

Courteous Anticipation: Improving Long-Lived Task Planning in Persistent Shared Environments

Md Ridwan Hossain Talukder, Roshan Dhakal, Elizabeth Phillips and Gregory J. Stein

Abstract—We consider a task planning scenario in which robots sharing a persistent environment are assigned tasks one at a time from a held-out sequence. Standard task planners, lacking foresight of future tasks and inconsiderate of others’ constraints, solve each task in isolation, leaving terminal states that increase future cost for all—side effects that compound over lengthy task sequences. To reduce cost over the sequence, a robot must anticipate how its actions now may impact performance on future tasks for all robots sharing the environment. Therefore, we present *courteous anticipatory planning*, wherein a model-based planner proposes candidate plans and selects the one that jointly minimizes immediate cost and aggregated expected future cost across all robots, estimated via independent per-robot learned estimators. This factored formulation avoids combinatorial joint rollouts and supports modular deployment: adding a robot requires only training its own estimator. We evaluate in two persistent PDDL domains, a home environment with robots that have similar capabilities but different responsibilities, and a restaurant environment where robots’ distinct capabilities create states that other robots lack the capability to resolve. During lengthy task sequences, our planner reduces total cost by 10.43% versus myopic and 4.03% versus selfish anticipatory planning in a two-robot home environment and by 17.41% and 13.24%, respectively, in a three-robot restaurant.

I. INTRODUCTION

We consider a persistent, home-scale environment shared by multiple robots, where each robot is given one task at a time. Traditional task planners [1]–[10] typically focus solely on achieving immediate goals and as quickly as possible. Lacking foresight of the impact of their decisions on *future* tasks, they do not account for their impact on others that share the environment. For long-lived deployments, where environments persist beyond a single task, this shortsighted and selfish behavior of each robot often creates negative side effects on others, making tasks more costly over time.

Consider a shared kitchen with three robots: a heat-shielded cook that alone can safely access the stove, a server, and a scrubber-equipped short cleaner that alone can wash cookware. Suppose the cook is assigned the *MakePasta* task which first requires moving a dirty bowl to clear the stove. A conventional plan for the cook completes the task, yet lacks foresight about future tasks the robots may be assigned. This *myopic* plan (Fig. 1, blue) clears the stove by moving the dirty bowl to the high counter, inadvertently blocking the server that is later assigned the *ServePasta*

In this shared home, asked to **MakePasta**, our cook *anticipates* the impact of its current actions on others’ future tasks and *courteously* completes it, reducing the overall planning cost.

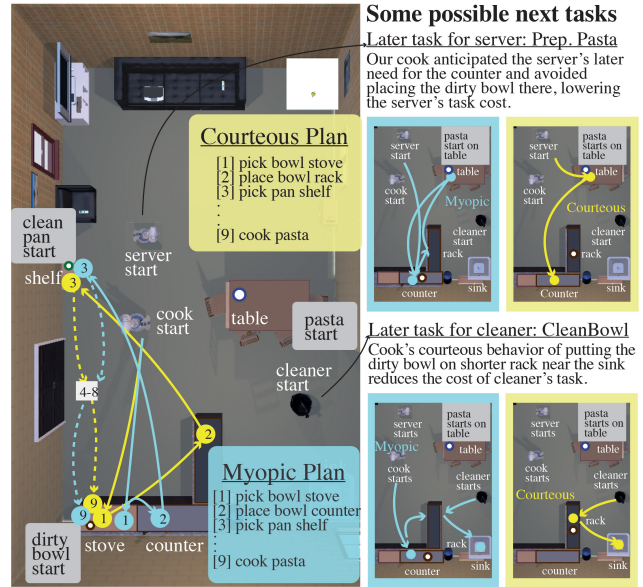


Fig. 1. **Courteous Anticipation:** Given the task *MakePasta* to the cook robot a **myopic** plan leaves the dirty bowl on *counter* blocking the server and forcing short cleaner to ask for help to complete their later task. **Our Approach** anticipates the impact of its action on all robots and *courteously* places the dirty bowl near *sink*, thus reducing the overall planning cost.

task and forcing the short cleaner to seek *help* from a tall robot to retrieve the bowl—a costly detour—before it can complete its *CleanBowl* task. Instead, the robot has an opportunity to be *courteous* and reduce downstream friction; absence such courtesy myopic plan increases the overall cost of the sequence of tasks eventually assigned to each robot.

To exhibit courteous behavior in a shared environment, a robot must reason beyond its immediate task and anticipate how its actions *now* will affect others as they are eventually assigned tasks of their own. In the earlier example, a courteous plan for the cook (Fig. 1, yellow) would place the dirty bowl near the sink (on a lower rack), freeing the counter for the server and reducing the cleaner’s cost of washing the bowl later. We call this behavior *courteous anticipation*: the ability of a robot to complete its own task while being considerate of the future tasks and capabilities of others and acting accordingly. However, achieving courteous anticipation is challenging: a robot must act without knowing what tasks it or others will be assigned next and how it impacts others physical constraints. With the broad responsibilities of each robot, reasoning directly over all such tasks that may be assigned quickly becomes combinatorially expensive.

The emerging field of anticipatory planning [11]–[13] proposes an objective function in which a robot considers both its immediate objective and the impact of its actions on tasks it may be later assigned. However, in shared spaces with several robots, self-interested anticipation is insufficient. A robot that considers only for its own future performance may still leave the space at a terminal state that is costly for robots sharing the space: behavior we term as *selfish* anticipation. To improve *collective* behavior in shared spaces, especially when each robot has different responsibilities and distinct capabilities, a robot must consider the impact of its actions on others. It is a key insight of this work that standard task planning in shared spaces, absent this consideration, routinely creates negative side effects and increases the cost of later tasks of others. Our *courteous anticipation* aims to mitigate these side effects by making the acting robot account for others, reducing the cost of sequential task planning.

We introduce *courteous anticipatory planning*: a planning objective that accounts for the distinct capabilities and responsibilities of all robots by jointly minimizing the immediate cost of completing the current task and the expected future cost it induces for all robots sharing the space. Directly evaluating this expected future cost across all robots requires brute-force joint rollouts over the combinatorial space of possible task sequences, which becomes computationally intractable as robots and responsibilities grow. Instead, we decompose this expected future cost into independent per-robot estimates, each trained to capture how the environment state affect that robot’s ability to fulfill its responsibilities under its structural constraints. At planning time, we aggregate these estimates to select the most courteous plan, avoiding joint rollouts and requiring no retraining of existing estimators when robots are added or removed. The contributions of this paper are as follows.

- We identify an important obstacle to effective sequential task planning in persistent shared environments. The pervasive absence of courtesy from standard task planning in these domains introduces negative side effects that increase overall planning cost, an effect amplified when robots differ in capabilities and responsibilities.
- We introduce *courteous anticipation*: a planning objective that jointly minimizes immediate plan cost and aggregated expected future task costs across all robots. Our learning-augmented, model-based planner achieves courteous anticipation via a factored formulation of independent per-robot estimators, avoiding combinatorial joint rollouts and supporting modular deployment without retraining.
- We introduce a shared-environment evaluation setting of increasing complexity: a home-setting that establishes improvements from robots’ distinct responsibilities alone, and a restaurant-setting that demonstrates that distinct capabilities amplify these gains—both evaluated against myopic and selfish anticipatory baselines.

Evaluated over lengthy task sequences, our planner reduces total cost by 10.43% versus myopic and 4.03% versus self-

ish anticipatory planning in a two-robot home environment and by 17.41% and 13.24%, respectively, in a three-robot restaurant.

II. RELATED WORK

A. Task Planning and Learning-Augmented Planning

Conventional task planning approaches [1]–[5] seek to solve each task at minimum immediate cost, without considering how the resulting terminal state affects future tasks. Some approaches augment planning with learned heuristics or value functions [6]–[10] to accelerate the search, but not to reason about the downstream consequences of these solutions. In persistent shared environments, however, this myopia is costly: plans may leave a terminal state that increases task cost for robots with different structural constraints and responsibilities. Our approach instead selects among plans by reasoning about downstream consequences for all robots, using learned per-robot estimators that capture each robot’s capabilities and responsibilities.

B. Anticipating Future Tasks in Persistent Environments

Recent work has presented approaches to anticipate and mitigate unforeseen side effects on future unseen tasks within a persistent physical space [11]–[13], estimating future costs using learned models to avoid expensive rollouts. Other techniques have also been developed to proactively anticipate the usage of objects to assist others [14], [15], to align human and robot expectations by designing environments [16], or to minimize divergence from a goal state [17]. Online task planning scenarios have also been considered for anticipatory planning [18], focusing on the temporal arrival of tasks for robots rather than sequential arrangement. Nevertheless, all these techniques are single-robot techniques; they only consider how a robot’s actions influence its future tasks but not how these actions limit or enable other robots with varying structural constraints and goals in a shared space. Klassen et al. [19] address negative side effects in symbolic planning by computing side effect-minimizing plans via compilations to cost-optimizing STRIPS but in a single-task setting, without modeling future task sequences or the distinct capabilities of robots sharing the space.

III. PROBLEM STATEMENT

A. Preliminaries: Task Planning with PDDL

In this work, tasks and actions are represented via PDDL (The Planning Domain Definition Language) [20]. Formally, a task planning problem is defined as a tuple $\langle \mathcal{O}, \mathcal{P}, s_0, \mathcal{A}, \tau \rangle$, where $\mathcal{O}$ represents all objects (e.g., robots, containers, and interactable objects), $\mathcal{P}$ denotes the set of predicates expressing relationships among these entities, s_0 is the initial state, and $\mathcal{A}$ is a set of parameterized actions. The *task* τ is defined as a union of predicates that specifies a subset of the state space in which the task is considered complete: the goal space $G_\tau \subseteq \mathcal{S}$.

B. Sequential Task Planning in a Shared Environment

We consider a shared environment with a set of robots R . Each robot $r \in R$ has its own *responsibilities*: a set of tasks τ it may be assigned and their respective probability $P_r(\tau)$. We are given a sequence of $N = n|R|$ tasks τ containing n sampled tasks for each robot r , which arrives one at a time. At each step i in the sequence, a robot r is randomly chosen and assigned a task $\tau_i \sim P_r(\tau)$ at random, drawn from its responsibilities.

The environment persists between tasks: the terminal state after completing one task becomes the starting state of the next. In notation, the state s_{g_i} which is the terminal state of completing τ_i is also the initial state of completing the task τ_{i+1} . It is the aim of each robot to plan in such a way that the cost over completing all tasks in the sequence is minimized. If the task sequence were known in advance, it would be possible to compute the minimum total-cost plan via the following equation.

$$s_{g_1}^*, \dots, s_{g_N}^* = \underset{s_{g_i} \in G_{\tau_i} \forall i \in 1 \dots n}{\operatorname{argmin}} [V_{s_{g_1}}^*(s_0) + V_{s_{g_2}}^*(s_{g_1}) + \dots + V_{s_{g_n}}^*(s_{g_{n-1}})]. \quad (1)$$

where $V_{s_{g_i}}^*(s_{g_{i-1}})$ is the cost of reaching the terminal state of task τ_i from $s_{g_{i-1}}$ considering the complete sequence of tasks ahead.

Since the robot will not, in general, know all its own tasks and those of others in advance, it instead seeks a plan to complete the immediate task that minimizes *expected total cost*, with future tasks drawn from each robot's responsibilities. Computing this expected total cost in general is computationally infeasible as the space of possible future task sequences grows combinatorially with the number of robots and breadth of their responsibilities which motivates our approximation approach described in Sec. V.

IV. COURTEOUS ANTICIPATION IN A PERSISTENT SHARED ENVIRONMENT

With the problem formulation of Sec. III in hand, we develop our courteous anticipatory planning objective, grounding it in single-robot anticipatory planning before presenting a formulation that reasons about the distinct capabilities and responsibilities of all robots sharing the environment.

Recent work [11], [12] proposes single-robot *anticipatory planning*: a planning objective that minimizes the joint cost of the current task and the expected cost associated with future tasks that robot may be assigned next. Given the current task τ_c with starting state s_0 and defining the terminal state of the plan π as $s'_g \equiv \text{TAIL}(\pi) \in G_{\tau_c}$ this objective is:

$$\begin{aligned} s_g^* &= \underset{s_g \in G_{\tau_c}}{\operatorname{argmin}} \left[V_{s_g}(s_0) + \sum_{\tau'} P(\tau') V_{\tau'}(s_g) \right] \\ &= \underset{s_g \in G_{\tau'}}{\operatorname{argmin}} \left[V_{s_g}(s_0) + V_{\text{A.P.}}(s_g) \right]. \end{aligned} \quad (2)$$

where $V_{s_g}(s_0)$ is the cost to reach goal state $s_g \in G_{\tau_c}$ from s_0 and $V_{\tau'}(s_g)$ is the cost of completing a future task τ' from

state s_g . $V_{\text{A.P.}}(s')$ is the *anticipatory cost*: shorthand for the expected cost to complete a follow-up task from state s' .

While effective for a single robot, this formulation remains *selfish* in shared environments. Absent any consideration of how robot's current actions impact *other* robots that share the environment, the robot may behave *selfishly*, creating downstream side-effects that increase total cost over task sequences. Consider the cook placing a dirty bowl on the counter while minimizing its own future cost yet blocking the server and increasing the cleaner's task cost (Fig. 1). So, minimizing Eq. (2) alone cannot resolve this, as it is inconsiderate to the capabilities and responsibilities of others. To capture the impact of planning decisions on all robots sharing the environment, we propose *courteous anticipation*.

$$s_g^* = \underset{s'_g \in G_{\tau}}{\operatorname{argmin}} \left[V_{s'_g}^*(s_0) + \sum_{r \in R} V_{\text{A.P.}}^r(s'_g) \right]. \quad (3)$$

Unlike Eq. (2), which models only the acting robot's future responsibilities, Eq. (3) aggregates distinct responsibility distributions and structural constraints across robots, altering the terminal-state ranking. Here, $V_{\text{A.P.}}^r(s'_g)$ denotes the anticipatory cost for robot r to complete a single subsequent task drawn from its responsibilities, starting from terminal state s'_g . The acting robot therefore evaluates candidate terminal states by considering both (i) the immediate cost of completing its current task and (ii) the expected downstream cost that its chosen state induces for each robot in the scene. Directly computing this expected future cost requires enumerating all possible future task assignments across all robots, one drawn from each robot's responsibilities, a combinatorial space that quickly becomes intractable as the number of robots and responsibilities grows.

Our approach addresses this intractability by factoring the sum across robots independently, estimating each robot's contribution separately rather than jointly. Specifically, $V_{\text{A.P.}}^r(s'_g)$ is estimated via a learned model trained specifically on robot r 's responsibilities, implicitly capturing how the terminal state s'_g interacts with that robot's capabilities, such as objects placed in locations inaccessible to robot r , or left in states it cannot change, will incur higher estimated costs, encoding the capability constraints. This decomposition remains tractable while capturing the primary source of cross-robot side effects: the physical configurations left by the acting robot.

V. COURTEOUS ANTICIPATORY PLANNING

In this section, we describe our courteous anticipatory planner Alg. 1 which proposes candidate plans that achieve the current task and, from the resulting set, selects the plan that minimizes the joint cost of the immediate plan and expected future cost aggregated from each robot, relying on a per-robot learned model to estimate the anticipatory cost for each. We detail the plan-sampling procedure in Sec. V-A (Alg. 1, Line 3) and the per-robot anticipatory cost estimator in Sec. V-B (Alg. 1, Line 5). Fig. 2 includes a schematic overview of our approach.

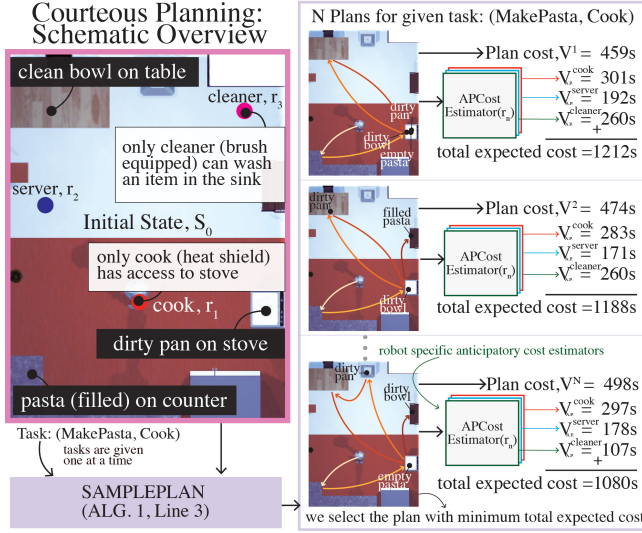


Fig. 2. **Courteous Planning:** Our courteous anticipatory planning generates a set of candidate plans and selects from them the plan that jointly minimizes the cost of immediate plan and the expected future cost aggregated for all robots.

Algorithm 1: Courteous Anticipatory Planning

Input: s_0, τ
Output: π^*

```

1  $\pi^* \leftarrow \emptyset, V_{\text{total}}^* \leftarrow \infty$ 
2 for  $i \in \{1, 2, \dots, N\}$  do
3    $\pi \leftarrow \text{SAMPLEPLAN}(s_0, \tau, r_{\text{act}})$ 
4    $s_g, V_{s_g} \leftarrow \text{TAIL}(\pi)$ 
5    $V_{\text{total}} \leftarrow V_{s_g} + \sum_{r \in R} \text{APCOSTESTIMATOR}_r(s_g)$ 
6   if  $V_{\text{total}} \leq V_{\text{total}}^*$  then
7      $V_{\text{total}}^* \leftarrow V_{\text{total}}$ 
8      $\pi^* \leftarrow \pi$ 
9 return  $\pi^*$ 

```

A. Generating Candidate Plans

In large environments there could be exponential number of plans that can satisfy the task, which might make the planning problem intractable quickly. We adopt the focused sampling procedure of Talukder et al. [12], which has been shown to generate sufficiently diverse satisficing plans in large-scale persistent environments while remaining computationally tractable. Rather than random sampling over the exponential space of satisficing plans, focused sampling selectively augments tasks with placement predicates and object state predicates—specifying exactly where moved objects should be placed and in what state they should be left—generating candidate plans that are structurally distinct in ways that matter for downstream cost while rejecting any plans that fail to accomplish the original task. This property is essential for courteous anticipation, where the acting robot must evaluate how different object placements and states affect the future costs of all robots sharing the environment. For example, a dirty pan left on the stove is inaccessible to the server, who cannot reach the stove, and unusable by the cook, who can reach the stove but cannot use a dirty pan, until the cleaner washes it.

For example, in a `ClearSink` task, the cleaner robot must move a dirty bowl from the sink but has discretion over its placement and final state (e.g.: clean or dirty); proxy tasks enumerate candidate placements and object states, generating structurally distinct plans whose downstream consequences differ meaningfully for robots sharing the environment. Any proxy plan that fails to accomplish the original task is rejected. We use FastDownward [21] with ff-astar [22] to solve each proxy task, producing a set of candidate plans from which we select the one that minimizes Eq. (3).

B. Per-Robot Anticipatory Cost Estimation

The full joint evaluation needed for the exact computation of expected future cost via Eq. (1) is combinatorially expensive. Instead, our courteous anticipatory cost formulation of Eq. (3) factors the expected future cost to separately reason about the expected impact of the immediate plan on each robot’s next task, drawn from their respective responsibilities. Thus, we train a separate anticipatory cost estimator for each robot independently, where the expected future cost is defined by that robot’s responsibilities: a distribution over what tasks it may be assigned. At deployment, the planner queries all per-robot estimators and aggregates their outputs to approximate the expected future cost across all robots sharing the environment. Because each estimator is trained independently on a single robot’s responsibilities and implicitly captures the robot’s structural constraints, adding a robot to the shared environment requires only training that robot’s own estimator, without retraining of existing ones, a property we validate empirically in Sec. VI.

1) *Data Generation:* We generate training data from held-out training-time environments. It is assumed that each robot has direct access to its responsibilities, which specifies both the tasks it may be assigned and their relative likelihoods. Starting from the initial environment state, we solve tasks from these responsibilities using FastDownward [21] with an informed search (ff-astar [22]) and use the resulting plan costs to compute the anticipatory cost, $V_{\text{A.P.}}^r(s'_g)$, for that robot r . We represent the world as a sparse scene graph: nodes are entities with semantic and geometric attributes and edges encode spatial or logical relations. This graph defines our graph neural network (GNN), which estimates the expected cost of a state. Because this data is generated offline, we can afford more computation to approximate the expectation accurately; at deployment, we avoid rollouts and only query the learned estimators, keeping online planning fast.

2) *Per-Robot Anticipatory Cost Estimators:* Each per-robot estimator is a graph neural network (GNN) operating over the scene graph representation, trained independently on a single robot’s responsibilities. We use four GINConv layers followed by batch normalization and leaky ReLU activations. Node features are aggregated into a graph-level representation via both mean and sum pooling before a final linear layer produces the anticipatory cost estimate. At planning time, the planner queries all per-robot estimators over the candidate terminal states produced by focused sampling, aggregates

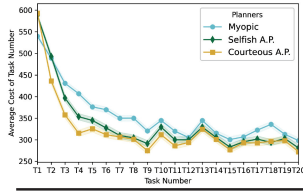


Table 1
AVERAGE COST-PER-TASK OVER 100
20-TASK SEQUENCE IN OUR
TWO-ROBOT HOME ENVIRONMENT

Approach	Avg. cost per task
Myopic Baseline	115s
Selfish Anticipatory Planning	108s
Courteous Anticipatory Planning	103s

Task 1: The *living-room-centric* robot is asked to **clear** the sofa (requires removing all items from the sofa).

Task 2: The *kitchen-centric* robot is asked to collect trash and put it in the bin and clear the counter.

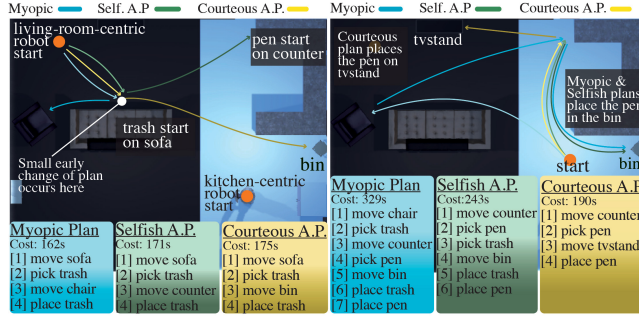


Fig. 3. **Courteous Planning in Home Environments:** (top) The per-task cost over the sequences, averaged over 100 20-task sequences. (bottom) An example from our results in which the living room organizer is first tasked to clear the sofa and then, later, the kitchen robot is assigned a task of its own. By clearing the sofa more courteously, our courteous living room organizer makes it easier for the kitchen robot later. See Sec. VI-A for details.

their outputs across all robots, and selects the plan whose terminal state minimizes Eq. (3).

VI. EXPERIMENTS AND RESULTS

We evaluate courteous anticipatory planning in two simulated environments of increasing complexity. In the home setting, courteous anticipation improves performance even without capability differences. In the restaurant environment, where robots have distinct capabilities and responsibilities, results validate that capability differences amplify the value of courteous anticipation.

In all experiments, planning cost reflects the physical time required to execute each action: only movement actions incur a dynamic cost proportional to travel distance, while other actions carry fixed unit costs. The specific manipulation actions vary by environment, with home environments involving pick and place and the restaurant environment additionally including cook, wash, wipe, restock, serve, pour and help, all evaluated under the same cost structure.

We compare against two baselines in both environments:

Myopic Baseline Classical planning via FastDownward

Selfish Anticipatory Planning (Selfish A.P.) Planning augmented with the acting robot’s own anticipatory cost estimator via Eq. (2), reducing cost for the acting robot’s future tasks but not for others sharing the environment.

Courteous Anticipatory Planning (Courteous A.P.)

Our approach, planning augmented with aggregated per-robot anticipatory cost estimates via Eq. (3), accounting for all robots sharing the environment.

A. Courteous Anticipation from Responsibilities Alone

The Home Environment: Home environments are curated from PROCTOR [23] and include two robots with similar capabilities but distinct responsibilities: rearrangement-style tasks curated from the ALFRED [24] task planning benchmark. Because both robots can physically interact with any asset (object and container) in the scene and perform the same actions, improvements arise solely from reasoning about distinct responsibilities and shared surfaces. The kitchen-centric robot’s responsibilities include clearing the counter, putting food in the fridge, and organizing other kitchen items: e.g., placing mugs and plates on shelves. The living-room-centric robot’s responsibilities include cleaning containers and organizing living room objects such as arranging pillows on the sofa or clearing the desk. Though robots are assigned tasks independently, their shared environment and surfaces create opportunities for courteous anticipation to reduce downstream friction, without explicitly encoding awareness of others’ responsibilities.

Qualitative Example: We include a specific example that illustrates this behavior in Fig. 3. First, the living-room organizer is tasked to *ClearSofa*, requiring it to remove all objects from the sofa (here, only a trash item). The myopic planner moves the trash to the nearby armchair without considering downstream consequences and the selfish planner places it on the counter outside the living room, considering only its own workspace. While our courteous planner places the trash directly in the bin, anticipating the kitchen organizer’s future responsibilities. So when the kitchen organizer is subsequently tasked to clear the counter and dispose of trash, our approach incurs lower cost than both baselines, which continue to produce side effects: both baseline planners leave the pen in the bin, increasing the cost of the living-room organizer’s next task.

Results: We conduct experiments consisting of 100 sequences of 20 tasks each (10 per robot, drawn from that robot’s responsibilities), and report statistics in Table 1 (Fig. 3). Courteous anticipation reduces cumulative planning cost by 10.4% versus the myopic planner and 4.0% versus the selfish anticipatory planner. As shown in the plot of average cost over the sequence in Fig. 3, because the PROCTOR are initialized with objects placed in somewhat random locations throughout the homes, all planners tend towards lower-expected-cost configurations over time, an observation shared by Talukder et al. [12]. However, our more courteous robots are able to leave objects in locations more favorable for *all* robots and improve over both baselines, made possible by their anticipation of how their actions will impact future tasks that both it and other robots may later be assigned. This improvement, achieved with similar capabilities and only distinct responsibilities, establishes a conservative lower bound on the value of courteous anticipation. As we show next, introducing distinct capabilities amplifies these gains substantially.

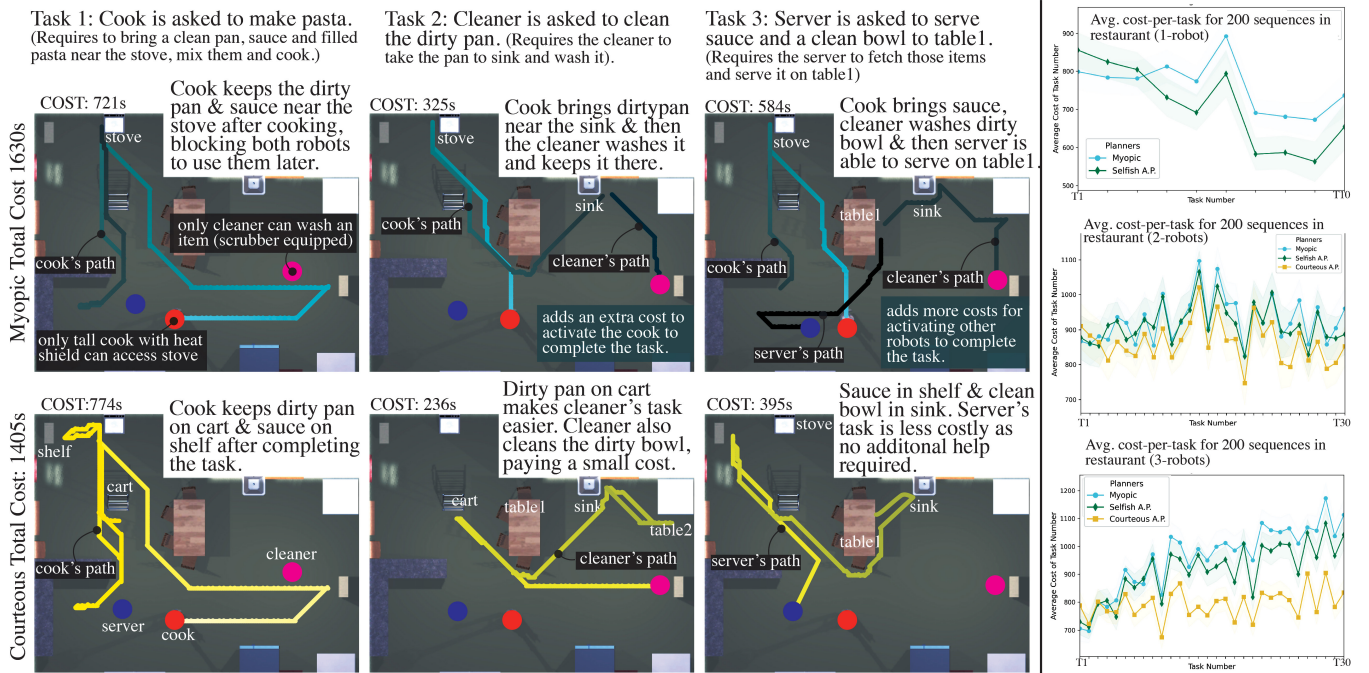


Fig. 4. **A Courteous Anticipation Sequence in the Restaurant Environment:** (left) An example sequence of three tasks that highlights the value of courtesy in our restaurant domain. Preemptive courteous action by the Cook during task 1 pays considerable dividends over time, improving the performance of both the Server and Cleaner later on as compared to the myopic planner, which has no such foresight. (right) Plots showing the average per-task performance over the task sequences for each of the 1-, 2-, and 3-robot settings. The 3-robot experiments in particular show the improvements over time afforded by our approach.

B. Distinct Capabilities Amplify Courteous Anticipation

The Restaurant Environment: To evaluate the full setting motivating our approach, we develop a restaurant-inspired benchmark environment with three heterogeneous robots that differ in physical capabilities and responsibilities, making one robot's decisions directly consequential for what others can accomplish. The *Cook* is tall, heat-resistant and alone can safely access the stove. The *Server* is also taller and alone can reach and serve items on dining table. The *Cleaner* is equipped with a scrubbing end-effector and alone can wash cookware. Consistent with their roles, the Cook's responsibilities include `MakeFood` objectives, the Cleaner's responsibilities include `CleanItem` and `ClearContainers` objectives, and the Server's responsibilities include `ServeFood` and `RestockFood` objectives.

We use two-room environments from PROCTOR as a base, populating them with domain-specific objects and actions suitable for food preparation and kitchen maintenance, from which complex multi-step tasks are constructed for each robot consistent with their roles.

Sometimes a robot cannot complete a task on its own because an object has been placed in a location it cannot physically access, or because an object is in a state it cannot change, for example a dirty pan that only the *Cleaner* can wash, or an empty pasta box on dining table that only the *Server* can fetch and restock. For this purpose, we add a `help` action that activates an otherwise *idle* robot—one not currently assigned a task—to satisfy the precondition on behalf of the acting robot. Because each robot has at least one

TABLE II
PER-ROBOT RESULTS AND IMPROVEMENTS.

Robots Present			Planner Cost			Improvement (%)	
Cook	Cleaner	Server	Myopic	Self. A.P.	Courteous	vs Myopic	vs Self. A.P.
✓	—	—	921	917*	917*	0.43%	N/A*
—	✓	—	456	411*	411*	9.86%	N/A*
—	—	✓	898	884*	884*	1.56%	N/A*
✓	✓	—	917	901	838	8.62%	7.01%
—	✓	✓	812	796	778	4.18%	2.26%
✓	—	✓	1052	1046	976	7.22%	6.69%
✓	✓	✓	873	831	721	17.41%	13.24%

Note: the differences between each robot configuration mean costs are not straightforwardly comparable across rows as they represent distinct planning problems.

*For single-robot settings, there are no other agents to be courteous to and so Courteous and Selfish (Self.) planning yield the same results.

exclusive action, such as inaccessible terminal state arise when robots place objects without regard for others' structural constraints. Invoking `help` accumulates an additional fixed cost of 100 sec, reflecting the real cost of interrupting an idle robot, incentivizing courteous behavior: a robot that reasons about what others can do and what they are likely to be assigned will leave objects in locations and states accessible to others, minimizing the need for costly assistance.

Qualitative Example: We highlight an illustrative example in Fig. 4 that shows the full three-robot restaurant, populated by the Cook, Server, and Cleaner. When tasked to `MakePasta`, our courteous anticipatory planner specifies a plan that fetches pasta, pours and cooks it, leaves the pasta box on the countertop while fetching sauce, mixes the

Given a task to MakeOats, courteous anticipation guides the cook robot to accommodate others' needs.

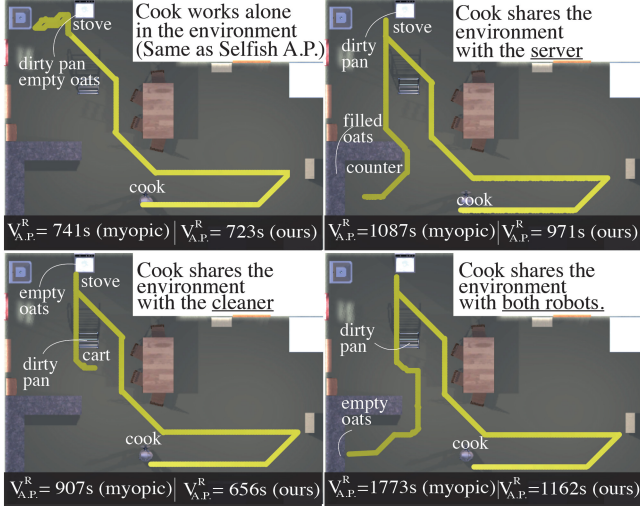


Fig. 5. **Courteous Planning in our Restaurant Environment:** Here we show an example of how the addition of other robots in the scene prompts behavior changes in our courteous robot. See Sec. VI-B for details.

sauce into the pasta, then stows the sauce in a shelf; on the return path the Cook additionally places the dirty pan directly on the bussing cart, for easy access by the cleaner. By contrast, the myopic planner tends to leave objects where they obstruct shared surfaces and on the hot stove, where other robots cannot access and so need costly assistance when assigned tasks of their own, poor-performing behavior shared by the selfish planner. Made other-agent-aware by our courteous anticipation, our Cook robot both effectively completes its task also expends additional effort to place objects in locations beneficial for the other robots, reducing cost overall.

Results: We evaluate performance beginning with a single robot, progressively adding robots to the shared environment, first two-robot subsets and then the full three-robot team, to showcase how courteous behavior evolves as robots are added or removed and to validate the modularity of our independently trained estimators. Across all configurations, we evaluate performance across 200 sequences of $10N$ tasks, where N is the number of robots present: i.e., that a 3-robot configuration would be given a total of 30 tasks. Our statistical results in Table II show that our courteous anticipatory planning approach outperforms myopic and selfish anticipatory planning by 17.4% and 13.2%, respectively, in the three-robot setting and by 6.7% and 5.5%, respectively, averaged over the two-robot settings, results that collectively demonstrate the value of courteous anticipation in shared persistent environments.

Modularity Validation. A key claim of our approach is that per-robot estimators trained independently can be composed without retraining when robots are added to or removed from the shared environment. Table II validates this directly: estimators trained in two-robot settings are reused unchanged in three-robot experiments with no joint

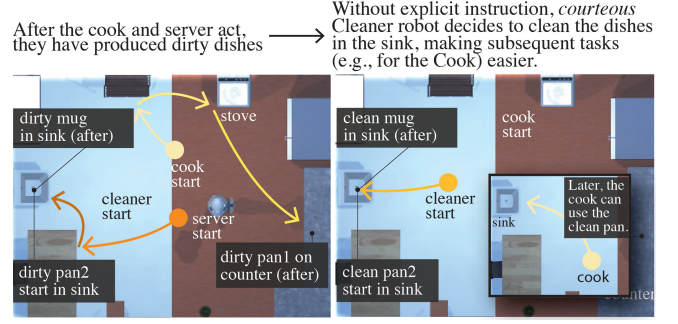


Fig. 6. **Proactivity via Courteous Anticipation:** In this scenario in our Restaurant domain, the Cleaner robot is not given explicit tasks and instead provided a small time budget to act during downtime when neither the Cook or Server are operating. Informed by its *courteous anticipation*, the Cleaner proactively cleans dirty dishes in the sink, action that later benefits the Cook. See Sec. VI-B for details.

retraining, confirming the plug-and-play property claimed in Sec. V-B. As expected, courteous and selfish planning yield identical behavior in the single-robot setting, confirming that courtesy emerges only when other robots are present.

Distinct Capabilities Drive Courteous Behavior. The robot-subset experiments, summarized in Fig. 5, reveal how distinct capabilities shape courteous behavior. When the Cook operates alone on *MakeOats*, courteous and selfish planning yield the same result. When the Server is added, the courteous Cook additionally moves finished items from the stove to the counter, since it knows the Server cannot interact with objects on the hot stove. When the Cleaner is added, the courteous Cook places dirty pans on the bussing cart where the Cleaner can reach them. When all three robots are present, the Cook exhibits both behaviors simultaneously—reasoning about what each robot can do and what they are likely to be assigned—important and effective behaviors not exhibited by myopic or selfish planners. This progressive behavioral change, driven by distinct capabilities, directly validates the core claim of our introduction: a robot must reason about what others can do, not only what they will do. Single-robot experiments are included for completeness, though when only the single robot is included both selfish and courteous anticipatory planning yield identical behavior.

Proactivity via Courteous Anticipation: We additionally conduct experiments in which one of the robots in our restaurant environment is given *no explicit task* and instead given a limited time budget to proactively ready the environment during downtime when the other two robots are not assigned tasks. Our experiments augment the two-courteous-robot Cook-Server experiments from the previous section and additionally include a free Cleaning robot that, after every two tasks, is given a cost budget of 100 sec to act in the environment to reach the state that maximally reduces the aggregated expected future cost for the Cook and Server robots; we use simulated annealing for this search process, iteratively perturbing the state space as computation allows.

Made possible by our courteous anticipatory planning

approach, the addition of the free courteous Cleaning robot reduces cost from 976 unit cost to 670 unit cost, a 31.4% improvement. Fig. 6 shows an instance of this behavior, where the Cleaner robot uses its time to proactively. Informed by its *courteous anticipation*, the Cleaner proactively cleans dirty dishes in the sink, an action that later benefits the Cook.

By contrast, a selfish or myopic robot would not take *any* action while idle, unaware of how its actions could help improve the performance of other robots in the scene. Thus, our courteous anticipation approach helps to motivate effective behaviors not exhibited by other approaches, affording improvements in persistent shared environments.

VII. LIMITATIONS AND FUTURE WORK

We have presented courteous anticipatory planning, a framework for reducing cumulative cost in persistent shared environments by reasoning about what other robots can do and what they are likely to be assigned. Our factored formulation estimates per-robot expected future costs independently, avoiding combinatorial joint rollouts and supporting modular deployment without retraining existing estimators. Across two environments of increasing complexity, amplified gains in the restaurant setting directly validate that distinct capabilities amplifies the value of courteous anticipation.

Our approach assumes tasks arrive sequentially, that the environment persists between tasks, which is reasonable for many shared environments but does not capture all possible real-world situations. We acknowledge that this setting does not capture all multi-agent challenges: tasks may arrive concurrently, robots may act simultaneously, and the environment may change mid-execution due to other agents or interruptions. However, we note that courteous anticipatory planning in shared environments is a symbolic planning behavior problem rather than a multi-agent coordination or communication problem. The multi-agent complexity we address is not joint policy learning [25], [26] or emergent coordination, but rather the challenge of planning with foresight about others, a challenge unaddressed by existing anticipatory planning methods [11]–[13].

In our future work, we plan to extend courteous anticipatory planning to concurrent settings, where multiple robots act simultaneously, and temporal planning over overlapping schedules is required.

REFERENCES

- [1] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, “PDDLStream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning,” in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 30, 2020, pp. 440–448.
- [2] E. Plaku and G. D. Hager, “Sampling-based motion and symbolic action planning with geometric and differential constraints,” in *2010 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 5002–5008.
- [3] M. Toussaint, “Logic-geometric programming: An optimization-based approach to combined task and motion planning,” in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
- [4] L. P. Kaelbling and T. Lozano-Pérez, “Hierarchical task and motion planning in the now,” in *2011 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2011, pp. 1470–1477.
- [5] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, “Combined task and motion planning through an extensible planner-independent interface layer,” in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 639–646.
- [6] B. Kim and L. Shimanuki, “Learning value functions with relational state representations for guiding task-and-motion planning,” in *Conference on Robot Learning*. PMLR, 2020, pp. 955–968.
- [7] B. Kim, Z. Wang, L. P. Kaelbling, and T. Lozano-Pérez, “Learning to guide task and motion planning using score-space representation,” *The International Journal of Robotics Research*, vol. 38, no. 7, pp. 793–812, 2019.
- [8] R. Chitnis, D. Hadfield-Menell, A. Gupta, S. Srivastava, E. Groshev, C. Lin, and P. Abbeel, “Guided search for task and motion plans using learned heuristics,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 447–454.
- [9] N. T. Dantam, Z. K. Kingston, S. Chaudhuri, and L. E. Kavraki, “Incremental task and motion planning: A constraint-based approach,” in *Robotics: Science and Systems*, vol. 12. Ann Arbor, MI, USA, 2016.
- [10] T. Silver, R. Chitnis, A. Curtis, J. B. Tenenbaum, T. Lozano-Pérez, and L. P. Kaelbling, “Planning with learned object importance in large problem instances using graph neural networks,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 13, 2021, pp. 11 962–11 971.
- [11] R. Dhakal, M. R. H. Talukder, and G. J. Stein, “Anticipatory planning: Improving long-lived planning by estimating expected cost of future tasks,” in *International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 538–11 545.
- [12] M. R. H. Talukder, R. I. Arnob, and G. J. Stein, “Anticipatory planning for performant long-lived robot in large-scale home-like environments,” in *International Conference on Robotics and Automation (ICRA)*, 2025.
- [13] R. Arora, S. Singh, K. Swaminathan, A. Datta, S. Banerjee, B. Bhowmick, K. 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 (ICRA)*, 2024.
- [14] M. Patel, A. G. Prakash, and S. Chernova, “Predicting routine object usage for proactive robot assistance,” in *7th Annual Conference on Robot Learning*, 2023.
- [15] M. Patel and S. Chernova, “Proactive robot assistance via spatio-temporal object modeling,” in *6th Annual Conference on Robot Learning*, 2022.
- [16] K. Sikes, S. Keren, and S. Sreedharan, “Reducing human-robot goal state divergence with environment design,” in *ICAPS 2024 Workshop on Human-Aware Explainable Planning*, 2024.
- [17] H. S. Koppula, A. Jain, and A. Saxena, “Anticipatory planning for human-robot teams,” in *Experimental Robotics: The 14th International Symposium on Experimental Robotics*. Springer, 2016, pp. 453–470.
- [18] E. Burns, J. Benton, W. Ruml, S. Yoon, and M. Do, “Anticipatory on-line planning,” *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 22, pp. 333–337, May 2012.
- [19] T. Q. Klassen, S. A. McIlraith, C. Muise, and J. Xu, “Planning to avoid side effects,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 9, 2022, pp. 9830–9839.
- [20] M. Fox and D. Long, “PDDL2.1: An extension to PDDL for expressing temporal planning domains,” *Journal of Artificial Intelligence Research*, vol. 20, pp. 61–124, 2003.
- [21] M. Helmert, “The Fast Downward planning system,” *Journal of Artificial Intelligence Research*, vol. 26, pp. 191–246, 2006.
- [22] J. Hoffmann and B. Nebel, “The FF planning system: Fast plan generation through heuristic search,” *J. Artif. Int. Res.*, vol. 14, no. 1, p. 253–302, May 2001.
- [23] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, J. Salvador, K. Ehsani, W. Han, E. Kolve, A. Farhadi, A. Kembhavi, and R. Mottaghi, “ProcTHOR: Large-Scale Embodied AI Using Procedural Generation,” in *NeurIPS*, 2022, outstanding Paper Award.
- [24] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, “ALFRED: A benchmark for interpreting grounding instructions for everyday tasks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 740–10 749.
- [25] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, “Multi-agent actor-critic for mixed cooperative-competitive

environments,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

- [26] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, “Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning,” in *International Conference on Machine Learning (ICML)*, 2018.