DaTAPlan: Data-driven Task Anticipation and Knowledge-driven Planning for Human-robot Collaboration

Author Names Omitted for Anonymous Review. Paper-ID [add your ID here]

***Abstract*—An agent assisting humans in daily living activities can collaborate more effectively by anticipating upcoming tasks. Data-driven methods represent the state of the art in task anticipation, planning, and related problems, but these methods are resource-hungry and opaque. Our prior work introduced a proof of concept framework that used an LLM to anticipate 3 high-level tasks that served as goals for a classical planning system that computed a sequence of low-level actions for the agent to achieve these goals. This paper describes DaTAPlan, our framework that significantly extends our prior work toward human-robot collaboration. Specifically, DaTAPlan's planner computes actions for an agent and a human to collaboratively and jointly achieve the tasks anticipated by the LLM, and the agent automatically adapts to unexpected changes in human action outcomes and preferences. We evaluate DaTAPlan's capabilities in a realistic simulation environment, demonstrating accurate task anticipation, effective human-robot collaboration, and the ability to adapt to unexpected changes.**

Project website: https://dataplan-hrc.github.io

***Index Terms*— Task anticipation, Large Language Models, classical planning, human-agent collaboration.**

I. Introduction

Consider a human getting ready to leave from home for work. This involves completing some high-level tasks, e.g., cooking breakfast and serving it at the table in Figure 1. Each of these tasks requires the execution of a sequence of actions, e.g., fetch and boil the egg to cook breakfast, and bring the cooked egg and juice to the table. There is a robot (*agent*†) that can assist in completing these tasks. The tasks can be completed more effectively by anticipating the upcoming tasks, with the agent and the human executing actions to collaboratively and jointly complete all these tasks with minimal effort. This happens in Figure 1(a), with the agent (in green) anticipating the serving task, fetching juice from the fridge to the table while fetching the egg that the human (in blue) cooks in a metal pot and (not shown here) brings to the table. There is no such collaboration in Figure 1(b), with the agent fetching juice to the table while the human fetches and boils the egg before bringing it to the table. Furthermore, actions may have unexpected outcomes, and changes in human preferences (e.g., the human decides to work from home) may change the tasks to be completed.

Data-driven methods and models are the state of the art for task anticipation and human-robot collaboration. These methods are resource-hungry, i.e., need considerable computation and training examples, and are opaque, i.e., it is difficult to understand their internal processes. Our recent work provided a basic demonstration of an agent in a household scenario (with no other actors) using a pretrained Large Language Model (LLM) for task anticipation and using classical planning to compute a sequence of finer-granularity actions to jointly achieve these tasks [1]. In this paper, we significantly extend this work to describe a framework (DaTAPlan) that combines data-driven task anticipation and knowledge-driven planning for human-robot collaboration. The key characteristics of DaTAPlan are:

†We use the terms ”robot” and ”agent” interchangeably.

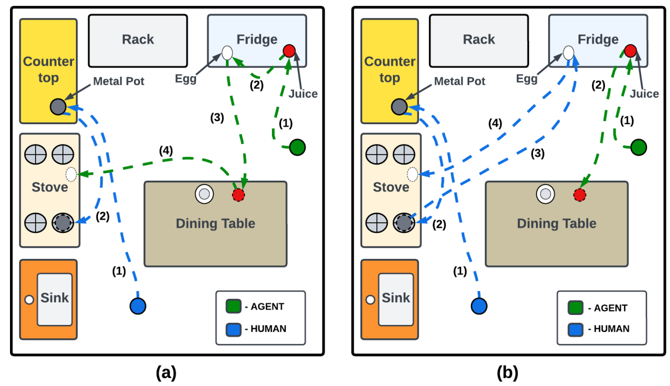


Fig. 1: Illustration of ”human-robot collaboration with anticipation”: (a) agent anticipates (serving task) and collaborates with human, fetching juice from the fridge to the table while fetching the egg that the human cooks in a metal pot and brings to the table; (b) The agent only serves the juice to the table and the human entirely performs the necessary actions needed to cook and serve the egg.

1) A pretrained LLM predicts a list of anticipated tasks based on a small number of prompts comprising a partial sequence of tasks in specific user scenarios.
2) A classical planner reasons with prior knowledge of the action theories of the agent and the human to compute a plan of finer-granularity actions that the agent executes, and the human is expected to execute, to collaboratively achieve the identified list of tasks.
3) If the human's action choices, action outcomes, or preferences deviate from expectations as the agent executes its actions, the agent automatically adapts by replanning or generating new task predictions.

The novelty is in: (a) supporting adaptation to different task patterns with limited prompting to LLMs; (b) computing a plan to jointly achieve the anticipated high-level tasks such that the agent and the human execute actions to collaboratively complete each task; and (c) automatically adapting to unexpected changes in the action outcomes or preferences of human. We use the Planning Domain Definition Language [2] and the Fast Downward solver [3] for classical planning. We evaluate our framework's capabilities using household scenarios (with multiple tasks, rooms, and objects) in the realistic CoppeliaSim environment.

II. Related Work

State of the art research in *human-robot collaboration* focuses on teleoperation, shared autonomy, or collaboration [4, 5, 6]. There have been promising breakthroughs in perception, learning, task planning, and adaptive control[7, 8, 9, 10], and work on improving collaborative interactions that enhance efficiency and quality of life[11, 12, 13, 14]. Different data-driven formulations represent the state of the art for human-robot collaboration and task anticipation[15, 16]. However, safety, trust, adaptability, real-time perception, and integration of human feedback remain open problems[17]. Planning Domain Definition Language (PDDL) has been widely used to encode prior knowledge for planning problems[2]. Within automated task planning methods, the process of defining the planning problem and prior domain knowledge (i.e., domain models) is labor-intensive and relies on closed-world assumptions, limiting adaptability to dynamic environments. There has been considerable research in learning the domain models for planning, with more recent work using LLMs for this purpose[18]. LLMs have also been used to produce goal states achievable by classical PDDL-based planners [19, 20]. and for generating diverse plans or translating natural language to structured planning problems[21, 22]. However, integrating learning methods with planning systems while ensuring that sound and executable plans are produced, remains challenging[23, 24]. Recent studies have also used LLMs for task planning in complex domains[25, 26, 27, 28], including scene rearrangement[29], but there is a growing body of research to show that LLMs are not really appropriate for planning in the classical sense[18]. There is well-established research in monitoring action outcomes and adapting to unexpected outcomes[30]. Recent work in human-robot collaboration uses human behavior models to detect unexpected behavior and has the agent replan to prevent failure[31]. Even systems that use LLMs for planning include a method for detecting collisions or hardware failures[21], or use probabilistic sequential decision-making for monitoring[32]. Our framework enables the agent to adapt if the human's action choices, action outcomes, or preferences deviate from expectations. Our prior work enabled an agent in a household scenario to use a pre-trained LLM for task anticipation, with classical planning used to compute and execute a sequence of finer-granularity actions to jointly achieve these tasks[1]. Here we extend this approach to human-robot collaboration, with planning directing the agent and the human to collaboratively execute actions to complete each task and enabling the agent to adapt when the human deviates from the plan.

III. Problem Formulation and Framework

Consider a household with two actors, *human* ($\mathcal{H}$) and *agent/robot* ($\mathcal{R}$). The objective is to complete a routine of high-level tasks $\mathcal{Q} = \{\tau_1, \tau_2, ..., \tau_n\}$ although the entire routine is not known in advance and can change over time. Completion of each $\tau_i \in \mathcal{T}$, a known list of high-level tasks such as *cook breakfast* and *do the laundry*, requires a sequence of finer-granularity actions $\{a_1, a_2, ..., a_k\}$ to be executed, e.g., to *cook breakfast*, it is necessary to *go to the fridge*, *fetch egg to the stove*, and *boil the egg*. Since $\mathcal{Q}$ can change over time, an actor usually tries to complete one task at a time at minimum cost (or time, effort). However, the tasks can be completed more efficiently if $\mathcal{R}$ and $\mathcal{H}$ anticipate upcoming tasks and collaborate to complete them. The agent has an action theory, $\mathcal{M}_\mathcal{R}$, describing preconditions and effects of its actions, and a similar theory $\mathcal{M}_\mathcal{H}$ describing its expectations of human behavior. There is no explicit communication between $\mathcal{R}$ and $\mathcal{H}$ and we only control $\mathcal{R}$'s action choices. Actions can be non-deterministic but the domain state is assumed to be fully observable.

Figure 2 outlines our framework, DaTAPlan, which combines data-driven task anticipation and knowledge-driven planning (with monitoring) for human-robot collaboration. In Figure 2 (a), an LLM is prompted with a partial task sequence to obtain a sequence of anticipated tasks. These tasks become goals in a PDDL problem file in Figure 2(b). A classical planner uses this problem description and a domain description to compute a joint plan of low-level actions for the agent and expected actions for the human—Figure 2(c). The execution of these actions is simulated in CoppeliaSim [33] in Figure 2(d). The agent monitors any deviation from the plan and adapts accordingly. We describe the individual components of DaTAPlan below.

A. LLM-based Task Anticipation

A pretrained LLM is tuned with two types of prompts: (i) *few shot*; and (ii) *chain of thought*. In both cases, the inputs include $\mathcal{T}$ and a JSON scene description. In the former, the input also includes 2-3 prior observations of user task patterns—see Figure 3;, whereas the latter considers two in-context examples [34] with step-by-step guidance [35] to understand user patterns. In both cases, the output is a sequence of anticipated tasks, with hallucinated tasks outside $\mathcal{T}$ not used for planning. Please see our project website for more examples. We show later (Section IV-B) that the LLM's task predictions match human expectations.

Apart from anticipating future tasks, the LLM also receives user feedback (during execution) if there is a change in tasks from a usual pattern, e.g., if the user shares that *guests are coming over*, then LLM is re-prompted to retrieve a fresh set of anticipated tasks.

B. Task and Motion Planning

The sequence of anticipated high-level tasks $\mathcal{Q}_A = \{\tau_1, \tau_2, ...\tau_n\}$ from the LLM is mapped to goal state $\mathcal{G}$ in the planning framework—see Figure 2(b). The next component of the framework computes a sequence of finer-granularity actions to be executed by $\mathcal{R}$ and $\mathcal{H}$ to achieve $\mathcal{G}$.

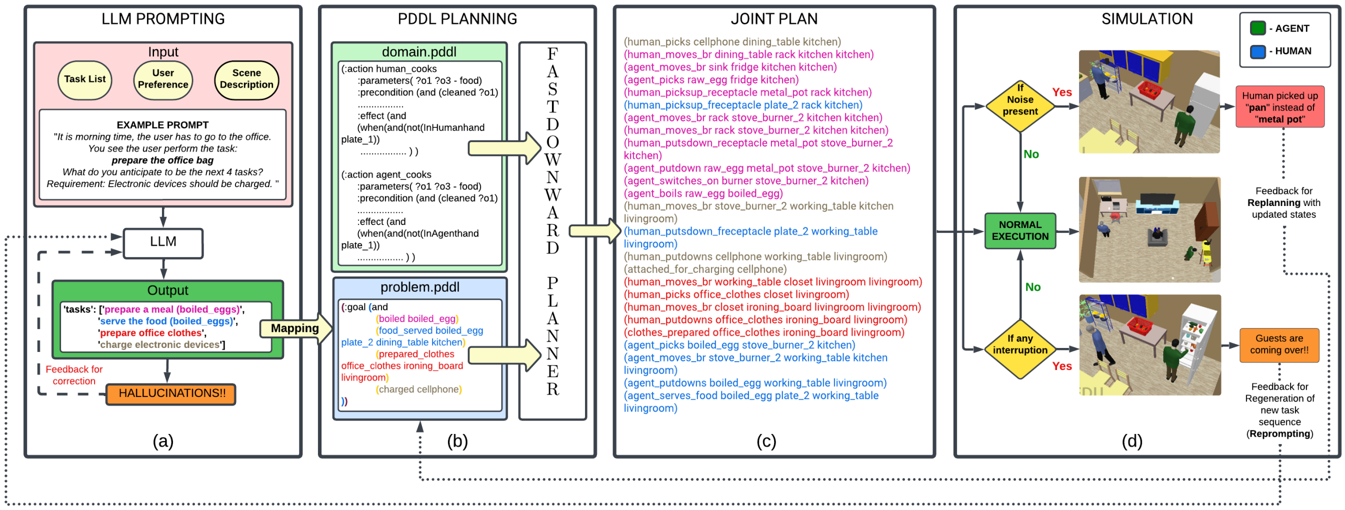


Fig. 2: Our framework's pipeline: (a) Input prompt contains the list of possible tasks, user preferences, and scene description, along with an example prompt and the corresponding output high-level tasks; (b) High-level tasks predicted by LLM are mapped to PDDL problem description; (c) The FD planner generates a plan of agent's actions and the expected human actions; (d) Deviations of the human from the expected plan are noted and used to trigger replanning when appropriate.

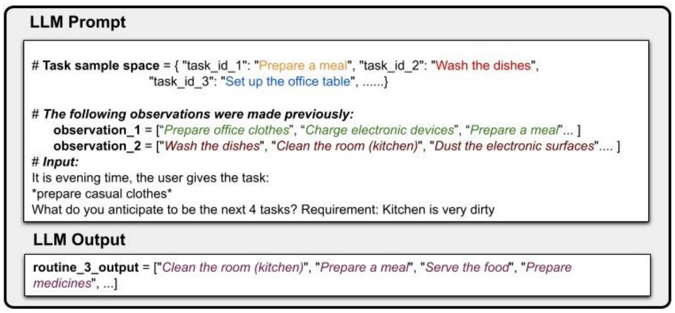


Fig. 3: Few shot prompting with LLMs.

Task planning. To generate a sequence of actions to achieve $\mathcal{G}$, the `domain` and `problem` are created in PDDL. The domain description $\mathcal{D} = \langle \mathcal{D}_\mathcal{R}, \mathcal{D}_\mathcal{H} \rangle$ comprises the domain descriptions of the agent ($\mathcal{D}_\mathcal{R}$) and the human ($\mathcal{D}_\mathcal{H}$). $\mathcal{D}_\mathcal{R} = \langle \mathcal{S}_\mathcal{R}, \mathcal{M}_\mathcal{R} \rangle$ comprises signature $\mathcal{S}$ and action theory $\mathcal{M}_\mathcal{R}$. $\mathcal{S}$ includes *types*, *constants*, and *predicates*. *Predicates* include *fluents*, which can change over time due to *actions*; and *statics*, which remain unchanged. For instance, in Figure 2(c), actions such as *human_picks* and *agent_picks* modify fluent *obj-at* that determines the location of objects. $\mathcal{M}_\mathcal{R}$ specifies each action (of the agent) in terms of its parameters, preconditions for the action to be executed, effects that will be true once the action is executed, and the action's cost. In a similar manner, $\mathcal{D}_\mathcal{H} = \langle \mathcal{S}_\mathcal{H}, \mathcal{M}_\mathcal{H} \rangle$, is the agent's estimate of the human's domain description. Here, we assume that these descriptions are accurate; the revision of this description is left to future work.

The `problem` $\mathcal{P} = \langle \mathcal{O}, \mathcal{I}, \mathcal{G} \rangle$ describes a specific scenario in terms of a set of specific objects ($\mathcal{O}$), the initial state $\mathcal{I}$ comprising ground literals of the *fluents* and *statics*, and a goal description $\mathcal{G}$ in the form of relevant ground literals. To create suitable scenarios, we designed a complex household domain with a range of actions and objects (72 predicates and 88 actions); *this is more complex than commonly used planning benchmarks and our prior work*. $\mathcal{I}$ is estimated from sensor inputs and $\mathcal{G}$ includes the anticipated tasks. Example problem and domain descriptions are available in our open-source project website.

The planning task is to compute a sequence of actions $\pi = \langle a_1, \dots, a_K \rangle$ that takes the system from $\mathcal{I}$ to a state where $\mathcal{G}$ is satisfied. Some actions in this sequence are to be executed by the agent, while the human is expected to execute the other actions. To compute the plan, $\mathcal{D}$ and $\mathcal{P}$ are given to the planner that tries to minimize the total cost of actions required to achieve $\mathcal{G}$. We use the LAMA[36] alias of the Fast Downward system [37] to compute the plan. To reduce costs, we opt for *satisficing* instead of *optimal* configurations of the planner. To find the *satisficing* plan π^* that minimizes cost, we define the objective function as:

$$\pi^* = \arg\min_\pi C(\pi), \quad C(\pi) = \left(\sum_{m=0}^{M} c_m^\mathcal{R} + \sum_{n=0}^{N} c_n^\mathcal{H} \right) \quad (1)$$

where $c_m^\mathcal{R}$ is the cost of the agent's action $a_m^\mathcal{R}$ in plan π, and $c_n^\mathcal{H}$ is the cost of the human's action $a_n^\mathcal{H}$ in plan π. The cost of each action corresponds to the time taken to execute it. The optimal plan π^* minimizes $C(\pi)$ for both actors.

Human-robot collaboration. The computed plan $\pi^* = \langle a_1$, a_2, ..., $a_k \rangle$ represents a collaboration between the human and the agent. For example, the goal states in Figure 2 involve *preparing and serving breakfast of boiled eggs*, *preparing office clothes*, and *charging the cellphone*. To prepare breakfast, while the agent fetches eggs from the fridge nearby to the stove, the human is expected to bring a metal pot to the stove. Doing so will result in the task being completed in the least amount of time; Figure 1(a) is an illustration of this "collaboration" setting. There is no explicit communication between the human and the agent. While the agent will execute the relevant actions in the plan, there is no guarantee that the human will execute the assigned actions. Figure 1(b) is an illustration of the "no-collaboration" setting in which the

```
(:action agent_boils
 :parameters (?o - toboil)
 :precondition(and
              (item_in ?o metal_pot stove kitchen)
              (agent_near stove kitchen)
              (agent_switched_on burner stove kitchen)
              (not(boiled ?o)))
 :effect(boiled ?o))
```

Fig. 4: Action for *boiling* an item ?o*. Precondition: item must be in the metal pot.

agent and the human are assigned specific actions that do not minimize the overall cost. We experimentally compare the collaboration and no-collaboration settings in Section IV-B.

Adaptation to unexpected situations. DaTAPlan includes an approach that enables the agent to adapt if the human's action choices, action outcomes or preferences deviate from the expectations. Recall that the plan π_i^* will succeed only if both the agent and the human execute the actions assigned to them. The agent will meet this requirement, but we cannot guarantee that the human will do so. In addition, the human's task-level preferences may change and make the anticipated tasks irrelevant. For example, consider an agent executing actions to achieve the anticipated tasks of preparing and serving breakfast, and preparing the human's office clothes. This agent may find that the human was unable to bring the metal pot to the stove (to cook breakfast), requiring the agent to generate a new plan, or may be told that the human is no longer going to the office, requiring the LLM to provide new predictions of anticipated tasks.

In this paper, we illustrate our adaptation approach in the context of a specific kind of unexpected outcome: when the human does not execute a planned *pick up* action or the execution of this action does not result in the desired object being picked up. For instance, the plan shown in Figure 2 involves boiling eggs in a metal pot. Here, the agent expects a human to bring the metal pot to the stove (see Figure 4) while it fetches the eggs from the fridge to the stove. If the human does not pick up the metal pot, the subsequent steps of the plan will not achieve the part of the goal related to cooking breakfast. Although the agent and the human do not communicate with each other, the agent believes that it has a good model of the human's domain description, and the system provides full observability. The agent thus adapts by replanning from the current state. We experimentally compare performance with and without this adaptation strategy in Section IV-B.

We also illustrate the ability to adapt to unforeseen changes in the human's preferences. In this case, the agent directly prompts the LLM for a new sequence of anticipated tasks before replanning to achieve the revised goal.

Motion Planning. The low-level actions from the Task Planning are interpreted and are provided to the actors through CoppeliaSim Remote API. For actions that entail movement between locations, the nearest free space goal position is given to the OMPL BiTRRT Planner [38] with 10000 maximum search point nodes and 0.1 seconds for search duration for each simulation pass as search parameters. For actions such as *picking* and *cooking*, the object's goal position is used to control the movement of the arm using the Inverse Kinematics method provided by the CoppeliaSim IK Plugin.

*Due to space constraints, not all preconditions and effects are shown.

IV. Experimental Sections and Results

We experimentally evaluated four hypotheses related to the performance of DaTAPlan:

H1: LLMs can accurately anticipate future tasks based on a small number of contextual examples.

H2: Combining task anticipation and action planning substantially improves efficiency of planning and execution in human-robot collaboration scenarios compared with using just the classical planner.

H3: Human-robot collaboration results in more efficient goal attainment compared with no active collaboration.

H4: Agent is able to automatically adapt to unexpected changes in action outcomes and preferences of humans.

The Fast-Downward system [3] provides different configurations: *lama*, *seq-sat-fdss-2018*, and *seq-sat-fd-autotune-1*. We experimentally determined that *lama* provides the best performance and used it for all experiments reported here.

A. Experimental Setup

We begin by describing the experimental set up process.

LLM Prompting. We evaluated H1 quantitatively using different LLMs: Gemini Pro [39], Claude 3 [40] and GPT-4 [41]. We also extended the diversity of household tasks introduced in our prior work [1], to obtain 16 global tasks. To explore specific user task patterns, we created two households, each with five different scenarios (e.g., tasks related to different time of day), in which the sequence of task execution is different. *household-1* is characterized by orderly task execution, whereas in *household-2* immediate needs are prioritized with tasks performed as they come. We collected and used actual human expectations as the *ground truth*. Specifically, for each scenario, human expectations were recorded from 11 humans, with each of them asked to anticipate four tasks at a time. The humans were provided the same information as the LLMs. There is variability in the responses of different humans to the same scenario, and it is often hard to identify the "correct" response. Hence, we obtained the ground truth for each scenario by selecting four tasks with the highest frequency from all human responses for that scenario. Since computing the frequency of tasks may lead to the loss of information about the sequencing of tasks, we (instead) measured overlap. Let g_i be the set of (most frequent) tasks in the human responses (ground truth) for the i^{th} scenario, and l_i be the sets of tasks predicted by the LLM.; $g_i \cap l_i$ denoted the set of tasks that appear in both g_i and l_i. We then used two measures to evaluate **H1**:

- *Mean Overlap:* the average overlap between the ground truth and the LLM responses.

$$\text{Mean Overlap} = \frac{\sum_{i=1}^{n} |g_i \cap l_i|}{n} \tag{2}$$

where $|\cdot|$ denotes the cardinality of a set, and n is the total number of response pairs.

- $\geq 50\%$ *and* $\geq 75\%$ *Overlap:* the proportion of LLM outputs that have an overlap of at least 50% or 75% with the ground truth. This is calculated as:

$$\geq \text{Overlap} = \frac{\sum_{i=1}^{n} I(|g_i \cap l_i| \geq k)}{n} \tag{3}$$

where $k \in \{2, 3\}$, and $I(\cdot)$ is the indicator function defined as: $I(x) = 1$ if x is true, and 0 if x is false. Note that 50% overlap implies two tasks in common between LLM response and the ground truth, and 75% overlap implies three common tasks.

In these measures, n is the number of response pairs, i.e., number of scenarios multiplied by the number of LLM prompts. We collected data for five scenarios from 11 humans, and prompted each LLM 25 times: $n = 5 \times 25 = 125$.

Planning. The household environment in which the agent and human operated had four rooms: *Bathroom*, *Kitchen*, *Storeroom*, and *Livingroom*. As stated in Section III-B, the mapping of the properties of this environment (and the agent, human) in PDDL had 72 predicates and 88 actions, with 39 *human-specific* actions, 39 *robot-specific* actions, and 10 actions common to both. There were 17 *types* of objects.

The default domain description had the agent and the human collaborating to achieve the desired goals. To mimic lack of collaboration between the human and the agent, plans were generated separately for the human and the agent with the corresponding domain description only containing the human-specific and the agent-specific actions (respectively).

Plans were computed to minimize the total cost. We determined action costs based on four factors: (i) *distance to the target location*, encoding the principle that the actor closest to a location should move to it; (ii) *object type*, encoding the preference that humans handle fragile objects if possible because it is harder for the agent to safely grasp and move such objects; (iii) *task completion time*, encoding prior knowledge that humans are more efficient with some complex tasks such as cooking while the agent is more efficient with some tasks such as cleaning; and (iv) *action priorities*, encoding the principle that an agent should handle the more repetitive tasks such as fetching objects. Note that the planner often had to trade-off between these factors when computing a plan, e.g., the agent is closer than the human to a fragile object that needs to be moved to a different room.

Next, we used two measures to evaluate the performance of the planning framework:

- *Execution cost:* the execution cost of plan π is defined as the sum of the costs of all actions executed by the agent ($\mathcal{R}$) and the human ($\mathcal{H}$) while following π; this is computed as shown in Equation 1.
- *Plan length:* the length of a plan is the total number of actions in the plan, computed as the sum of the number of actions executed by the agent and the human.

Adaptation. As with collaboration, the standard operation of DaTAPlan had the agent adapting to unexpected changes in the human's action outcomes and preferences as described in Section III-B. Specifically, any unexpected outcome of a *pick up* action executed by a human resulted in a new plan being generated from the current state. The agent then executed the actions (in the plan) allocated to it and expected the human to do the same. The planning and execution costs of this new plan were added to the plan(s) computed and executed so far. In a similar manner, a change in the human's high-level preference resulted in a new sequence of anticipated tasks being generated by the LLM, followed by a new plan.

We simulated the lack of such an adaptive approach by having the agent and the human continue executing the actions allocated to them (to the extent possible) even when the human's action did not have the expected outcome. The agent became aware of the unexpected outcome only when the current plan resulted in failure, at which point a new plan could be generated. In this case, the time made available to the agent and the human to compute and execute plans matched the time taken to achieve the goal when the adaptation approach was used. The performance measure was the fraction of the goal achieved in the time available.

B. Experimental Results

Evaluating H1. As stated in Section IV-A, we evaluated the ability of three different LLMs to anticipate upcoming tasks in two different households, compared these predictions with those of a set of humans, and used Equations 2-3 as the performance measures. We also considered the two prompting mechanisms described in Section III-A. The results are documented in Table I. Recall that *household-1* was characterized by an orderly task execution whereas *household-2* prioritized immediate needs. The results in Table I indicate that there was a greater degree of overlap between the predictions of the LLMs and those of humans in household-1. In the case of household-2, there was greater variability in the predictions provided by the LLMs and in those provided by the human subjects; as a result, the degree of overlap between humans and LLMs was lower.

Among the prompting methods, better performance, i.e., a higher degree of overlap between the predictions of the LLMs and the humans, was obtained with the chain of thought prompting compared with the few shot prompting method. In fact, the mean overlap value was as high as 86% for the *Claude-3* LLM, and it correctly anticipated at least three out of four tasks with the chain of thought prompting (compared with the ground truth). In addition, in the absence of this prompting method, which includes more contextual information, the LLMs often went into a loop of hallucination. Overall, these results support hypothesis **H1**.

Evaluating H2. We compared the performance of DaTAPlan (i.e., our framework) with that of a framework that used a classical planner to compute the sequence of actions for one high-level task at a time. As with **H1**, we considered the five scenarios in each of two households. In each scenario, the number of anticipated tasks was varied from 0-4, with 0 corresponding to no anticipation. The planner was provided

	LLM Models →	Claude		GPT-4		Gemini	
		few-shot	CoT	few-shot	CoT	few-shot	CoT
Household 1	Overlap	0.72	**0.86**	0.75	0.82	0.59	0.77
	$\geq 50\%$ overlap	0.99	**1**	**1**	**1**	0.92	**1**
	$\geq 75\%$ overlap	0.77	**1**	0.8	0.96	0.46	0.78
Household 2	Overlap	0.70	0.71	0.68	**0.72**	0.57	0.6
	$\geq 50\%$ overlap	0.94	0.98	0.96	**1**	0.9	0.93
	$\geq 75\%$ overlap	0.69	**0.79**	0.68	0.76	0.32	0.42

TABLE I: Evaluating LLM-based task anticipation for two separate households based on few-shot prompting and chain-of-thought reasoning. Results support **H1**.

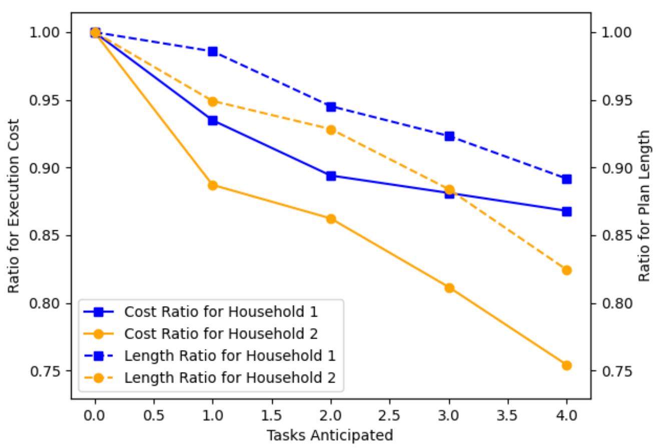


Fig. 5: Evaluating **H2**. Values of *execution cost* and *plan length* with different levels of anticipation computed as a ratio over values computed for no anticipation; paired trials conducted for different scenarios in two different households. The combination of task anticipation and action planning improved performance.

an initial search time limit that was then increased in fixed increments. We conducted paired trials, i.e., for any particular scenario in a specific household, we would pick a particular initial state and run trials with and without anticipation. Recall that the performance measures for this experiment were *execution cost* and *plan length*. Since the values of these measures can change drastically depending on the initial conditions, we computed the ratio of the value (of a performance measure) for a given level of anticipation (ranges from 1-4 tasks being anticipated) with the value for no anticipation. The results are summarized in Figure 5; each point in the plots is the average of the ratios computed for five scenarios in a particular household.

The results in Figure 5 clearly indicated a decrease in the planning time and execution cost as the number of anticipated tasks increased. Overall, there was a drop in total execution cost of $\approx 12.5\%$ and $\approx 25\%$ in household-1 and household-2 respectively as the number of anticipated tasks increases from one to four. Similarly, we observed a decrease of approximately 10% and 17.5% in plan length in Household 1 and 2 respectively. These plots support hypothesis **H2** and demonstrate the advantages of integrating LLM-based high-level task anticipation with planning (a sequence of low-level actions) for achieving the tasks.

Evaluating H3. We conducted paired trials with and without collaboration, and measured the total time taken by both actors (agent, human) to achieve the corresponding goal states.

	Overlap
Gemini	0.59
Claude	0.98
GPT-4	0.91

TABLE III: Evaluating **H4**. Examined re-prompting of different LLMs during task execution in response to an unexpected change in human preference that makes the current set of anticipated tasks irrelevant. We measured the overlap between new set of anticipated tasks predicted by LLMs and the new predictions provided by human subjects; results support **H4**.

Specifically, we created 16 different initial conditions by considering different combinations of initial state of the robot and the human, e.g., agent initially in the *Kitchen* with human in the *Storeroom*. We also considered two settings: (i) *HFAS (Human-Fast-Agent-Slow)*, in which the human was faster than the robot and had a lower movement cost; and (ii) *AFHS (Agent-Fast-Human-Slow)*, in which the agent was faster and had a lower movement cost. Then, for each initial condition and each setting, we ran 50 trials, each with two tasks in the goal state. In each trial, we computed ζ as the ratio of the total execution time with collaboration over the value of the measure without collaboration; $\zeta < 1$ denotes effective collaboration. We then computed the average of these ratios for a particular initial state, and report the average of these ratios in Table II.

Results indicate that we achieved $\zeta < 1$ for all initial conditions and all settings, clearly demonstrating the benefits of collaboration between the agent and the human in terms of the reduction in execution time; the maximum reduction in time (or equivalently the cost) was 25.3%, which was obtained in the AFHS setting. Overall, these results clearly support hypothesis **H3**.

Evaluating H4. We first explored the ability of the agent to adapt to unexpected changes in human preferences, i.e., in situations that required the agent to generate a new set of anticipated tasks (from the LLM) before computing and executing the corresponding plan. In these experiments, the interrupt (due to change in human preference) occurs as the agent and the human are executing the actions. We conducted experiments similar to those for **H1**, measuring the overlap between the new set of anticipated tasks predicted by the LLMs and the tasks predicted by human subjects; we conducted these experiments over 25 repetitions (each) of five different scenarios (with chain of thought promoting). The corresponding results, summarized in Table III, indicated support for hypothesis **H4**.

Next, we explored the ability to adapt to unexpected changes in the outcomes of human actions; as stated in Section III-B, we randomly introduced errors in the outcome of the *pick up* actions executed by the human. We considered combinations of 16 different goals, four each with 1-4 high-level tasks, and 1-3 instances of errors (in human pickup actions). The goal states contained tasks ranging from simple ones such as picking up clothes from the closet and placing them on the ironing board, to complex tasks such as *cleaning* and *slicing* multiple fruits before placing them in a *bowl* to *prepare salad. Our adaptation approach results in the goals being*

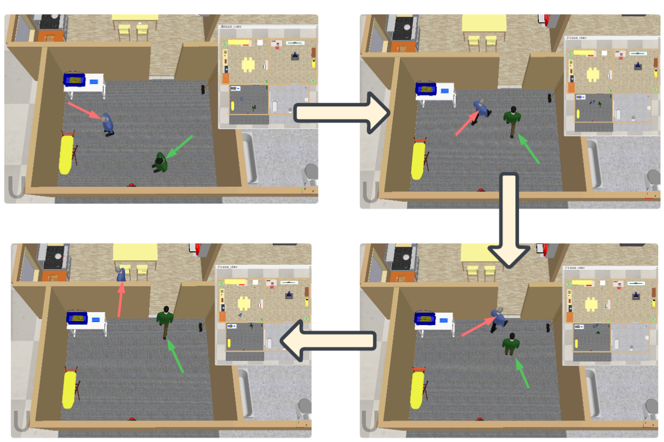
	ROBOT								
	Init →	Livingroom		Storeroom		Bathroom		Kitchen	
	Init ↓	ζ_{HFAS}	ζ_{AFHS}	ζ_{HFAS}	ζ_{AFHS}	ζ_{HFAS}	ζ_{AFHS}	ζ_{HFAS}	ζ_{AFHS}
HUMAN	Livingroom	0.913	0.778	0.883	**0.747**	0.913	0.788	0.895	0.757
	Storeroom	0.969	0.873	0.959	0.863	0.98	0.843	0.929	0.853
	Bathroom	0.968	0.838	0.926	0.794	0.949	0.828	0.911	0.833
	Kitchen	0.943	0.846	0.927	0.81	0.925	0.826	0.924	0.856

TABLE II: Evaluating **H3**. 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**.

No. of Tasks	No. of erroneous outcomes			Success Rate %
	1	2	3	
1	0 ± 0	0 ± 0	0 ± 0	0.0 ± 0.0
2	0.5 ± 0	0.25 ± 0.25	0 ± 0	25.0 ± 8.33
3	0.49 ± 0.16	0.33 ± 0.23	0.25 ± 0.14	36.11 ± 9.21
4	0.62 ± 0.12	0.56 ± 0.11	0.43 ± 0.11	54.17 ± 9.32

TABLE IV: Evaluating **H4**. Considered combinations of goal states with 1-4 high-level tasks with 1-3 instances of erroneous human (pick up) actions. Computed mean and standard deviation of the fraction of tasks completed without adaptation in the same amount of time taken by the adaptation approach (in DaTAPlan) to achieve the goal. Results support **H4**.

(a) (b)



(d) (c)

Fig. 6: Collision avoidance between human (orange arrow) and the agent (green arrow) in CoppeliaSim: (a) initial position of human and agent; (b) both move in a direction that could result in a collision in the near future; (c) agent stops and waits for the human to pass; (d) agent continues once the collision situation is addressed.

accomplished in all trials despite the errors in human actions by replanning from a revised initial state, but there is an increase in execution (and planning) time. We thus conducted paired trials and computed (in each trial) the fraction of the high-level tasks (in the goal) achieved in absence of our adaptation approach in the same amount of time taken by our adaptation approach. The corresponding results are summarized in Table IV, with the last value in each row representing the average performance over different levels of errors for a goals with a specific number of high-level tasks. There was a reduction in performance in the absence of our adaptation approach, and this reduction was (understandably) more pronounced with the increase in the number of instances of errors in the outcomes of the human actions. These results support **H4**.

Finally, we evaluated the ability of our framework to detect and address instances of possible collisions between the agent and the human. Any such potential collisions were successfully avoided in all trials and we show a qualitative example in Figure 6. Our implementation successfully tracks the trajectory of both actors. When a potential collision is detected, the agent stops and allows the human to pass first, and proceeds with its trajectory once it is safe to do so again.

V. Conclusions and Future Works

This paper described DaTAPlan, a framework that combines data-driven task anticipation with knowledge-driven planning for reliable and efficient human-agent collaboration. This builds on our recent work that combined LLM-based task anticipation with classical planning for a single agent in a domain with no other actors [1]. Here, the LLM-based anticipation is extended to accommodate different user patterns. Also, the anticipated high-level tasks form the goal for a classical planning system that generates a sequence of finer-granularity actions, with both the agent and the human executing actions to collaboratively achieve each task in the goal. In addition, an approach enables adaptation to unexpected changes in the action outcomes and preferences of the human. We experimentally evaluated these capabilities and demonstrated substantial improvement in performance compared with a framework without collaboration or without the adaptation approach. The paper opens up multiple directions of further research. First, we currently do not model or support any active communication between the agent and the human; this constraint can be relaxed to consider communication as an action. Second, we consider the domain descriptions (including action costs) to be complete and accurate; future work can explore reasoning with incomplete descriptions and the incremental revision of existing descriptions. Third, action execution has limited uncertainty in the current implementation and there is full observability; these assumptions can be relaxed in the future. Overall, the long-term objective is to enable physical robots to provide reliable and efficient assistance to humans in complex domains.

References

[1] R. Arora, S. Singh, K. Swaminathan, A. Datta, S. Banerjee, B. Bhowmick, K. M. Jatavallabhula, M. Sridharan, and M. Krishna, “Anticipate & act : Integrating llms and classical planning for efficient task execution in household environments,” in *International Conference on Robotics and Automation*, 2024.

[2] M. Ghallab, C. Knoblock, D. Wilkins, A. Barrett, D. Christianson, M. Friedman, C. Kwok, K. Golden,

S. Penberthy, D. Smith, Y. Sun, and D. Weld, "Pddl - the planning domain definition language," 08 1998.

[3] M. Helmert, "The fast downward planning system," *Journal of Artificial Intelligence Research*, vol. 26, pp. 191–246, jul 2006.

[4] A. Smith and B. Jones, "Advancements in human-robot interaction for household tasks: A review," *International Journal of Robotics Research*, vol. 41, no. 3, pp. 345–367, 2022.

[5] C. Lee and D. Kim, "Collaborative robots: Challenges and opportunities in human-robot interaction," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 210–228, 2023.

[6] M. Kraus, N. Wagner, W. Minker, and et al., "Kurt: A household assistance robot capable of proactive dialogue," in *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '22. IEEE Press, 2022, p. 855–859.

[7] R. Garcia and S. Martinez, "Learning from demonstration for adaptive control in human-robot collaboration," *Journal of Artificial Intelligence Research*, vol. 56, pp. 78–95, 2022.

[8] J. Park and H. Kim, "Collaborative task planning for human-robot interaction: Recent advances and future directions," *IEEE Robotics and Automation Magazine*, vol. 31, no. 2, pp. 77–94, 2024.

[9] Y. Wang and J. Li, "Context-aware task planning for robot assistance in daily living," *Robotics and Autonomous Systems*, vol. 95, pp. 210–225, 2022.

[10] S. Kim and H. Lee, "Human-robot task planning using probabilistic graphical models in smart homes," *ACM Transactions on Intelligent Systems and Technology*, vol. 12, no. 3, pp. 45–62, 2021.

[11] Q. Chen and Z. Liu, "Collaborative task planning for household service robots: A review of approaches and challenges," *International Journal of Social Robotics*, vol. 16, no. 2, pp. 189–207, 2024.

[12] X. Li and H. Wang, "Learning-based task planning for human-robot collaboration in smart homes," *IEEE Transactions on Robotics*, vol. 41, no. 2, pp. 321–338, 2022.

[13] L. Yang and Q. Zhang, "Adaptive task planning and execution for human-robot collaboration in dynamic environments," *Robotics and Autonomous Systems*, vol. 85, pp. 150–165, 2021.

[14] W. Liu and J. Huang, "Reinforcement learning for task planning and execution in human-robot interaction," *IEEE Transactions on Cybernetics*, vol. 54, no. 1, pp. 78–94, 2024.

[15] Z. Cai, Z. Feng, L. Zhou, C. Ai, H. Shao, and X. Yang, "A framework and algorithm for human-robot collaboration based on multimodal reinforcement learning," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–13, September 2022.

[16] J. Felip, D. Gonzalez-Aguirre, and L. Nachman, "Intuitive & efficient human-robot collaboration via real-time approximate bayesian inference," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 3093–3099.

[17] E. Brown and F. White, "Addressing safety and trust in human-robot collaboration for household tasks," *Robotics and Autonomous Systems*, vol. 87, pp. 123–139, 2021.

[18] K. Valmeekam, A. Olmo, S. Sreedharan, and S. Kambhampati, "Large language models still can't plan (a benchmark for LLMs on planning and reasoning about change)," in *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

[19] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, and P. Stone, "Llm+p: Empowering large language models with optimal planning proficiency," 2023.

[20] Y. Xie, C. Yu, T. Zhu, J. Bai, Z. Gong, and H. Soh, "Translating natural language to planning goals with large-language models," 2023.

[21] F. Joublin, A. Ceravola, P. Smirnov, F. Ocker, J. Deigmoeller, A. Belardinelli, C. Wang, S. Hasler, D. Tanneberg, and M. Gienger, "Copal: Corrective planning of robot actions with large language models," 2023.

[22] S. Izquierdo-Badiola, G. Canal, C. Rizzo, and G. Alenyà, *PlanCollabNL: Leveraging Large Language Models for Adaptive Plan Generation in Human-Robot Collaboration*. IEEE, Jan. 2024.

[23] K. Valmeekam, M. Marquez, A. Olmo, S. Sreedharan, and S. Kambhampati, "Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change," 2023.

[24] E. Hirsch, G. Uziel, and A. Anaby-Tavor, "What's the plan? evaluating and developing planning-aware techniques for llms," 2024.

[25] T. Silver, S. Dan, K. Srinivas, J. B. Tenenbaum, L. P. Kaelbling, and M. Katz, "Generalized planning in pddl domains with pretrained large language models," 2023.

[26] T. Birr, C. Pohl, A. Younes, and T. Asfour, "Autogpt+p: Affordance-based task planning with large language models," 2024.

[27] T. Silver, V. Hariprasad, R. S. Shuttleworth, N. Kumar, T. Lozano-Pérez, and L. P. Kaelbling, "PDDL planning with pretrained large language models," in *NeurIPS Workshop on Foundation Models for Decision Making*, 2022.

[28] L. Guan, K. Valmeekam, S. Sreedharan, and S. Kambhampati, "Leveraging pre-trained large language models to construct and utilize world models for model-based task planning," in *Advances in Neural Information Processing Systems*, A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 79 081–79 094.

[29] Z. Zhao, W. S. Lee, and D. Hsu, "Large language models as commonsense knowledge for large-scale task planning," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[30] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: Theory and Practice*. San Francisco, USA:

Morgan Kaufmann, 2004.
[31] S. Izquierdo-Badiola, G. Canal, C. Rizzo, and G. Alenyà, "Improved task planning through failure anticipation in human-robot collaboration," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 7875–7880.
[32] S. Yang, X. Mao, Q. Wang, and Y. Huang, "A hybrid planning approach for accompanying information-gathering in plan execution monitoring," *Journal of Intelligent & Robotic Systems*, vol. 103, 2021.
[33] E. Rohmer, S. P. N. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1321–1326.
[34] T. Brown, B. Mann, N. Ryder, M. Subbiah, and et al, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.
[35] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 2023.
[36] S. Richter and M. Westphal, "The lama planner: guiding cost-based anytime planning with landmarks," *J. Artif. Int. Res.*, vol. 39, no. 1, pp. 127—-177, sep 2010.
[37] M. Helmert, "The fast downward planning system," *Journal of Artificial Intelligence Research*, vol. 26, pp. 191—-246, Jul. 2006.
[38] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, December 2012.
[39] G. Team, R. Anil, S. Borgeaud, and et al., "Gemini: A family of highly capable multimodal models," 2023.
[40] Anthropic, "Claude 3: A family of large, capable, and steerable models," https://www.anthropic.com/news/claude-3-family, Mar. 2023, accessed: 16 March, 2024.
[41] OpenAI, "Gpt-4 technical report," *ArXiv*, vol. abs/2303.08774, 2023.