Plan Explicability and Predictability for Robot Task Planning

Yu Zhang, Sarath Sreedharan, Anagha Kulkarni, Tathagata Chakraborti,
Hankz Hankui Zhuo and Subbarao Kambhampati

Abstract—**Intelligent robots and machines are becoming pervasive in human populated environments. A desirable capability of these agents is to respond to goal-oriented commands by autonomously constructing task plans. However, such autonomy can add significant cognitive load and potentially introduce safety risks to humans when agents behave unexpectedly. Hence, for such agents to be helpful, one important requirement is for them to synthesize plans that can be easily understood by humans. While there exists previous work that studied socially acceptable robots that interact with humans in "natural ways", and work that investigated legible motion planning, there lacks a general solution for high level task planning. To address this issue, we introduce the notions of plan *explicability* and *predictability*. To compute these measures, first, we postulate that humans understand agent plans by associating abstract tasks with agent actions, which can be considered as a labeling process. We learn the labeling scheme of humans for agent plans from training examples using conditional random fields (CRFs). Then, we use the learned model to label a new plan to compute its explicability and predictability. These measures can be used by agents to proactively choose or directly synthesize plans that are more explicable and predictable to humans. We provide evaluations on a synthetic domain and with human subjects using physical robots to show the effectiveness of our approach.**

I. INTRODUCTION

Intelligent robots and machines are becoming pervasive in human populated environments. Examples include robots for education, entertainment and personal assistance just to name a few. Significant research efforts have been invested to build autonomous agents to make them more helpful. These agents respond to goal specifications instead of basic motor commands, which requires them to autonomously synthesize task plans and execute those plans to achieve the goals. However, if the behaviors of these agents are incomprehensible, it can increase the cognitive load of humans and potentially introduce safety risks to them.

As a result, one important requirement for such intelligent agents is to ensure that the synthesized plans are comprehensible to humans. This means that instead of considering only the planning model of the agent, plan synthesis should also consider the interpretation of the agent behavior from the human's perspective. This interpretation is related to our modeling of other agents. More specifically, we tend to have expectations of others' behaviors based on our understanding (modeling) of their capabilities, mental states and etc. If their behaviors do not match with these expectations, we would often be confused. One of the major reasons of this confusion is due to the fact that our understanding of others' models is often partial and inaccurate. This is also true when humans

interact with intelligent agents. For example, to darken a room that is too bright, a robot can either adjust the window blinds, switch off the lights, or break the light bulbs in the room. While breaking the light bulbs may well be the least costly plan to the robot under certain conditions (e.g., when the robot cannot easily move in the environment but we are unaware of it), it is clear that the other two options are far more desirable in the context of robots cohabiting with humans. One of the challenges here is that the human's understanding of the agent model is inherently hidden. Thus, its interpretation from the human's perspective can be arbitrarily different from the agent's own model. While there exists previous work that studied socially acceptable robots [11, 12, 21, 18] that interact with humans in "natural ways", and work that investigated legible motion planning [6], there lacks a general solution for high level task planning.

In this paper, we introduce the notions of plan *explicability* and *predictability* which are used by autonomous agents (e.g., robots) to synthesize "explicable plans" that can be easily understood by humans. Our problem settings are as follows: an intelligent agent is given a goal by a human (so that the human knows the goal of the agent) working in the same environment and it needs to synthesize a plan to achieve the goal. As suggested in psychological studies [24, 5], we assume that humans naturally interpret a plan as achieving abstract tasks (or subgoals), which are functional interpretations of agent action sequences in the plan. For example, a robot that executes a sequence of manipulation actions may be interpreted as achieving the task of "*picking up cup*". Based on this assumption, intuitively, the easier it is for humans to associate tasks with actions in a plan, the more explicable the plan is. Similarly, the easier it is to predict the next task given actions in the previous tasks, the more predictable the plan is. In this regard, explicability is concerned with the association between human-interpreted tasks and agent actions, while predictability is concerned with the connections between these abstract tasks.

Since the association between tasks and agent actions can be considered as a labeling process, we learn the labeling scheme of humans for agent plans from training examples using conditional random fields (CRFs). We then use the learned model to label a new plan to compute its explicability and predictability. These measures are used by agents to proactively choose or directly synthesize plans that are more explicable and predictable without affecting the quality much. Our learning approach does not assume any prior knowledge
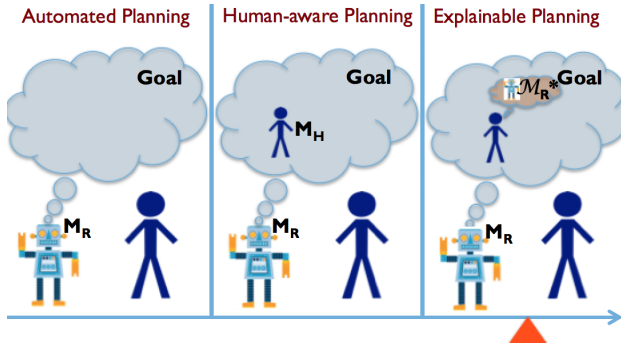
Fig. 1. From left to right, the scenarios illustrate the differences between automated task planning, human-aware planning and explicable planning (this work). In human-aware planning, the robot needs to maintain a model of the human (i.e., M_H) which captures the human's capabilities, intents and etc. In explicable planning, the robot considers the differences between its model from the human's perspective (i.e., $\mathcal{M}_R^*$) and its own model M_R.

on the human's interpretation of the agent model. We provide evaluation on a synthetic domain in simulation and with human subjects using physical robots to demonstrate the effectiveness of our approach.

II. RELATED WORK

To build autonomous agents (e.g., robots), one desirable capability is for such agents to respond to goal-oriented commands via automated task planning. A planning capability allows agents to autonomously synthesize plans to achieve a goal given the agent model (M_R as shown in the first scenario in Fig. 1) instead of following low level motion commands, thus significantly reducing the human's cognitive load. Furthermore, to work alongside of humans, these agents must be "human-aware" when synthesizing plans. In prior works, this issue is addressed under human-aware planning [22, 4, 2] in which agents take the human's activities and intents into account when constructing their plans. This corresponds to human modeling in human-aware planning as shown in the second scenario in Fig. 1. A prerequisite for human-aware planning is a plan recognition component, which is used to infer the human's goals and plans. This information is then used to avoid interference, and plan for serendipity and teaming with humans. There exists a rich literature on plan recognition [14, 3, 20, 16], and many recent works use these techniques in human-aware planning and human-robot teaming [23, 2, 25].

While our work on plan explicability and predictability falls within the scope of human-in-the-loop planning (which also includes human-aware planning), it differs significantly from the previous work. This is illustrated in Fig. 1. More specifically, in human-aware planning, the challenge is to obtain the human model (M_H in Fig. 1) which captures human capabilities [26], intents [23, 2] and etc. The modeling in this work is one level deeper: it is about the interpretation of the agent model from the human's perspective ($\mathcal{M}_R^*$ in Fig. 1). In other words, R needs to understand the model of *itself* in H's eyes. This information is inherently hidden, difficult to

convey, and can be arbitrarily different (e.g., having different representations) from R's own model (M_R in Fig. 1).

There exists work on generating legible robot motions [6] which considers a similar issue in motion planning. We are, on the other hand, concerned with task planning. Note that two different task plans may map to exactly the same motions which can be interpreted vastly differently by humans. In such cases, considering only motion becomes insufficient. Nevertheless, there exists similarities between [6] and our work. For example, legibility there is analogous to predictability in ours.

In the human-robot interaction (HRI) community, there exists prior works that discuss how to enable natural and fluent human-robot interaction [11, 12, 21, 18] to create more socially acceptable robots [7]. These works, however, apply only to behaviors in specific domains. Compared with model learning via expert teaching, such as inverse reinforcement learning [1] and tutoring systems [17], which is about learning the "right" model from teachers, our work, on the other hand, is concerned with learning model differences. Furthermore, as an extension to our work, when robots cannot find an explicable plan that is also cost efficient, they need to explain the situation. In this regard, our work is also related to excuse [9] and explanation generation [10]. Finally, while our learning approach appears to be similar to information extraction [19], we use the learned model to proactively guide planning instead of passively extracting information.

III. EXPLICABILITY AND PREDICTABILITY

In our settings, an agent R needs to achieve a goal given by a human in the same environment (so that the human knows about the goal of the robot). The agent has a model of itself (referred to as M_R) which is used to autonomously construct plans to achieve the goal. In this paper, we assume that this model is based on PDDL [8], a general planning domain definition language. As we discussed, for an agent to generate explicable and predictable plans, it must not only consider M_R but also $\mathcal{M}_R^*$, which is the interpretation of M_R from the human's perspective.

A. Problem Formulation

Keeping the problem settings in mind, given a domain, the problem is to find a plan for a given goal that satisfies the following:

$$\underset{\pi_{M_R}}{\operatorname{argmin}} \, cost(\pi_{M_R}) + \alpha \cdot dist(\pi_{M_R}, \pi_{\mathcal{M}_R^*}) \qquad (1)$$

where π_{M_R} is a plan that is constructed using M_R (i.e., the agent's plan), $\pi_{\mathcal{M}_R^*}$ is a plan that is constructed using $\mathcal{M}_R^*$ (i.e., the human's anticipation of the agent's plan), $cost$ returns the cost of a plan, $dist$ returns the distance (i.e., capturing the differences) between two plans, and α is the relative weight. The goal of Eq. (1) is to find a plan that minimizes a weighted sum of the cost of the agent plan and the differences between the two plans. Since the agent model M_R is assumed to be given, the challenge lies in the second part in Eq. (1).

Note that if we know $\mathcal{M}_R^*$ or it can be learned, the only thing left would be to search for a proper $dist$ function.

However, as discussed previously, $\mathcal{M}_R^*$ is inherently hidden, difficult to convey, and can be arbitrarily different from M_R. Hence, our solution is to use a learning method to directly approximate the returned values. We postulate that humans understand agent plans by associating abstract tasks with actions, which can be considered as a labeling process. Based on this, we assume that $dist(\pi_{M_R}, \pi_{\mathcal{M}_R^*})$ can be functionally decomposed as:

$$dist(\pi_{M_R}, \pi_{\mathcal{M}_R^*}) = F \circ \mathcal{L}^*(\pi_{M_R}) \tag{2}$$

where F is a domain specific function that takes plan labels as input, and $\mathcal{L}^*$ is the labeling scheme of the human for agent plans based on $\mathcal{M}_R^*$. As a result, Eq. (1) now becomes:

$$\underset{\pi_{M_R}}{\operatorname{argmin}} \, cost(\pi_{M_R}) + \alpha \cdot F \circ \mathcal{L}_{CRF}^*(\pi_{M_R} | \{S_i | S_i = \mathcal{L}^*(\pi_{M_R}^i)\}) \tag{3}$$

where $\{S_i\}$ is the set of training examples and $\mathcal{L}_{CRF}^*$ is the learned model of $\mathcal{L}^*$. We can now formally define plan explicability and predictability in our context. Given a plan of agent R as a sequence of actions, we denote it as π_{M_R} and simplified below as π for clarity:

$$\pi = \langle a_0, a_1, a_2, ...a_N \rangle \tag{4}$$

where a_0 is a null action that denotes plan starting. Given the domain, we assume that a set of task labels T is provided to label agent actions:

$$T = \{T_1, T_2, ...T_M\} \tag{5}$$

1) Explicability Labeling: Explicability is concerned with the association between abstract tasks and agent actions; each action in a plan is associated with an action label. The set of action labels for explicability is the power set of the task labels:

$$L = 2^T \tag{6}$$

When an action label includes multiple task labels, the action is interpreted as contributing to multiple tasks; when an action label is the empty set, the action is interpreted as inexplicable. When a plan is labeled, we can compute its explicability measure based on its action labels in a domain specific way. More specifically, we define:

Definition 1 (Plan explicability): Given a domain, the explicability θ_π of an agent plan π is computed by a mapping, $F_\theta : \mathbf{L}_\pi \to [0, 1]$ (with 1 being the most explicable). $\mathbf{L}_\pi$ above denotes the sequence of action labels for π. An example of F_θ used in our evaluation is given below:

$$F_\theta(\mathbf{L}_\pi) = \frac{\sum_{i \in [1,N]} \mathbf{1}_{L(a_i) \neq \emptyset}}{N} \tag{7}$$

where N is the plan length, $L(a_i)$ returns the action label of a_i, and $\mathbf{1}_{formula}$ is an indicator function that returns 1 when the $formula$ holds or 0 otherwise. Eq. (7) basically computes the ratio between the number of actions with non-empty action labels and the number of all actions.

2) Predictability Labeling: Predictability is concerned with the connections between tasks in a plan. An action label for predictability is composed of two parts: a current label and a next label (i.e., $L \times L$). The current label is also the action label for explicability. The next label (similar to the current label) is used to specify the tasks that are anticipated to be achieved next. A next label with multiple task labels is interpreted as having multiple candidate tasks to achieve next; when this label is the empty set, it is interpreted as that the next task is unpredictable, or there are no more tasks to be achieved.

Definition 2 (Plan Predictability): Given a domain, the predictability β_π of a plan π is computed by a mapping, $F_\beta : \mathbf{L}_\pi^2 \to [0, 1]$ (with 1 being the most predictable). $\mathbf{L}_\pi^2$ denotes the sequence of action labels for predictability. An example of F_β is given below which is used in our evaluation when assuming that the current and next labels are associated with at most one task label:

$$F_\beta(\mathbf{L}_\pi^2) = \frac{\sum_{i \in [0,N]} \mathbf{1}_{|L(a_i)|=1} \wedge \left(\mathbf{1}_{L^2(a_i)=L(a_j)} \vee \mathbf{1}_{L^2(a_{i:N})=\emptyset} \right)}{N+1} \tag{8}$$

where $a_j(j > i)$ is the first action that has a different current label as a_i or the last action in the plan if no such action is unfound, $L^2(a_i)$ returns the next label of a_i and $\mathbf{1}_{L^2(a_{i:N})=\emptyset}$ returns 1 only if the next labels for all actions after a_i (including a_i) are $\emptyset$. Eq. (8) computes the ratio between number of actions that we have correctly predicted the next task and the number of all actions.

B. A Concrete Example

Before discussing how to learn the labeling scheme of the human from training examples, we provide a concrete example to connect the previous concepts and show how training examples can be obtained. In this example, there is a rover in a grid environment working with a human. An illustration of this example is presented in Fig. 2. There are resources to be collected which are represented as boxes. There is one storage area that can store one resource which is represented as an open box. The rover can also make observations. The rover actions include $\{navigate \, l_{from} \, l_{to}\}$, $\{observe \, l\}$ $\{load \, l\}$, and $\{unload \, l\}$, each representing a set of actions since l (i.e., representing a location) can be instantiated to different locations (i.e., $0 - 8$ in Fig. 2). $navigate$ (or nav) can move the rover from a location to one of its adjacent locations; $load$ can be used to pick up a resource when the rover is not already loaded; $unload$ can be used to unload a resource at a storage area if the area is empty; $observe$ (or obs) can be used to make an observation. Once a location is observed, it remains observed. The goal in this example is for the rover to make the storage area non-empty and observe two locations that contain the eye symbol in Fig. 2.

In this domain, we assume that there are three abstract tasks that may be used by the human to interpret the rover's plans: COLLECT (C), STORE (S) and OBSERVE (O). Note that we do not specify any arguments for these tasks (e.g., which resource the rover is collecting) since this information may not be important to the human. This also illustrates that M_R and
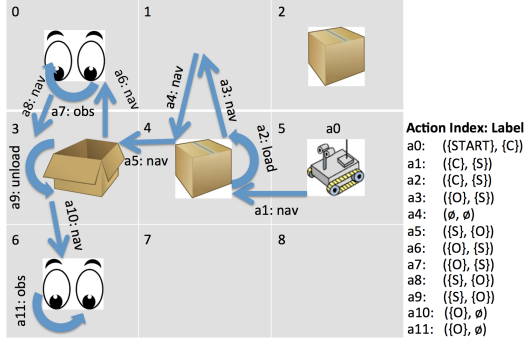
Fig. 2. Example for plan explicability and predictability with action labels (on the right) for a given plan in the rover domain.

$\mathcal{M}_R^*$ can be arbitrarily different. In Fig. 2, we present a plan of the rover as connected arrows starting from the its initial location.

Human Interpretation as Training Examples: Let us now discuss how humans may interpret this plan (i.e., associating labels with actions) as the actions are observed *incrementally*: when labeling a_i, we only have access to the plan prefix $\langle a_0, ..., a_i \rangle$. At the beginning for labeling a_0, the observation is that the rover starts at l_5. Given the environment and knowledge of the rover's goal, we may infer that the first task should be COLLECT (the resource from l_4). Hence, we may choose to label a_0 as ({START}, {C}). The first action of the rover (i.e., $nav\ l_5\ l_4$) seems to match with our prediction. Furthermore, given that the storage area is closest to the rover's location after completing COLLECT, the next task is likely to be STORE. Hence, we may label a_1 as ({C}, {S}) as shown in the figure. The second action (i.e., $load\ l_4$) also matches with our expectation. Hence, we label a_2 too as ({C}, {S}). The third action, $nav\ l_4\ l_1$, however, is unexpected since we predicted STORE in the previous steps. Nevertheless, we can still explain it as contributing to OBSERVE (at location l_0). Hence, we may label this navigation action (a_3) as ({O}, {S}). For the fourth action, the rover moves back to l_4, which is inexplicable since the rover's behavior seems to be oscillating without particular reasons. Hence, we may choose to label this action as ($\emptyset, \emptyset$). The labeling for the rest of the plan continues in a similar manner. This thought process reflects how training examples can be obtained from human labelers.

IV. LEARNING APPROACH

To compute θ_π and β_π from Defs. (1) and (2) for a given plan π, the challenge is to provide a label for each action. This requires us to learn the labeling scheme of humans (i.e., $\mathcal{L}^*$ in Eq. (2)) from training examples and then apply the learned model to π (i.e., $\mathcal{L}_{CRF}^*$ in Eq. (3)). To formulate a learning method, we consider the sequence of labels as hidden variables. The plan that is executed by the agent (which also captures the state trajectory), as well as any cognitive cues that may be obtained (e.g., from sensing) during the plan execution constitute the observations. The graphical model that we choose for our learning approach is conditional random

fields (CRFs) [15] due to their abilities to model sequential data. An alternative would be HMMs; however, CRFs have been shown to relax assumptions about the input and output sequence distributions and hence are more flexible.

The distributions that are captured by CRFs have the following form:

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \Pi_A \Phi(\mathbf{x}_A, \mathbf{y}_A) \tag{9}$$

in which Z is a normalization factor that satisfies:

$$Z = \sum_{\mathbf{x}, \mathbf{y}} \Pi_A \Phi(\mathbf{x}_A, \mathbf{y}_A) \tag{10}$$

In the equations above, $\mathbf{x}$ represents the sequence of observations, $\mathbf{y}$ represents the sequence of hidden variables, and $\Phi(\mathbf{x}_A, \mathbf{y}_A)$ represents a factor that is related to a subgraph in the CRF model associated with variables $\mathbf{x}_A$ and $\mathbf{y}_A$. In our context, $\mathbf{x}$ are the observations made during the execution of a plan; $\mathbf{y}$ are the action labels. Each factor is associated with a set of features that can be extracted during the plan execution. Next, we discuss some possible features that can be used for plan explicability and predictability.

A. Features for Learning

Given an agent plan, the immediate set of features that we have access to is the plan and its associated state trajectory. Note that the human may not be required (nor it is necessary) to fully understand this information. When the dynamics of the agent are known, given the plan, it may also be possible to derive low level motor commands that implement the motions, which can be used to extract motion related features.

When the agent is equipped with sensors such as cameras and lasers, we can also extract features from sensor information. For example, from video streams and depth information, we can extract features about the environment, e.g., how crowded the workspace is.

Sensor information can also be used to extract dynamic features such as the location of the human. However, note that this information will not be available during the testing phase, and thus these features need to be estimated based on other information (e.g., projected plan of the human based on plan recognition techniques [20, 16]).

In this work, we use a linear chain CRF. However, our formulation is easily extensible to more general types of CRFs. Given an agent plan $\pi = \langle a_0, a_1, a_2, ... \rangle$, each action is associated with a set of features. Hence, each training example is of the following form:

$$\langle (F_0, L_0^2), (F_1, L_1^2), (F_2, L_2^2), ... \rangle \tag{11}$$

where L_i^2 is the action label for predictability (and explicability) for a_i. F_i is the set of features for a_i. We discuss several feature categories in more detail below:

1) Plan Features: Given the agent model (specified in PDDL), the set of plan features for a_i includes the action description and the state variables after executing the sequence of actions $\langle a_0, ..., a_i \rangle$ from the initial state. This information can be easily extracted given the model. For example, in our rover example in Fig. 2, this set of features for a_1 includes *navigate*, *at rover* l_4, *at resource0* l_2, *at resource1* l_4, *at storage0* l_3.

2) Action Features: Action features for a_i describes the motion (e.g., dynamics) of this action. These features can be used to capture, for example, smoothness of execution within and across actions. Action features sometimes serve as important cognitive cues for humans to understand agent actions. For example, an action that enables a robot to cross a river may be interpreted as *swimming*, *pedaling*, or *propelling* depending on how the robot motion looks like. Action features can be extracted for a plan given the dynamics of the robot.

3) Interaction Features: Interaction features are intended to capture a_i's influence on the human. For example, it can include how far the agent is from human and what the human is performing when a_i is being executed. In other words, this set of features captures characteristics of the interactions between the human and agent. Interaction features can be extracted from sensor information or estimated based on the projected human plan.

B. Using the Learned Model

Given a set of training examples in the form of Eq. (11), we can train the CRF model to learn the labeling scheme in Eq. (3). We discuss two ways to use the learned CRF model.

1) Plan Selection: The most straightforward method is to perform plan selection on a set of candidate plans which can simply be a set of plans that are within a certain cost bound of the optimal plan. Candidate plans can also be generated to be diverse with respect to various plan distances. For each plan, the agent must first extract the features of the actions as we discussed earlier. It then uses the trained model (i.e., $\mathcal{L}^*_{CRF}$) to produce the labels for the actions in the plan. θ and β can then be computed given the mappings in Defs. (1) and (2). These measures can then be used to choose a plan that is more explicable and predictable.

2) Plan Synthesis: A more efficient way is to incorporate these measures as heuristics into the planning process. Here, we consider the FastForward (FF) planner with enforced hill climbing [13]. To compute the heuristic value given a planning state, we use the relaxed planning graph to construct the remaining planning steps. However, since relaxed planning does not ensure a valid plan, we can only use action descriptions as plan features for actions that are beyond the current planning state when estimating the θ and β measures. These estimates are then combined with the relaxed planning heuristic (which only considers plan cost) to guide the search. The algorithm for generating explicable and predictable plans is presented in Alg 1.

The capability to synthesize explicable and predictable plans is useful for autonomous agents. For example, in domains

---

Algorithm 1 Synthesizing Explicable and Predictable Plans

Input: agent model M_R, trained human labeling scheme $\mathcal{L}^*_{CRF}$, initial state I and goal state G.

Output: π_{EXP}

1: Push I into the open set O.
2: **while** open set is not empty **do**
3:    $s = \text{GetNext}(O)$.
4:    $h^* = MAX$.
5:    **if** G is reached **then**
6:       **return** s.plan (i.e., the plan that leads to s from I).
7:    **end if**
8:    Compute all possible next states N from s.
9:    **for** $n \in N$ **do**
10:      Compute the relaxed plan π_{RELAX} for n.
11:      Concatenate s.plan (with plan features) with π_{RELAX} (with only action descriptions) as $\bar{\pi}$.
12:      Compute and add other relevant features.
13:      Compute $\mathbf{L}^2_\pi = \mathcal{L}^*_{CRF}(\bar{\pi})$.
14:      Compute θ and β based on $\mathbf{L}^2_\pi$ for $\bar{\pi}$.
15:      Compute $h = f(\theta, \beta, h_{cost})$ (f is a combination function; h_{cost} is the relaxed planning heuristic).
16:    **end for**
17:    Find the state $n^* \in N$ with the minimum h.
18:    **if** $h(n^*) < h^*$ **then**
19:      Clear O.
20:      Push n^* into O.
21:    **else**
22:      Push all $n \in N$ into O.
23:    **end if**
24: **end while**

---

where humans interact closely with robots (e.g., in an assembly warehouse), more preferences should be given to plans that are more explicable and predictable since there would be high risks if the robots act unexpectedly. One note is that the relative weights of explicability and predictability may vary in different domains. For example, in domains where robots do not engage in close interactions with humans, predictability may not matter much.

V. EVALUATION

We first evaluate our approach systematically on a synthetic dataset based on the rover domain. Then, we evaluate it with human subjects using physical robots to validate that the synthesized plans are more explicable to humans in a blocks world domain.

A. Systematic Evaluation with a Synthetic Domain

The aim is twofold here: evaluate how well the learning approach can capture an arbitrary labeling scheme; evaluate the effectiveness of plan selection and synthesis with respect to the θ and β measures.

1) Dataset Synthesis: To simplify the data synthesis process, we make the following assumptions: all rover actions have the same cost; all rover actions are associated with at

most one task label (i.e., $L = T \cup \{\emptyset\}$ in Eq. (6)). To construct a domain in which the optimal plan (in terms of cost) may not be the most explicable (in order to differ M_R from $\mathcal{M}_R^*$), we add "oscillations" to the plans of the rover. These oscillations are incorporated by randomly adding locations for the rover to visit as *hidden goals*. For these locations, the rover only needs to visit them. As a result, it may demonstrate "unexpected" behaviors given only the *public goal*, denoted by G, which is known to both the rover and human. We denote the goal that also includes the hidden goals as G'. Given a problem with a public goal G, we implement a labeling scheme to automatically provide the "ground truth" of a rover plan, which is constructed by the rover to achieve G'.

Given a plan of the rover, we label it incrementally by associating each action with a current and next label. These labels are chosen from $\{\{\text{COLLECT}\}, \{\text{STORE}\}, \{\text{OBSERVE}\}, \emptyset\}$. We denote the plan prefix $\langle a_0, ...a_i \rangle$ for a plan π as π_i, the state after applying π_i as s_i from the initial state, and a plan that is constructed from s_i to achieve G (i.e., using s_i as the initial state) as $P(s_i)$. For the current label of a_i:

1) If $|P(s_i)| \geq |P(s_{i-1})|$, we label a_i as $\emptyset$ (i.e., inexplicable). This rule means that humans may label an action as inexplicable if it does not contribute to achieving G.
2) If $|P(s_i)| < |P(s_{i-1})|$, we label a_i based on the distances from the current rover location to the targets (i.e., storage areas or observation locations), current state of the rover (i.e., loaded or not), and whether a_i moves the rover closer to these targets. For example, if the closest target is a storage area and the rover is loaded, we label a_i as $\{\text{STORE}\}$. When there are ties, we label a_i as $\emptyset$ (i.e., unclear and hence interpreted as inexplicable).

For the next label of a_i:

1) This label is determined by the target that is closest to the rover state after the current task is achieved. When there are ties, a_i is labeled as $\emptyset$ (i.e., unclear and hence interpreted as unpredictable). If the current label is $\emptyset$, we also label a_i as $\emptyset$ (i.e., unpredictable).
2) If the current task is also the last task, we label a_i as $\emptyset$ since there is no next task.

For evaluation, we define F_θ and F_β as in Eqs. (7) and (8). We randomly generate problems in a 4×4 environment. For each problem, we randomly generate $1-3$ resources as a set RE, $1-3$ storage areas as a set ST, $1-3$ observation locations as a set OB. The public goal G of a problem, first, includes making all storage areas non-empty. To ensure a solution, we force $|RE| = |ST|$ if $|RE| < |ST|$. Furthermore, the rover must make observations at the locations in OB. G' for the rover includes G above, as well as a set of hidden goals. Locations of the rover, RE, ST, OB and hidden goals are randomly generated in the environment and do not overlap in the initial state. Although seemingly simple, the state space of this domain is on the order of 10^{20}.

2) Results: We use only plan features here. First, we evaluate our approach to learning the labeling scheme (i.e., $\mathcal{L}_{CRF}^*$) as the difference between M_R and $\mathcal{M}_R^*$ gradually
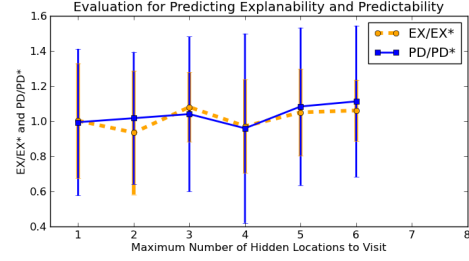

Fig. 3. Evaluation for predicting θ and β measures as the difference between M_R and $\mathcal{M}_R^*$ increases (i.e., as the maximum number of hidden goals increases).

increases (i.e., as the number of hidden goals increases). Afterwards, we evaluate the effectiveness of plan selection and synthesis with respect to the θ and β measures. To verify that our approach can generalize to different problem settings, we fix the level of oscillation when generating training samples while allowing it to vary in testing samples.

Using CRFs for Plan Explicability and Predictability: In this evaluation, we randomly generate $1 - 3$ hidden goals to include in G' in 1000 training samples. After the model is learned, we evaluate it on 100 testing samples in which we vary the maximum number of hidden goals from 1 to 6 with step size 1. The result is presented in Fig. 3. We can see that the prediction performance (i.e., the ratios between θ and β computed based on $\mathcal{L}_{CRF}^*$ and $\mathcal{L}^*$) is generally between $50\% - 150\%$, We can also see that the oscillation level does not seem to influence the prediction performance much. This shows that our approach is effective whether M_R and $\mathcal{M}_R^*$ are similar or largely different.

Selecting Explicable and Predictable Plans: We evaluate plan selection using θ and β measures and compare the selected plans (denoted by EXPD-SELECT) with plans selected by a baseline approach (denoted by RAND-SELECT). Given a set of candidate plans, EXPD-SELECT selects a plan according to the highest predicted explicability or predictability measure while RAND-SELECT randomly selects a plan from the set of candidate plans. To implement this, for a given public goal G, we randomly construct 20 problems with a given level of oscillation as determined by the maximum number of hidden goals. Each such problem corresponds to a different G' and a plan is created for it. The set of plans for these 20 problems associated with the same G is the set of candidate plans for G. For each level of oscillation, we randomly generate 50 different Gs and then construct the set of candidate plans for each G. The model here is trained with 1900 samples using the same settings as in our first evaluation and we gradually increase the level of oscillation.

We compare the θ and β values computed from the ground truth labeling of the chosen plans. The result is provided in Fig. 4. When the oscillation is small, the performances of both approaches are similar. As the oscillation increases, the performances of the two approaches diverge. This is expected since RAND-SELECT randomly chooses plans and hence its performance should decrease as the oscillation increases. On
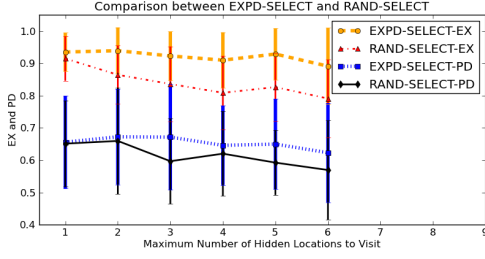
Fig. 4. Comparison of EXPD-SELECT and RAND-SELECT

Fig. 5. Comparison of FF and FF-EXPD considering only θ.

Fig. 6. Comparison of FF and FF-EXPD considering only β.

TABLE I
PLAN STEPS COMPARISON FOR FIG. 6

Trial ID	1	2	3	4	5	6
FF (avg. # steps)	21.9	24.0	24.1	23.9	22.1	22.4
FF-EXPD (avg. # steps)	23.5	26.3	25.2	24.0	23.4	25.0

the other hand, EXPD-SELECT is not influenced as much although its performance also tends to decrease. This is partly due to the fact that the model used in this evaluation is trained with samples having a maximum of 3 hidden goals.

In Fig. 4 for explicability, almost all results are significantly different at 0.001 level (except at 1); for predictability, results are significantly different at 0.01 level at 3, 5 and 6. The trend to diverge is clearly present. Note that we use linear-chain CRFs in our evaluations, which does not directly model correlations among observations across states. These features are common in our rover domain (e.g., navigating back and forth). Hence, we can anticipate performance improvement with more general CRFs.

Synthesizing Explicable and Predictable Plans: We evaluate here plan synthesis using Alg. 1. More specifically, we compare FF planner that considers the predicted θ and β values in its heuristics with a normal FF planner that only considers the action cost. The FF planner with the new heuristic is called FF-EXPD. In this evaluation, we set the maximum number of hidden locations to visit to be 6. For each trial, we generate 100 problems and apply both FF and FF-EXPD to solve the problems. Given that we are interested in comparing the cases when explicability is low, we only consider problems when the predicted plan explicability for the plan generated by FF is below 0.85.

First, we consider the incorporation of θ only. The result is presented in Fig. 5. For the explicability measure, we see a significant difference in all trials. Another observation is that the difference in plan predictability is present but not as significant. This evaluation suggests that our heuristic search can produce plans of high explicability.

Next, we consider the incorporation of β only. The result is presented in Fig. 6. Similarly, we see a significant difference in all trials for both explicability and predictability. One observation is that improving on plan predictability also improves plan explicability which is expected given Eqs. (7) and (8)).

Plan Cost: We consider plan cost here for the evaluation in Fig. 6. The result is presented in Table I. We can see that the plan length for FF-EXPD is longer than the plan produced by FF in general. This is expected since FF only considers plan cost. However, in all settings, FF-EXPD penalizes the plan cost slightly (about 10%) to improve the plan explicability and predictability measures.
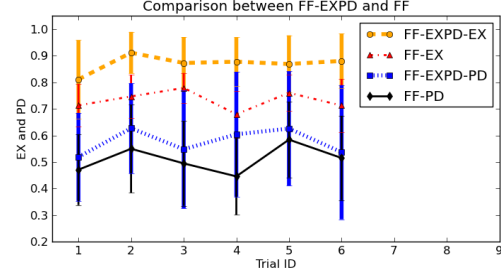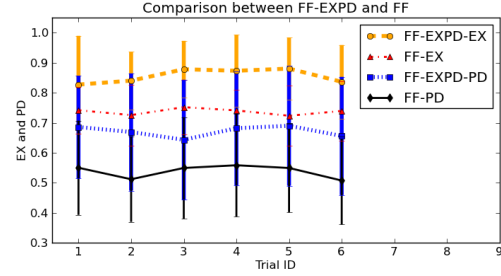
B. Evaluation with Physical Robots

In this section we evaluate our approach in a blocks world domain with a physical robot. It simulates a smart manufacturing environment where robots are working beside humans. Although the human and robot do not have direct interactions – the robot's goal is independent of the human's, generating explicable plan is still an important issue since it will help humans concentrate more on their own tasks. Here, we evaluate plans generated by the robot using FF-EXPD and a cost-optimal planner (OPT) in various scenarios and compare the plans with human subjects in terms of explicability.

1) Domain Description: In this domain, the robot's goal (which is known to the human) is to build a tower of a certain height using blocks on the table. The towers to be built have different heights in different problems. There are two types of blocks, light ones and heavy ones, which are indistinguishable externally but the robot can identify them based on the markers. Picking up the heavy blocks are more costly than the light blocks for the robot. Hence, the robot may sometimes choose seemingly more costly (i.e., longer) plans to build a tower from the human's perspective.

2) Experimental Setup: We generated a set of 23 problems in this domain in which towers of height 3 are to be built. The plans for these problems were manually generated and labeled as the training set. For 4 out of these 23 problems, the optimal plan is not the most explicable plan. To remove the influence of

grounding, we also generated permutations of each plan using different object names for these 23 problems, which resulted in a total of about 15000 training samples. We then generated a set of 8 testing problems for building towers of various heights (from $3-5$) to verify that our approach can generalize. Testing problems were generated only for cases where plans are more likely to be inexplicable. For each problem, we generated two plans, one using OPT and the other using FF-EXPD, and recorded the execution of these plans on the robot. We recruited 13 subjects on campus and each human subject was tasked with labeling two plans (generated by OPT and FF-EXPD respectively) for each of the 8 testing problems, using the recorded videos and following a process similar to that used in preparing training samples. After labeling each plan, we also asked the subject to provide a score ($1 - 10$ with 10 being the most explicable) to describe how comprehensible the plan was overall.

3) Results: In this evaluation, we only use one task label "*building tower*". For all testing problems, the labeling process results in 77.8% explicable actions (i.e., actions with a task label) for OPT and 97.3% explicable actions for FF-EXPD. The average explicability measures for FF-EXPD and OPT are 0.98 and 0.78, and the average scores are 9.65 and 6.92, respectively. We analyze the results using a paired T-test which shows a significant difference between FF-EXPD and OPT in terms of the explicability measures (using Eq. (7)) computed from the human labels and the overall scores ($p < 0.001$ for both). Furthermore, after normalizing the scores from the human subjects, the Cronbach's α value shows that the explicability measures and the scores are consistent for both FF-EXPD and OPT ($\alpha = 0.78, 0.67$, respectively). These results verify that: 1) our explicability measure does capture the human's interpretation of the robot plans and 2) our approach can generate plans that are more explicable to humans. In Fig. 7, we present the plans for a testing scenario. The left part of the figure shows the plan generated by OPT and the right part shows the plan generated by FF-EXPD. A video is also attached showing the different behaviors with the two planners in this scenario.

VI. CONCLUSION

While we are still far from having intelligent robots and agents working side-by-side of humans as teammates (rather than as tools), it becomes increasingly important to consider issues when such autonomous agents appear in our everyday life. These agents need to create and execute complex plans. In this paper, we introduced plan explicability and predictability for such agents so that they can synthesize plans that are more comprehensible to humans. To achieve this, they must consider not only their own models but also the human's interpretation of their models. To the best of our knowledge, this is the first attempt to model plan explicability and predictability for task planning which differs from previous work on human-aware planning. The proposed measures have a variety of applications (e.g., achieving fluent human-robot interaction and ensuring human safety). To compute these measures, we learn
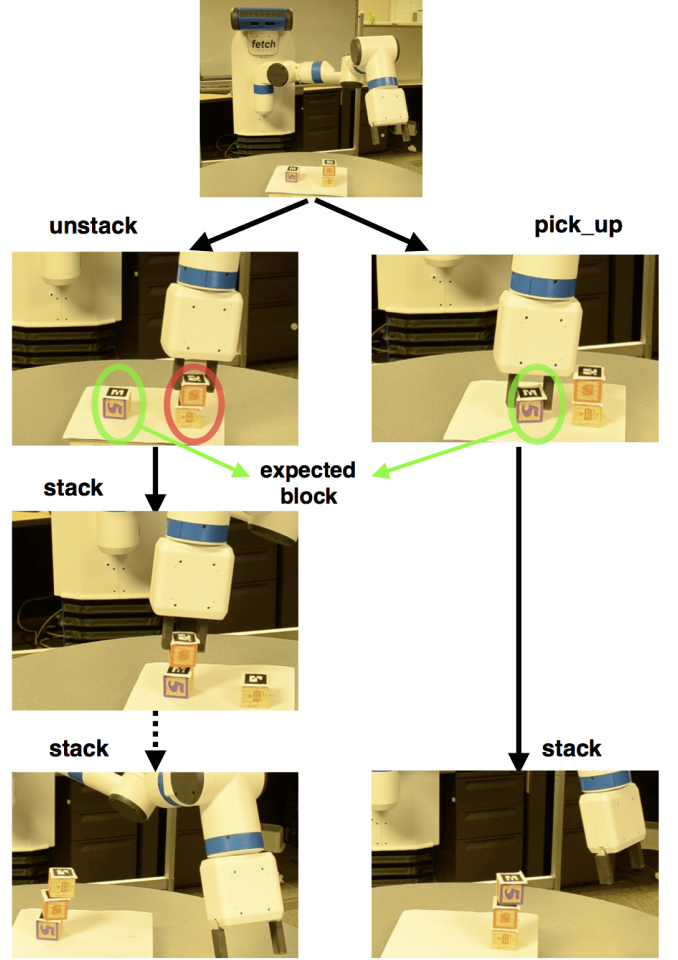

Fig. 7. Plan execution of two plans generated by OPT (left) and FF-EXPD (right) for one out of the 8 testing scenarios. The top figure shows the setup of this scenario where the goal is to build a tower of height 3. The block that is initially on the left side of the table is a heavy block. The optimal plan involves more actions with the light blocks (i.e., putting the two light blocks on top of the heavy one) while the explicable plan is more costly since it requires moving the heavy one.

the labeling scheme of humans for agent plans from training examples based on CRFs. We then use this learned model to label a new plan to compute its explicability and predictability. The proposed approach is evaluated on a synthetic domain and with human subjects using physical robots to show its effectiveness. A natural extension of our work is to consider human-robot teaming where there exists close interactions. Humans in our current settings are observers.

Finally, while we focus on the explicability and predictability measures for robot task planning, they also have many other interesting applications. For example, many defense applications use planning to create unpredictable and inexplicable plans, which can help deter or confuse enemies and are also useful for testing defenses against novel or unexpected attacks. These applications can be implemented using our approach by minimizing the θ and β measures instead of maximizing them.

REFERENCES

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, 2004.

[2] Tathagata Chakraborti, Gordon Briggs, Kartik Talamadupula, Yu Zhang, Matthias Scheutz, David Smith, and Subbarao Kambhampati. Planning for serendipity. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[3] Eugene Charniak and Robert P. Goldman. A bayesian model of plan recognition. *Artificial Intelligence*, 64(1): 53 – 79, 1993. ISSN 0004-3702.

[4] M. Cirillo, L. Karlsson, and A. Saffiotti. Human-aware task planning for mobile robots. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pages 1–7, June 2009.

[5] Gergely Csibra and György Gergely. Obsessed with goals?: Functions and mechanisms of teleological interpretation of actions in humans. *Acta psychologica*, 124 (1):60–78, 2007.

[6] Anca Dragan and Siddhartha Srinivasa. Generating legible motion. In *Proceedings of Robotics: Science and Systems*, Berlin, Germany, June 2013.

[7] Terrence W Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. A survey of socially interactive robots. *Robotics and Autonomous Systems*, 2003.

[8] Maria Fox and Derek Long. Pddl2.1: An extension to pddl for expressing temporal planning domains. *J. Artif. Int. Res.*, 20(1), December 2003.

[9] Moritz Gbelbecker, Thomas Keller, Patrick Eyerich, Michael Brenner, and Bernhard Nebel. Coming up with good excuses: What to do when no plan can be found. In *International Conference on Automated Planning and Scheduling*, 2010.

[10] Marc Hanheide, Moritz Gbelbecker, Graham S. Horn, Andrzej Pronobis, Kristoffer Sj, Alper Aydemir, Patric Jensfelt, Charles Gretton, Richard Dearden, Miroslav Janicek, Hendrik Zender, Geert-Jan Kruijff, Nick Hawes, and Jeremy L. Wyatt. Robot task planning and explanation in open and uncertain worlds. *Artificial Intelligence*, 2015. ISSN 0004-3702.

[11] Guy Hoffman and Cynthia Breazeal. Cost-based anticipatory action selection for human–robot fluency. *Robotics, IEEE Transactions on*, 23(5):952–961, 2007.

[12] Guy Hoffman and Cynthia Breazeal. Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 1–8. ACM, 2007.

[13] Jörg Hoffmann and Bernhard Nebel. The ff planning system: Fast plan generation through heuristic search. *J. Artif. Int. Res.*, 14(1):253–302, May 2001. ISSN 1076-9757.

[14] Henry A. Kautz and James F. Allen. Generalized Plan Recognition. In *National Conference on Artificial Intelligence*, pages 32–37, 1986.

[15] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, 2001. ISBN 1-55860-778-1.

[16] Steven James Levine and Brian Charles Williams. Concurrent plan recognition and execution for human-robot teams. In *Twenty-Fourth International Conference on Automated Planning and Scheduling*, 2014.

[17] Tom Murray. Authoring Intelligent Tutoring Systems: An analysis of the state of the art.

[18] Vignesh Narayanan, Yu Zhang, Nathaniel Mendoza, and Subbarao Kambhampati. Automated planning for peer-to-peer teaming and its evaluation in remote human-robot interaction. In *ACM/IEEE International Conference on Human Robot Interaction (HRI)*, 2015.

[19] Fuchun Peng and Andrew McCallum. Information extraction from research papers using conditional random fields. *Information Processing & Management*, 42(4): 963 – 979, 2006. ISSN 0306-4573.

[20] Miquel Ramírez and Hector Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, 2010.

[21] Julie Shah, James Wiken, Brian Williams, and Cynthia Breazeal. Improved human-robot team performance using chaski, a human-inspired plan execution system. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 29–36. ACM, 2011.

[22] E.A Sisbot, L.F. Marin-Urias, R. Alami, and T. Simeon. A human aware mobile robot motion planner. *Robotics, IEEE Transactions on*, 23(5):874–883, Oct 2007.

[23] Kartik Talamadupula, Gordon Briggs, Tathagata Chakraborti, Matthias Scheutz, and Subbarao Kambhampati. Coordination in human-robot teams using mental modeling and plan recognition. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2957–2962, Sept 2014.

[24] Robin R. Vallacher and Daniel M. Wegner. What do people think they're doing? action identification and human behavior. *Psychological Review*, 94(1):3–15, 1987.

[25] Yu Zhang, Vignesh Narayanan, Tathagata Chakraborty, and Subbarao Kambhampati. A human factors analysis of proactive assistance in human-robot teaming. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[26] Yu Zhang, Sarath Sreedharan, and Subbarao Kambhampati. Capability models and their applications in planning. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, 2015.