Anticipatory Planning for Human-Robot Teams

Hema Koppula, Ashesh Jain, and Ashutosh Saxena

Department of Computer Science, Cornell University.
{hema,ashesh,asaxena}@cs.cornell.edu

Abstract. When robots work alongside humans for performing collaborative tasks, they need to be able to anticipate human's future actions and plan appropriate actions. The tasks we consider are performed in contextually-rich environments containing objects, and there is a large variation in the way humans perform these tasks. We use a graphical model to represent the state-space, where we model the humans through their low-level kinematics as well as their high-level intent, and model their interactions with the objects through physically-grounded object affordances. This allows our model to anticipate a belief about possible future human actions, and we model the human's and robot's behavior through an MDP in this rich state-space. We further discuss that due to perception errors and the limitations of the model, the human may not take the optimal action and therefore we present robot's anticipatory planning with different behaviors of the human within the model's scope. In experiments on Cornell Activity Dataset, we show that our method performs better than various baselines for collaborative planning.

Keywords: Collaborative task planning, Anticipation, Human activity perception, Object affordances, Human-robot interaction.

1 Introduction

Currently, robots are being incorporated into human workspaces where they perform tasks with humans – assistive settings in nursing homes (e.g., [19]), collaborative assembly line manufacturing (e.g., [32]), or in other outdoor applications. The challenge here is two-fold: the robots often have to operate in *contextually-rich environments*, where they have to perform tasks involving manipulation of objects, and they have to work closely *with humans* performing the same task (see Fig. 1).

Collaborative tasks are more challenging as compared to both reactive and role-based tasks. In collaborative tasks, the goal of the robot is to perform actions along side humans in order to achieve the goal of the task. For example, if the task is to set the dinner table, the various actions involved are reaching for the objects (e.g., plates, cups and spoons), and moving them to appropriate locations on the table. The robot can perform any action in order to achieve the goal as opposed to a role-based scenario where the robot has a pre-assigned role of setting plates or cups, etc. It needs to plan its actions by taking into account the actions of the human. In order to achieve this, there are three aspects we need to address: (i) model the contextually-rich environment to reason about

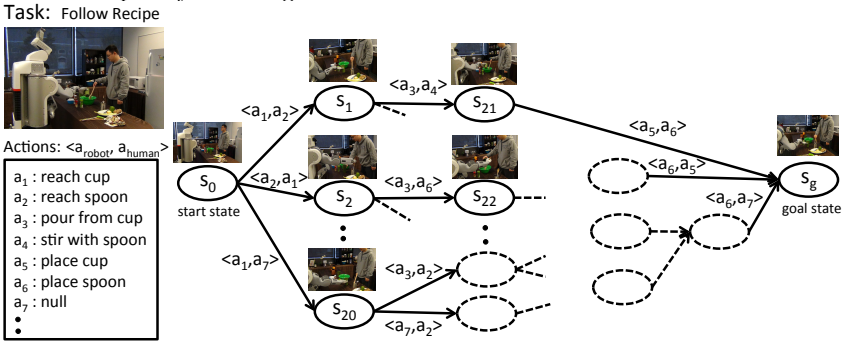


Fig. 1: **Robot performing collaborative task with human:** The human and the robot are asked to follow a recipe – pour the ingredients in to the bowl and stir. At every time step both the agents execute an action and change the state of the environment. The robot needs to plan its actions by taking in to account what actions the human would perform, where some human actions are more likely than the others based on the human's strategy. The challenge here is to model the rich environment context, and the ambiguity in human's actions.

what can be done and how, (ii) perceiving the human's actions and anticipating their future moves, and (iii) plan robot's actions taking into account the inherent uncertainty in the human actions.

In our previous works [21, 18, 19], we presented a perception algorithm for modeling the spatio-temporal relations of activities which allows us to detect the past actions and anticipate the future actions. However, the robotic responses were only reactive and hand-designed. In related works, Nikolaidis and Shah [33] consider collaboration for assembling tasks with pre-assigned roles for human and robot, where they do not explicitly model anticipation. Mainprice and Berenson [27] anticipate human actions to minimize penetration of robot in human workspace and Uyanik et al. [44] introduced social-affordances for planning. In comparison, we look at a more generic collaborative task planning problem, where the role of robot and human are indistinguishable.

In this paper, we formulate the collaborative task completion problem as a two-agent planning problem, where we model the ambiguities in perception as well as in the human's choice of actions. Unlike planning for multi-robot scenarios, where one has control over all agent behaviors [42, 3, 14], the human does not perform his actions according to a fixed strategy. Humans tend to follow their habits when possible in a familiar environment, but will also try to adapt in response to the other agents in the environment. Therefore, our problem of robot-human collaborative planning can be viewed as a two-agent cooperative Markov game, where the goal of each game is to complete a pre-specified activity in a given environment. We aim to learn the optimal policy for the robot while taking into account the various human behaviors or strategies.

In detail, we represent the contextually rich environment in terms of the object affordances and incorporate them as the states of our collaborative Markov decision process. We propose a distributed Q-learning algorithm to learn the policies for both the agents. We model the human's actions in several ways—taking the ϵ-optimal action according to the MDP model, taking actions based on past habits as seen in a RGB-D video dataset, and taking appropriate actions

by adapting to the environment and robot actions. Each human behavior results in exploring a different subspace of states by the robot, resulting in a different robot policy as shown in Fig. 1. Therefore, during learning we first estimate how adaptive the human would be in the given environment and then jointly estimate the robot and human policies.

We test our approach on five high-level activities in 60 environments from the CAD-250 dataset. We evaluate our algorithm both on the dataset as well as in a user study. We predict the current object affordances from RGB-D videos and use our algorithm to plan appropriate actions by the robot to be performed along with the human. We compare our approach against the baselines on several metrics, and find that our approach performs collaborative planning for the tasks better. Specifically, our robot policy learnt with an adaptive human model achieves the highest savings in task completion time of 36.5% as compared to 13.8% obtained when the human agent is not modeled explicitly.

2 Related Work

Our approach of anticipatory planning has three main aspects: human-robot interaction, perception in contextually-rich environments, and planning algorithms for finding the best action for the robot to perform. We now review the relevant works specific to these different aspects in more detail.

Human-robot collaboration. Many tasks are parallelizable or involve complex interactions with objects in the environment, and can be more efficiently completed if human and robot collaborate. Some recent works have addressed this problem of collaboration in human-robot teams. Nikolaidis et al. [33] consider collaboration for assembling tasks with pre-assigned roles for human and robot. Mainprice et al. [27] anticipates human actions to minimize penetration of robot in human workspace. Uyanik et al. [44] introduced social-affordance where robot's action depends on help from human. As opposed to them, in our work the role of robot and human are indistinguishable, and for task completion they interact with multiple objects performing different activities.

Another aspect of human-robot collaboration is the interaction between the agents and their compatibility. Some works [28, 38, 37, 29] encode the compatibility in the form of constraints on the distance of robot from user, the visibility of robot and user arm comfort. Strabala et al. [39] and Cakmak et al. [5] consider handover tasks wherein robot reason about its location w.r.t. human and object handover configuration. We differ from these in that, in our tasks both human and robot are active participants and collaborate towards a common goal.

Affordances. The concept of affordances was described by J.J. Gibson [8] as the "Action possibilities in the environment in relation to the action capabilities of an actor". Affordances have been widely used in robotics for obtaining a functional understanding of the scene as well as enabling robots to interact and manipulate objects. These works range from predicting opportunities for interaction with an object by using only visual cues [40, 9, 2] to observing effects of exploratory behaviors [31, 36, 30, 10, 13]. For instance, Sun et al. [40] proposed a probabilistic graphical model that leverages visual object categorization for learning affordances. Katz et al. [13] propose a framework for learning to manipulate objects in clutter by choosing robot actions based on object affordances.

There is some recent work in interpreting human actions and interaction with objects [25, 1, 17] in context of learning to perform actions from demonstrations. Lopes et al. [25] use context from objects in terms of possible grasp affordances to focus the attention of their recognition system. Aksoy et al. [1] construct a dynamic graph sequence of tracked image segments from human demonstrations and this representation is used by the robot for manipulating objects. Affordances have also been used in planning (e.g., [26, 43]). In this work, we use object affordances to represent the state of the environment, and these affordances evolve as the objects are used in an activity [20].

Multi-agent Reinforcement Learning. The multi-agent reinforcement learning (MARL) literature focus on multiple autonomous agents learning how to solve dynamic tasks online. Besides single-agent reinforcement learning, MARL has strong connections with game theory, evolutionary computation, and optimization theory. We refer the reader to [4] for a survey of the works in this area and discuss some relevant ideas here. Many multi-agent algorithms exist for different tasks which range from fully cooperative setting [22, 16, 48] to fully competitive setting [23, 15]. When collaborating with humans, the robot needs to be aware of the human's behavior, which might not always be fully cooperative.

Adaptation of agents has been studied previously [41, 34, 46], where an agent's adaptation depends on the degree of awareness of other agent's behavior maintained by the learning algorithms. These algorithms use some form of opponent modeling to keep track of the other agent's policies [6, 11]. There is a tradeoff between the stability (convergence) of the algorithms and the degree of adaptability. We build upon some of these ideas and propose a two-agent reinforcement learning algorithm, which models the various human behaviors allowing the robot to learn an adaptive policy.

3 Approach

Our goal is to learn which actions a robot can perform in order to collaborate with the human and assist in the task. As illustrated in Fig. 2, we first learn the spatio-temporal structure present in activities using a conditional random field (CRF) from RGBD videos of people performing these activities. We model the sub-activities and affordances of the objects, how they change over time, and how they relate to each other (for details see [21]). We then learn a Q-value function in simulation using the learnt activity and affordance models. When working with the human, the robot first estimates the state of the environment by detecting the object affordances and human actions, and then chooses an appropriate action and executes it.

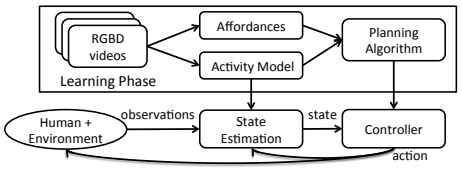


Fig. 2: **System Overview.**

In detail, we consider a robot r working with a human h in an environment having objects O. The goal is to learn a policy for the robot, π^r, which maps the current environment to an action. We formulate the collaborative task planning problem as a Markov decision process (MDP) with two agents – the human and the robot. We define the following:

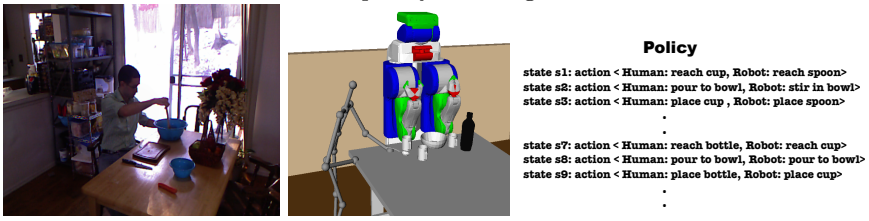


Fig. 3: **Collaborative planning by the robot.** In order to collaborate with the human on a *recipe following* task, the robot learns the activity model from RGB-D videos of human preparing a recipe (left), represents the environment via affordances and uses our planning algorithm (middle) to generate a policy for jointly performing the activity with the human (right).

- State Space $\mathcal{S}$: Let $\mathbf{s_t} = \{s_t^1, ..., s_t^n\}$ denote the state of the environment, where s_t^i denotes the state of the i^{th} object at time t and n denotes the number of objects.
- Action Space $\mathcal{A}$: Let $a_t = \langle a_t^h, a_t^r \rangle$ denote the joint action at time t, where a_t^h and a_t^r denote the human and robot actions respectively.
- Robot's policy $\pi^r : \mathcal{S} \times \mathcal{A}^r \rightarrow [0, 1]$, where $\mathcal{A}^r$ denotes the set of possible robot actions. $\pi^r(\mathbf{s}, a^r)$ specifies the probability of choosing action a^r in state $\mathbf{s}$.

We address the following challenging aspects of this problem: (i) Defining an efficient state-action representation that captures the contextually rich environments for performing complex activities, (ii) Learning the task model for each activity that specifies the effect of actions on the environment and also which actions need to be executed for completing the task, and (iii) Modeling the human agent's actions for learning the robot's policy.

3.1 Collaborative Markov Decision Processes

We use RGB-D videos of a single human performing the activities to define the state-action representation and learn the task model of the activities.[1] Once we have the set of states, set of actions and the task model, we can solve the MDP using dynamic programming techniques [35]. However, with large joint state-action space, computing the optimal policy is computationally very expensive and therefore, we take the model-free approach of Q-learning and learn the Q-functions offline with the help of the learnt task models. When collaborating with humans, the robot choses actions greedily with respect to its learnt Q-function. We fix the robot's policy after the offline learning, however, one can also further refine the Q-functions on-the-fly while working with humans in the real world. We now describe the details of our collaborative MDP algorithm.

State-Action Representation: We represent the environment in terms of the object affordances, which leads to an efficient state action space for planning. For example in Fig. 3, the state of the environment is represented in terms of the affordance labels of the objects in the scene, i.e., the bowl is *stirrable*, the spoon is the *stirrer* and the rest of the objects are *stationary*. The *stir* action corresponds to the temporal motion trajectory of the spoon from the grounded stir affordance. On performing the stir action, the spoon becomes *placeable*, thus changing the state of the environment.

[1] Such data is easier to collect for a wide variety of activities in a variety of environments [45, 20] as compared to collecting data of humans working with robots.

Algorithm 1 RUN-EPISODE (Q, π^h, π^r)

INPUT: State space S, Action space A

Initialize environment to start state; $R \leftarrow 0$; $i \leftarrow 0$;
loop
  if goal state **then**
    return Q-functions, R
  end if
  Sample a_h from π^h and a_r from π^r
  Take action (a_h, a_r) and observe r, $\mathbf{s}'$
  Update Q-functions as in Eq. 2
  $R \leftarrow R + \gamma^i * r$
  $i \leftarrow i + 1$
end loop

Task Model: State Transitions and Rewards. The affordance-based representation of the environment allows for factored representation of the transition and reward functions. That is, it is sufficient to specify the state transitions with respect to only a subset of affordances that are effected by an action. For example, a move action would change only the state of the movable object where as a pour action would change the state of the pourable and the pour-to objects. We assume that each action can be completed in one time step and hence given the nature of activities and affordances, the state transitions are deterministic. That is, on performing a valid action, the affordance of the object changes to another fixed affordance. The reward function allows us specify valid actions at any given affordance state, where all valid actions receive a fixed positive reward and non-valid actions will incur a negative cost. We also learn this task model (i.e., transition and the reward functions) from the labeled RGB-D videos of a human performing the activities.

Learning Robot Policy. Given the deterministic nature of the state transitions, we use the distributed Q-learning [22] algorithm to learn the local value functions $q_t^h(s,a)$ and $q_t^r(s,a)$ for the human and the robot respectively. Each agent assumes that other agents are acting optimally and only updates their local Q- functions when it results in an increase. This ensures that the local Q-value always captures the maximum of the joint-action Q-values. Therefore at each iteration, the local Q-functions are updated as in Eq. 1 while maintaining the invariants in Eq. 2.

$$q_{t+1}^j(\mathbf{s}_t, a_t^j) = \max\{q_t^j(\mathbf{s}_t, a_t^j),\ R(\mathbf{s}_t, a_t^h, a_t^r) + \gamma \max_{a \in \mathcal{A}^j} q_t^j(\mathbf{s}_{t+1}, a)\},\ j \in \{r, h\} \quad (1)$$

$$q_t^r(\mathbf{s}, a) = \max_{a^h \in \mathcal{A}^h} Q_t(\mathbf{s}, a^h, a^r = a); \quad q_t^h(\mathbf{s}, a) = \max_{a^r \in \mathcal{A}^r} Q_t(\mathbf{s}, a^h = a, a^r) \quad (2)$$

Our collaborative distributed Q-learning algorithm is summarized in Algorithm 1. Here, an episode is defined as the sequence of actions performed by the robot and human from the initial configuration to the goal configuration.

4 Models of Human Behavior

Many studies on human behavior have shown that there are primarily two systems which drive the way humans think – the first being fast, intuitive and emotional; and the second system which is slower, more deliberative and logical

[12]. This also applies to our problem of performing collaborative tasks, where humans can either think fast and perform activities following their habits or think more carefully about collaborating by taking into account what the robot can do. Therefore, the actions chosen by the human can range from fully cooperative, when humans are thinking for collaboration, to somewhat adversarial when their habits conflict with the robot's actions. Modeling these various types of human behavior becomes extremely important for collaboration.

Previous works in game theory literature study such senarios in the setting of general sum Markov games, where the types of opponent behaviors have been roughly classified into fixed strategies or best-response strategies [24]. In the fixed strategy case, the opponent always executes a fixed unknown policy and Q-learning finds the best response with respect to the fixed opponent. In the second case, it is assumed that the opponent adapts and chooses the best response so that it is mutually beneficial to both agents. Following these ideas, we model the following behaviors of a human agent:

- ***Habit-following human.*** In this model, we consider the perceptual data of the human from RGB-D videos, and assume that the human follows close to what he has done in the training videos. This is a fixed strategy behavior, where the human has a preferred way of performing activities and follows the same approach even when working with a robot. Let D be the set of activity videos and let $c(\mathbf{s}, a)$ be the number of times the human performed action a when in state $\mathbf{s}$ in D. The policy followed by a habit-following human, $\pi_d^h(\mathbf{s}, a)$, is defined as
$$\pi_d^h(\mathbf{s}, a_i) = \begin{cases} c(\mathbf{s}, a_i) / \sum_a c(\mathbf{s}, a) & \text{if } \mathbf{s} \in D \\ 1/n & \text{if } \mathbf{s} \notin D \\ 0 & \text{otherwise} \end{cases}$$
where n is the number of possible actions in state $\mathbf{s}$.
- ***ϵ-optimal human.*** In this model, we assume that the human takes the best action according to the value function most of the time, but makes a random choice ϵ fraction of the time. Here, human chooses a response that is mutually beneficial most of the time, according to the value function learnt so far.[2] This is equivalent to the ϵ-greedy exploration strategy which was shown to have better convergence properties [47] compared to always choosing the action greedily. The human policy is defined as

[2] The *ϵ-optimal human* behavior can differ from the *habit-following human* behavior, even when the reward model used to learn the value function for *ϵ-optimal human* is extracted from the the same data as the actions of *habit-following human*. There are two reasons for this: (i) The test environment is not present in the training data, and therefore, the reward function learnt from the training environments might not capture all valid ways of performing the activity in the test environment. This would lead to difference in what a human might do in this scenario and the policy learnt from an incomplete reward function; (ii) Humans can follow a different reward model when working alone as compared to when collaborating with others. Since we have adapted the reward function learnt from a single-agent scenario to a two-agent scenario, it is possible for the optimal-human policy to deviate from the habit-following human.

Algorithm 2 Learn Robot Policy

INPUT: State space S , Action space A, Data D

Initialize π^h and π^r uniformly
$Q \leftarrow 0$; $\eta \leftarrow 0.5$
while *burn-in period* **do**
  Sample $s \sim Bern(\eta)$; $s \in \{d, \epsilon\}$
  Update π^h_s
  $Q, R \leftarrow$RUN-EPISODE(Q, π^h_s, π^r)
end while
Update η
loop
  Update π^h_a using Eq. 3
  $Q, R \leftarrow$RUN-EPISODE(Q, π^h_a, π^r)
end loop
return Robot's Policy π_r

$$\pi^h_\epsilon(\mathbf{s}, a_i) = \begin{cases} (1-\epsilon) + (\epsilon/n) & \text{if } a_i = argmax_a(q^h(\mathbf{s}, a)) \\ \epsilon/n & \text{otherwise} \end{cases}$$

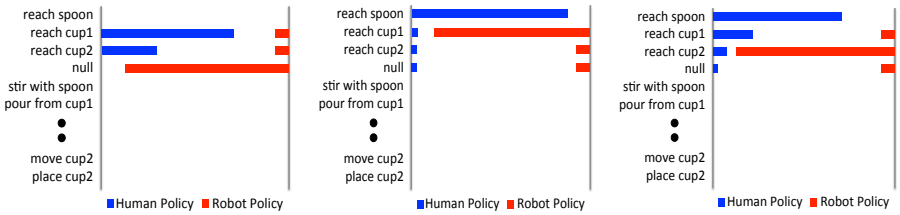
– ***Adaptive human.*** In the real world, when collaborating, humans usually adapt to other agents while trying to maintain their preferences or habits. That is, they follow their habits when possible in familiar situations, but when faced with new situations while working with the robot, they adapt and try to perform the action that is beneficial to both for completing the activity. We model this behavior by computing the probability of the human choosing one of the above two behaviors and define the human policy as

$$\pi^h_a(\mathbf{s}, a) = \eta * \pi^h_d(\mathbf{s}, a) + (1-\eta) * \pi^h_\epsilon(\mathbf{s}, a); \ \ \forall \mathbf{s}, a \tag{3}$$

where η denotes the probability of the human to follow habits.

During test time, when the robot is collaborating with the human on a new task, it should choose actions from the policy which is learnt with matching human behavior. One approach is to assume that the opponent type is known and fixed, and use the policy learnt with that type when executing the activities. Some works try to identify the opponents strategy on the fly and adapt accordingly. Such an approach requires the robot to work with humans to perform activities for a long time, which is not very practical in most scenarios. In contrast to these approaches, we present an algorithm (Algorithm 2) which adaptively selects the human's actions for exploration during the learning phase.

We need to estimate the Q-values along with the value of η, which is the probability with which the human follows his habits. This probability depends on the familiarity of the environment to the human as well as the cost of deviating from the optimal policy. Therefore, we model this probability as a function of the joint reward obtained when the human follows one of the two extreme behaviors – *habit-following* and ϵ-*optimal*, throughout the activity. Therefore, during an initial burn-in period, we sample the behavior uniformly and fix the behavior throughout an episode and learn the Q-values. We maintain a score for each of the behaviors, denoted by w_d and w_ϵ for the *habit-following* and ϵ-*optimal* behaviors



(a) Habit-following Human. (b) ϵ-optimal Human. (c) Adaptive Human.

Fig. 4: **Illustration of policies learnt with different human behaviors:** Each figure shows the learnt probability distributions of the various possible actions at the start state of the *following recipe* activity. Blue and red bars represent the probability of choosing the corresponding actions by the human and robot respectively.

respectively. At the end of Q-learning episode, we compute the normalized joint reward and update the corresponding score value as in Eq. 4.

$$\hat{R} \leftarrow \frac{R - \frac{r_{min}}{1-\gamma}}{\frac{r_{max}}{1-\gamma} - \frac{r_{min}}{1-\gamma}} \; ; \;\; w \leftarrow w + \alpha(\hat{R} - w) \tag{4}$$

where r_{max} and r_{min} are the maximum and minimum reward at any given time, respectively, γ is the discount factor and α is the learning rate. At the end of the burn-in period we compute the value of η as:

$$\eta \leftarrow \frac{e^{(w_d)}}{e^{(w_d)} + e^{(w_\epsilon)}} \tag{5}$$

We then continue learning the Q-values for the adaptive human by updating the human policy according to Eq. 3 using the estimated value of η.

Effect of the human behavior on the learned robot policies. Fig. 4 illustrates different human and robot policies corresponding to the different possible human behaviors we consider. Here, we consider an environment in which there are two cups, a bowl and a spoon, and the robot can only reach the cups and the bowl where as the human can reach all objects. The goal of the activity is to follow a recipe involving transfer of the ingredients from the cups to the bowl and mix them the spoon. In the training videos, at the beginning of the activity, the human reaches the first cup more often than the second cup as shown by the policy for the start state in Fig. 4-(a). The corresponding learned robot policy is to not do any action as reaching for a cup could result in a conflict.

Instead of following habits, if the human tries to optimizes for the joint reward, he would reach for the spoon and let the robot use the cups, which allows them to perform the activity together and complete it sooner. Fig. 4-(b) shows the policies corresponding to the ϵ-*optimal* human behavior. Given this environment, following habits turns out to be less rewarding and therefore a human would try to adapt more. This is reflected in the estimated value of η which is low for this particular scenario. Fig. 4-(c) shows our learnt *adaptive human* policy and the corresponding robot policy. Note that even small changes in human behavior can result in significant changes in the robot's actions.

5 Experiments

We test our proposed algorithm and other baseline methods for generating collaborative plans for several household activities. We evaluate the learnt robot

Table 1: **Collaborative Planning Evaluation.** Metrics computed for the collaborative plans generated on our RGB-D dataset.

Model	% time saving					% conflicts
	Recipe	Setting	Cleaning	Loading	Overall	
Human Expert Plans	36.8	53.1	16.4	42.4	37.2	0
Chance	3.3	10.5	-33.1	23.7	1.1	3.7
Mental-model MDP[33]	-2.6	30.4	-5.1	32.3	13.8	6.4
Our Model – ϵ-optimal human	27.5	45.6	18.3	30.8	31.2	13.5
Our Model – habit following human	28.4	48.1	18.6	41.4	33.4	11.9
Our Model – adaptive human	**32.8**	**48.5**	**22.9**	**41.9**	**36.5**	13.7

policies on both an activity dataset as well as in interaction with real humans. In this section we describe the data, experimental setup and the results.

Data: In order to evaluate our affordance and anticipatory planning models we expanded the CAD-120 dataset [21] to CAD-250 dataset, which has 130 additional RGB-D activity videos which contain more interesting object affordances and activities which allow human-robot collaboration. The sub-activities in the CAD-250 dataset include {*moving, stirring, pouring, drinking, cutting, eating, cleaning, reading, answering phone, wearing, exercising, hammering, measuring*} and the corresponding affordances are {*movable, stirrable, pourable, pourto, drinkable, cuttable, edible, cleanable, cleaner, readable, hearable, wearable, exercisable, hammer, hammerable, measurer, measurable*}.

We evaluated our planning algorithm on 60 RGB-D videos from the CAD-250 dataset which allow for collaboration. These activities include *two recipe making tasks*, *setting dinner table*, *cleaning house*, and *loading shelves*. These activities were performed by four subjects where each high-level activity is performed three times by each subject in a different environment. For each activity video, we labeled the sub-activities and the object affordances of the objects used in the activity. We will make this dataset publicly available.

Baselines: We compare our method against the following baselines:

- **Human Expert**: A human expert manually designed collaborative plans for each activity in the dataset.
- **Chance**: This algorithm chooses actions uniformly at random from the set of possible actions.
- **Mental-model MDP [33]**: We follow Nikolaidis et al. and define a MDP to formulate the robot's mental model [33]. In this approach, the human actions are incorporated into the state transition function and the policy specifies only the robot's actions. Therefore, we use the same state and action spaces and reward function as described in our approach with only one agent and compute the transition function from the state action sequences from the training data. Note that in our adaptation of [33] we fix the transition function learned from the data and do not perform any cross training iterations as the roles are fully exchangeable in our collaborative setting.

5.1 Evaluation on Data

We evaluate the generated collaborative plans on the following two metrics: (i) Percentage time saving: The percentage of savings in time for task completion is computed as $\frac{n_h - n_c}{n_h} * 100$, where n_h denotes the number of time steps taken if only human performs the task and n_c denote the number of steps to task completion following the collaborative plan. (ii) Percentage conflicts: The percentage of time steps robot's chosen action conflicted with that of the human.

For each activity video in the dataset, we give the environment extracted from the first frame and the goal state as input to the planning algorithm. We perform leave-one-out cross-validation and use the rest of the activity videos for learning the task model and the robot policies as described in Section 3.1 and Section 4. The sequence of human actions are taken form the test video and executed together with the robot actions specified by the learnt robot policy. Table 1 shows the results averaged for each high-level activity as well as for all activities in the dataset.

As can be seen in Table 1, our algorithm allows for more collaboration between human and the robot, resulting in higher savings in the time required for task completion compared to the baseline algorithms. When the robot chooses actions uniformly at random (Chance baseline), it sometimes chooses action sequences that help in achieving the goal sooner, but can also perform undesirable actions requiring additional time to complete the activity. Therefore, on average it does not result in any savings in the execution time. These results show that modeling the human actions along with the contextually rich environments is very important for collaborative planning.

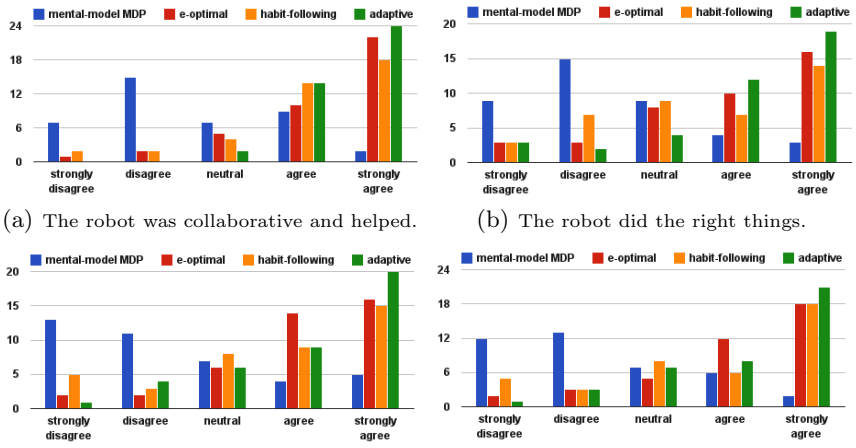
5.2 User Study

Experiment setup. We performed an user study with five subjects to evaluate the learned robot policies. We considered two high-level activities – *setting table* and *making recipe*, and four different environments for each activity. The subjects were asked to work with the robot to complete the tasks in a simulator. We re-created the environments in OpenRAVE [7] and provided an interface to the subjects to select an action they wish to execute. At every time instant, the users were shown the current state of the environment, and were asked to choose an action. The robot also selects an action based on the current state using its learned policy. Both the human and robot actions are then executed in the simulator. After completing each task, the users were asked to rate the following statements on Likert scale from 1 (strongly disagree) to 5 (strongly agree).

(a) The robot was collaborative and helped in the activity.
(b) The robot did the right thing at the right time.
(c) I am satisfied working with the robot.
(d) I will work with this robot again in future.

In this study, we compared the robot policies generated by the mental-model MDP [33] and our method learned with the three human behaviors. Therefore, every user performed each task four times, resulting in a total of 640 ratings.

Results. Fig. 5 shows the comparison of the user ratings for the four different robot policies on the four criteria mentioned above. Users rated the robot trained with our collaborative MDP model significantly higher ($p < 0.001$) than the robot using the mental-model MDP on all four criteria. For the robot policies learnt with our collaborative MDP using different human behaviors, when asked if they thought the robot did the right action at the right time, the users rated the robot trained with *adaptive human* higher than others ($p = 0.08$). For other criteria, there is no significant difference in the user ratings, however, as can be seen in Fig. 5, there is a slight preference for the robot trained with *adaptive*



(a) The robot was collaborative and helped.

(b) The robot did the right things.

(c) I am satisfied working with the robot.

(d) I will work with this robot again.

Fig. 5: **User Study Results:** The subjects collaborated with the robot on two different tasks in a total of eight different activities. They rated their experience based on four different criteria. The plots (a)-(d) show the comparison of the user ratings for four different robot policies – mental-model MDP [33], and our collaborative MDP trained with ϵ-*optimal human*, *habit-following human* and *adaptive human*.

Table 2: **Collaborative Planning Evaluation for User Study.** Metrics computed for the collaborative plans generated when working with humans during the user study.

Model	% time saving			% conflicts
	Recipe	Setting	Overall	
Mental-model MDP[33]	-0.9	13.9	6.5	2.7
Our Model – ϵ-optimal human	34.3	46.5	40.4	4.6
Our Model – habit following human	16.5	48.9	32.7	4.4
Our Model – adaptive human	**38.2**	**52.7**	**45.5**	4.6

human compared to others. We also compute the % saving in execution time and the % of conflicts for the collaborative plans generated in the user study. Table 2 summarizes these results. The users completed the tasks faster when working with robot trained with the *adaptive human* as compared to others.

5.3 Robot Experiment

We have also used the learned robot policy on our Kodiak (PR2) robot to work with a human on a *following recipe* task. Fig. 6 shows the robot collaborating with human to prepare a recipe, where the robot is executing the pour action as the human is stirring, based on its learnt policy. Videos showing the human and robot collaborating are available at: `http://pr.cs.cornell.edu/collaborativeplanning/`.

Fig. 6: Robot and human collaborating to prepare a recipe.

5.4 Discussion

We discuss the results of the evaluation on our dataset as well as the user study in the light of the following questions.

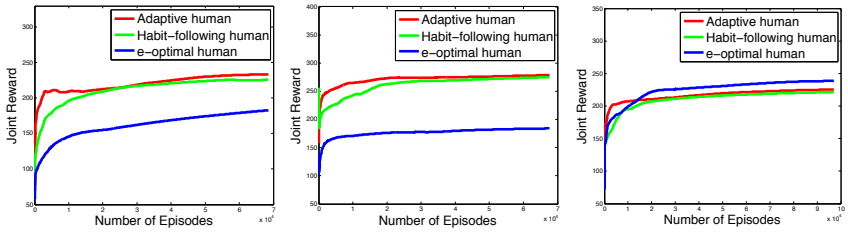


Fig. 7: **Joint reward received during Q-learning.** Plots showing the joint reward as a function of the number of Q-learning episodes for three different test environments of *setting table* activity.

What is the advantage of our collaborative MDP model over a single-agent MDP model? The mental-model MDP doesn't model the human explicitly as an agent but incorporates the effect of human actions in to the state transition probabilities. We observe a large variation in the performance of the mental-model MDP [33] baseline across the types of activities – it performs well on the setting table and loading shelf tasks, but takes longer to complete the recipe and cleaning tasks. It is interesting to note that the setting table and loading shelf tasks have a smaller action space as compared to the cleaning and recipe tasks. Therefore, given limited training data, the mental-model MDP is sensitive to the estimated state transition probabilities and fails when the action space is large. On the other hand, our collaborative MDP approach, which models the actions of human explicitly, overcomes this problem and performs significantly better on all tasks.

How important is modeling human behavior for collaboration? As humans tend to have specific preferences for executing tasks, the robot policy learnt with *habit-following human* strategy, which incorporates these preferences into planning, achieves an additional 2.2% saving in time compared to the ϵ-*optimal human*. However, when tested with new humans, whose habits were never seen in the training data, the robot policy learnt with ϵ-*optimal* human performs better (see Table 2). Modeling human as an adaptive agent always performs better and results in more collaboration – increasing the savings in the task completion time by 3.1% when working with a familiar human (human seen in the training data) and by 5.1% when working with a new human.

We also study how the joint reward evolves over the Q-learning episodes during training. At the end of each episode, we use the learned robot policy to perform the activity with a human following the policy π_d^h corresponding to the test environment. Fig. 7 shows the the joint reward received by the human and the robot as a function of the number of training episodes for three test environments, when training with the three human behaviors described in Section 4. We see that the policy learnt with the *adaptive human* converges to the highest joint reward much faster in most cases. However, in some cases (Fig. 7-right) the policy learnt with the *adaptive human* performs sub-optimally, due to the incorrect estimation of the adaptation probability.

How often does the robot conflict with human? The savings in task completion time increase as a result of the robot's increased participation in the task. This also leads to an increase in the % of conflicts between the robot's and

the human's actions. However, our model learnt with the *habit-following human* strategy reduces the % of conflicts compared to other baselines as it models the human's preferences. The number of conflicts again increase in case of *adaptive human* due to increased participation of the robot in the activity. When a conflict occurs, the preference is given to the human and the robot stops executing the action and chooses a new action in the next time step. In our current model we prefer plans with increased collaboration and do not penalize conflicts heavily. However, it is possible to modify the reward function to incorporate this, and we plan to explore this in future work.

6 Conclusion

In this work, we considered the problem of anticipatory planning for human robot teams, for enabling robots to work along side humans in contextually rich environments to accomplish complex tasks. We proposed a two agent collaborative MDP model and learn robot policies by taking into account the actions that can be performed by the human. We represented the contextually rich environments in terms of the object affordances and learn the activity model from RGB-D videos of a human performing the activities. We used this learned task model in a distributed Q-learning algorithm to learn the robot policy for a given new environment. We model the different possible human behaviors – taking the ϵ-optimal action according to the MDP model, taking actions based on past habits, and taking appropriate actions by adapting to the environment and robot actions. We tested our collaborative MDP model on the activity dataset as well as while directly interacting with humans in a user study. We show that explicitly modeling the human actions in the MDP formulation results in learning better robot policies. We also showed that changes in the human behavior can lead to significant changes in desirable robot actions. Therefore, modeling human behavior is essential for collaborative planning.

Bibliography

[1] E. E. Aksoy, A. Abramov, J. Dörr, K. Ning, B. Dellen, and F. Wörgötter. Learning the semantics of object-action relations by observation. *IJRR*, 30(10), 2011.

[2] A. Aldoma, F. Tombari, and M. Vincze. Supervised learning of hidden and non-hidden 0-order affordances and detection in real scenes. In *ICRA*, 2012.

[3] Jos Bento, Nate Derbinsky, Javier Alonso-Mora, and Jonathan S. Yedidia. A message-passing algorithm for multi-agent trajectory planning. In *NIPS*, 2013.

[4] L. Busoniu, R. Babuska, and B. De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE SMC, Part C*, 2008.

[5] M. Cakmak, S. .S. Srinivasa, M. K. Lee, J. F. Forlizzi, and S. Kiesler. Human preferences for robot-human hand-over configurations. In *IROS*, 2011.

[6] David Carmel and Shaul Markovitch. Opponent modeling in multi-agent systems. In *Adaption And Learning In Multi-Agent Systems*, volume 1042. 1996.

[7] R. Diankov. *Automated Construction of Robotic Manipulation Programs*. PhD thesis, CMU, RI, August 2010.

[8] J. J. Gibson. *The Ecological Approach to Visual Perception*. 1979.

[9] T. Hermans, J. M. Rehg, and A. Bobick. Affordance prediction via learned object attributes. In *ICRA Workshop on SPME*, 2011.

[10] T. Hermans, J. M. Rehg, and A. Bobick. Decoupling behavior, perception, and control for autonomous learning of affordances. In *ICRA*, 2013.
[11] J. Hu and M. P. Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *ICML*, 1998.
[12] Daniel Kahneman. *Thinking, fast and slow.* 2011.
[13] D. Katz, A. Venkatraman, M. Kazemi, J. A. Bagnell, and A. Stentz. Perceiving, learning, and exploiting object affordances for autonomous pile manipulation. In *RSS*, 2013.
[14] A. Kimmel and K. E. Bekris. Minimizing conflicts between moving agents over a set of non-homotopic paths through regret minimization. In *AAAI Workshop on Intelligent Robotic Systems*, 2013.
[15] K. Klein and S. Suri. Multiagent pursuit evasion, or playing kabaddi. In *WAFR*, 2010.
[16] S. Koenig, P. Keskinocak, and C. A Tovey. Progress on agent coordination with cooperative auctions. In *AAAI*, 2010.
[17] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto. Robot learning from demonstration by constructing skill trees. *IJRR*, 31, 2012.
[18] H. S. Koppula and A. Saxena. Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation. In *ICML*, 2013.
[19] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. In *RSS*, 2013.
[20] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *IJRR*, 2013.
[21] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *IJRR*, 2013.
[22] M. Lauer and M. Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *ICML*, 2000.
[23] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *ICML*, 1994.
[24] M. L. Littman. Friend-or-foe q-learning in general-sum games. In *ICML*, 2001.
[25] M. Lopes and J. Santos-Victor. Visual learning by imitation with motor representations. *IEEE SMC, Part B,*, 2005.
[26] C. Lorken and J. Hertzberg. Grounding planning operators by affordances. In *Int'l conf Cog Sys*, 2008.
[27] J. Mainprice and D. Berenson. Human-robot collaborative manipulation planning using early prediction of human motion. In *IROS*, 2013.
[28] J. Mainprice, E. A. Sisbot, L. Jaillet, J. Cortés, R. Alami, and T. Siméon. Planning human-aware motions using a sampling-based costmap planner. In *ICRA*, 2011.
[29] E. Meisner, V. Isler, and J. Trinkle. Controller design for human-robot interaction. *Autonomous Robots*, 24(2), 2008.
[30] B. Moldovan, M. van Otterlo, P. Moreno, J. Santos-Victor, and L. De Raedt. Statistical relational learning of object affordances for robotic manipulation. In *Latest Adv Inductive Logic Prog.*, 2012.
[31] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor. Learning object affordances: From sensory–motor coordination to imitation. *IEEE Trans Robotics*, 24(1):15–26, Feb. 2008.
[32] S. Nikolaidis and J. Shah. Human-robot teaming using shared mental models. In *HRI, Workshop on Human-Agent-Robot Teamwork*, 2012.
[33] S. Nikolaidis and J. Shah. Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy. In *HRI*, 2013.
[34] R. Powers and Y. Shoham. New criteria and a new algorithm for learning in multi-agent systems. In *NIPS*, 2004.

[35] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1994.
[36] B. Ridge, D. Skočaj, and A. Leonardis. Unsupervised learning of basic object affordances from object properties. In *CVWW*, 2009.
[37] E. A. Sisbot, L. F. Marin, and R. Alami. Spatial reasoning for human robot interaction. In *IROS*, 2007.
[38] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon. A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 2007.
[39] K. W. Strabala, M. K. Lee, A. D. Dragan, J. L. Forlizzi, S. .S. Srinivasa, M. Cakmak, and V. Micelli. Towards seamless human-robot handovers. *JHRI*, 2013.
[40] J. Sun, J. L. Moore, A. Bobick, and J. M. Rehg. Learning visual object categories for robot affordance prediction. *IJRR*, 2009.
[41] G. Tesauro. Extending q-learning to general adaptive multi-agent systems. In *NIPS*, 2004.
[42] M. Turpin, N. Michael, and V. Kumar. Trajectory planning and assignment in multirobot systems. In *WAFR*, 2012.
[43] E. Ugur, E. Sachin, and E. Oztop. Affordance learning from range data for multi-step planning. In *Epirob*, 2009.
[44] K. F. Uyanik, Y. Caliskan, A. K. Bozcuoglu, S. Kalkan O. Yuruten, and E. Sahin. Learning social affordances and using them for planning. In *CogSys*, 2013.
[45] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, 2012.
[46] T. Wongpiromsarn, A. Ulusoy, C. Belta, E. Frazzoli, and D. Rus. Incremental synthesis of control policies for heterogeneous multi-agent systems with linear temporal logic specifications. In *ICRA*, 2013.
[47] M. Wunder, M. L. Littman, and M. Babes. Classes of multiagent q-learning dynamics with epsilon-greedy exploration. In *ICML*, 2010.
[48] M. Zhu and S. Martínez. An approximate dual subgradient algorithm for multi-agent non-convex optimization. In *IEEE ICDC*, 2010.