110
Kitchen	Bedroom	60
Garden	Bedroom	170

Figure 2: The cost(in the form of time in secs) for each action is considered and the time for the movement from one room to another corresponding to the Room Map is given in the table

1 Activities

1.1 Regular Activities

We created a dataset $\mathcal{T}$ of high-level tasks in a household environment. These tasks belong to activities such as *Cooking*, *Cleaning*, *Washing*, *Baking*, and *Gardening*. We then generated a set of task routines $\mathcal{R}_i$, each with ≈ 20 tasks, by sampling tasks across different activities while preserving the relative order of tasks within each activity. We follow a pipelining approach while creating the routines, where long duration tasks are pipelined with shorter duration tasks. This methodology allowed us to create task routines spanning activities of daily living in a home. The dataset of activities is defined as follows:

- **Cooking:** Get a vegetable, Clean the vegetable, Cut the vegetable, Cook the vegetable, Serve food, Take out leftover food
- **Washing clothes:** Collect dirty clothes, Load dirty clothes in the washing machine, Run the washing machine, Collect clean clothes, Dry the clean clothes, Iron the dry clothes, Fold the clothes, Put away the clean clothes
- **Cleaning the house:** Collect cleaning supplies, Dust the surfaces, Vacuum the floors, Put away the cleaning supplies, Take out the trash
- **Washing dishes:** Collect dirty dishes, Load dirty dishes in the dishwasher, Run the dishwasher, Put away the clean dishes
- **Gardening:** Cut the grass, Water the plants, Put away the gardening tools
- **Baking:** Preheat the oven, Prepare baking ingredients, Pour the batter into the baking pan, Bake in the oven, Serve cake

Here, we measure the accuracy of any LLM to complete tasks belonging to a given activity, while maintaining the relative order of tasks within an activity.

1.1.1 Illustrative Example

Let's examine a specific instance:

```
Input: Collect dirty dishes, Get a vegetable
```

The Large Language Model (LLM) generates the following sequence of tasks:

```
LLM Output: Collect dirty dishes, Load dirty dishes in the dishwasher, Run the dishwasher,
Get a vegetable, Clean the vegetable, Put away the clean dishes, Cut the vegetable,
Cook the vegetable, Serve food, Take out leftover food.
```

This sequence showcases the LLM's ability to not only follow the order of input tasks but also logically complete each activity. Notably, it recognizes that 'Run the dishwasher' is a long-duration task and intelligently pipelines tasks from other activities (in this case, tasks related to 'Cooking') during this period. This allows for efficient task management, and we see that LLMs are able to understand this pattern of pipelining.

1.2 Special Sequence: Weekly Routine

We also created a dataset W of weekly routines. These routines are a sequence of tasks that are performed on specific days of the week. The dataset is defined as follows: Sure, here's how you can complete the list of weekly tasks in LaTeX:

- **Monday:** Cook omelette, Serve the food, Collect dirty dishes, Load dirty dishes in the dishwasher, Run the dishwasher, Put away the clean dishes, Dust the surfaces, Cut the grass, Water the plants, Put away the gardening tools
- **Tuesday:** Cook cereal, Serve the cereal, Collect dirty clothes, Load dirty clothes in the washing machine, Run the washing machine, Collect clean clothes, Dry the clean clothes, Iron the dry clothes, Fold the clothes, Put away the clean clothes, Collect dirty dishes, Load dirty dishes in the dishwasher, Run the dishwasher, Put away the clean dishes
- **Wednesday:** Cut a mango, Cut an apple, Serve the fruits, Prepare a coffee, Serve the coffee, Collect cleaning supplies, Dust the surfaces, Vacuum the floors, Put away the cleaning supplies, Take out the trash, Collect dirty clothes, Load dirty clothes in the washing machine, Run the washing machine
- **Thursday:** Get a vegetable, Clean the vegetable, Cut the vegetable, Cook the vegetable, Serve food, Prepare milk, Serve milk, Take out leftover food, Collect dirty dishes, Load dirty dishes in the dishwasher, Run the dishwasher, Put away the clean dishes, Cut the grass, Water the plants, Put away the gardening tools

There are some extra tasks are as follows: Extinguish the fire, Set up laptop, Repair the washing machine, Repair the dishwasher, Repair the oven, Serve wine, Clean windows

In this case, we measure how accurately an LLM can complete tasks for a given day while considering any specific prompt given by the user. The prompts by user change the usual routine for a day by either removing some tasks, or replacing some existing tasks with a new task.

1.2.1 Illustrative Example

Let's examine a specific instance:

```
Input: Today is Monday, I have an urgent meeting in the morning.
```

The LLM outputs as following:

```
LLM Output: Cook omelette, Serve the food, Set up Laptop, Collect dirty dishes,
Load dirty dishes in the dishwasher, Run the dishwasher, Put away the clean dishes,
Dust the surfaces, Cut the grass, Water the plants, Put away the gardening tools
```

2 PDDL plans

Here, we will present some examples of plans created by our agent *with Anticipation* and *without Anticipation*.

1. **Plan to serve breakfast in the morning. Breakfast contains Veggies, Cereal, and fruits.**

   - *Plan without anticipation*

```
(move_agent bedroom kitchen)
(switch_on burner kitchen)
(pickup veggie fridge kitchen)
(putdown veggie basin kitchen)
(clean_edible veggie)
(pickup veggie basin kitchen)
(putdown veggie stove kitchen)
(cook veggie veggie)
(pickup veggie stove kitchen)
(move_agent kitchen bedroom)
(putdown veggie plate bedroom)
(serve_vegetable veggie bedroom)
; cost = 287 (general cost)

(move_agent bedroom kitchen)
(pickup cereal shelf kitchen)
(putdown cereal stove kitchen)
(switch_on burner kitchen)
(cook cereal cooked_cereal)
(pickup cooked_cereal stove kitchen)
(move_agent kitchen bedroom)
(putdown cooked_cereal plate bedroom)
(serve_food cooked_cereal bedroom)
; cost = 269 (general cost)

(move_agent bedroom kitchen)
(pickup apple fridge kitchen)
(putdown apple basin kitchen)
(clean_edible apple)
(pickup apple basin kitchen)
(putdown apple countertop kitchen)
(slice apple sliced_apple)
(pickup sliced_apple countertop kitchen)
(move_agent kitchen bedroom)
(putdown sliced_apple plate bedroom)
(serve_fruit sliced_apple bedroom)
; cost = 186 (general cost)

; Total cost = 742 (general cost)
```

- Plan with anticipation

```
(move_agent bedroom kitchen)
(switch_on burner kitchen)
(pickup cereal shelf kitchen)
(putdown cereal stove kitchen)
(cook cereal cooked_cereal)
(pickup veggie fridge kitchen)
(putdown veggie basin kitchen)
(clean_edible veggie)
(pickup veggie basin kitchen)
(putdown veggie stove kitchen)
(cook veggie veggie)
(pickup apple fridge kitchen)
(putdown apple basin kitchen)
(clean_edible apple)
(pickup apple basin kitchen)
(putdown apple countertop kitchen)
(slice apple sliced_apple)
(pickup veggie stove kitchen)
(pickup cooked_cereal stove kitchen)
(pickup sliced_apple countertop kitchen)
(move_agent kitchen bedroom)
(putdown veggie plate bedroom)
(putdown cooked_cereal plate bedroom)
(putdown sliced_apple plate bedroom)
(serve_food cooked_cereal bedroom)
(serve_vegetable veggie bedroom)
(serve_fruit sliced_apple bedroom)

; Total cost = 501 (general cost)
```

2. Plan of serving cereal and sliced fruit and then cleaning the remaining food and fruit.

- Plan without anticipation

```
(move_agent bedroom kitchen)
(pickup cereal shelf kitchen)
(putdown cereal stove kitchen)
(switch_on burner kitchen)
(cook cereal cooked_cereal)
(pickup cooked_cereal stove kitchen)
(move_agent kitchen bedroom)
(putdown cooked_cereal plate bedroom)
(serve_food cooked_cereal bedroom)
; cost = 269 (general cost)

(move_agent bedroom kitchen)
(pickup apple fridge kitchen)
```

```
(putdown apple basin kitchen)
(clean_edible apple)
(pickup apple basin kitchen)
(putdown apple countertop kitchen)
(slice apple sliced_apple)
(pickup sliced_apple countertop kitchen)
(move_agent kitchen bedroom)
(putdown sliced_apple plate bedroom)
(serve_fruit sliced_apple bedroom)
; cost = 186 (general cost)

(pickup remaining_food plate bedroom)
(move_agent bedroom kitchen)
(putdown remaining_food dustbin_1 kitchen)
(cleaned_remaining_food remaining_food bedroom)
; cost = 73 (general cost)

(pickup remaining_fruit plate bedroom)
(move_agent bedroom kitchen)
(putdown remaining_fruit dustbin_1 kitchen)
(cleaned_remaining_food remaining_fruit bedroom)
; cost = 73 (general cost)

; Total cost = 601 (general cost)
```

- Plan with anticipation

```
(move_agent bedroom kitchen)
(switch_on burner kitchen)
(pickup cereal shelf kitchen)
(putdown cereal stove kitchen)
(pickup apple fridge kitchen)
(putdown apple basin kitchen)
(cook cereal cooked_cereal)
(clean_edible apple)
(pickup apple basin kitchen)
(putdown apple countertop kitchen)
(slice apple sliced_apple)
(pickup cooked_cereal stove kitchen)
(pickup sliced_apple countertop kitchen)
(move_agent kitchen bedroom)
(putdown cooked_cereal plate bedroom)
(putdown sliced_apple plate bedroom)
(serve_fruit sliced_apple bedroom)
(serve_food cooked_cereal bedroom)
(pickup remaining_food plate bedroom)
(pickup remaining_fruit plate bedroom)
(move_agent bedroom kitchen)
(putdown remaining_food dustbin_1 kitchen)
(putdown remaining_fruit dustbin_1 kitchen)
(cleaned_remaining_food remaining_food bedroom)
```

```
(cleaned_remaining_food remaining_fruit bedroom)

;Total cost = 421 (general cost)
```

3. **Clearing the trash in the house and then cutting the grass in the garden followed by watering the plants.**

   - Plan without anticipation

```
(pickup trash_2 dustbin_2 bedroom)
(move_agent bedroom kitchen)
(pickup trash_1 dustbin_1 kitchen)
(move_agent kitchen pantry)
(move_agent pantry garden)
(putdown trash_2 master_dustbin garden)
(putdown trash_1 master_dustbin garden)
(trash_cleared )
; cost = 182 (general cost)

(move_agent garden pantry)
(pickup_object lawnmower pantry)
(move_agent pantry garden)
(putdown_object lawnmower garden)
(switch_on lawnmower garden)
(cutting_the_grass )
; cost = 151 (general cost)

(move_agent garden pantry)
(pickup_object watering_hose pantry)
(move_agent pantry garden)
(putdown_object watering_hose garden)
(switch_on watering_hose garden)
(pickup_object watering_hose garden)
(water_the_plants )
; cost = 76 (general cost)

; Total cost = 409 (general cost)
```

   - Plan with anticipation

```
(pickup trash_2 dustbin_2 bedroom)
(move_agent bedroom kitchen)
(pickup trash_1 dustbin_1 kitchen)
(move_agent kitchen pantry)
(pickup_object watering_hose pantry)
(pickup_object lawnmower pantry)
(move_agent pantry garden)
(putdown_object lawnmower garden)
(putdown trash_2 master_dustbin garden)
(putdown trash_1 master_dustbin garden)
(trash_cleared )
```

```
(switch_on lawnmower garden)
(cutting_the_grass )
(putdown_object watering_hose garden)
(switch_on watering_hose garden)
(pickup_object watering_hose garden)
(water_the_plants )

; Total cost = 329 (general cost)
```

3 Baseline

As a baseline for the task anticipation capabilities of LLM, we sample future tasks using a simple Markov chain. For this Markovian baseline, we sampled 100 routines of tasks and created a probability transition matrix representing the likelihood of transitioning from one task to another within the dataset. A representation of the transition probabilities for regular activities is shown in 3. Subsequently, we established a Markov chain for the tasks, where starting from an initial task, successive tasks are generated through iterative sampling from the probability transition matrix.

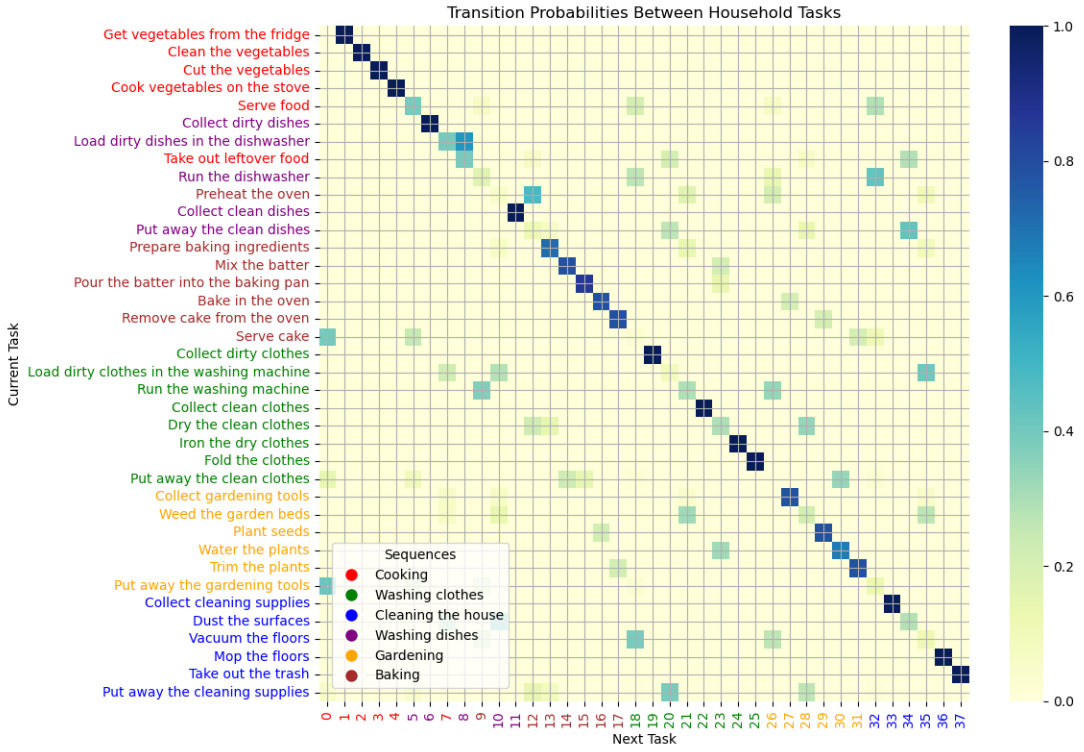

Figure 3: Markovian baseline computing transitional probabilities of tasks

4 Results

		Number of anticipated tasks			
Task	**Task Name**	**0**	**1**	**3**	**7**
T1	serve veggie	319	392	654	1088
T2	clean remaining food	73			
T3	clean dishes	31	272		
T4	clean house	241			
T5	dust sofa	285	447	604	
T6	clear trash	232			
T7	cutting plants	151	257		
T8	watering plants	76			
Total Cost		**1408**	**1368**	**1258**	**1088**

Table 1: Example table showing a particular day's activity

The Table 1 is an example of execution of low-level tasks that showcases a comparison between the myopic task and the sequentially increasing number of anticipated tasks in the way of execution. As the horizon for anticipation becomes longer, more and more tasks can be parallelized, hence reducing the execution costs overall.