Explicable Planning as Minimizing Distance from Expected Behavior

Anagha Kulkarni, Yantian Zha, Tathagata Chakraborti,
Satya Gautam Vadlamudi, Yu Zhang and Subbarao Kambhampati
School of Computing, Informatics, and Decision Systems Engineering
Arizona State University, Tempe, AZ 85281 USA
{akulka16, yzha3, tchakra2, yzhan442, rao} @ asu.edu and gautam.vadlamudi@creditvidya.com

Abstract

In order to have effective human-AI collaboration, it is necessary to address how the AI agent's behavior is being perceived by the humans-in-the-loop. When the agent's task plans are generated without such considerations, they may often demonstrate *inexplicable behavior* from the human's point of view. This problem may arise due to the human's partial or inaccurate understanding of the agent's planning model. This may have serious implications from increased cognitive load to more serious concerns of safety around a physical agent. In this paper, we address this issue by modeling plan explicability as a function of the distance between a plan that agent makes and the plan that human expects it to make. We learn a regression model for mapping the plan distances to explicability scores of plans and develop an anytime search algorithm that can use this model as a heuristic to come up with progressively explicable plans. We evaluate the effectiveness of our approach in a simulated autonomous car domain and a physical robot domain.

1 Introduction

There is growing interest in applications that involve human-robot collaboration. An important challenge in such human-in-the-loop scenarios is to ensure that agent's behavior is not just optimal with respect to its own model, but is also explicable and comprehensible to the humans in the loop. Without it, the robot runs the risk of increasing the cognitive load of humans which can result in reduced productivity, safety, and trust (Fan et al. 2008). This mismatch between the robot's plans and the human's expectations of the robot's plans may arise because of the difference in the actual robot model $\mathcal{M}^{\mathcal{R}}$, and the *human's expectation of the robot model* $\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}$. For example, consider a scenario with an autonomous car switching lanes on a highway. The autonomous car, in order to switch the lane, may make sharp and calculated moves, as opposed to gradually moving towards the other lane. These moves may well be optimal for the car and backed by the car's superior sensing and steering capabilities. Nevertheless, a human passenger sitting inside may perceive this as dangerous and reckless behavior, in as much as they might be ascribing the car the sort of driving abilities they themselves have.

In order to avoid inexplicable behavior, the robot has to take the human mental model into account and compute the plans that align with this model. As long as the robot's behavior is aligned with the human mental model, the human can make sense of it. Therefore, the objective of explicable planning is to generate robot plans that not only minimize the cost of the plan, but also the distance between $\mathcal{M}^{\mathcal{R}}$ and $\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}$. Of course, an immediate question is, if $\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}$ is available to the robot, why is $\mathcal{M}^{\mathcal{R}}$ required in the plan generation process at all? We note that this is necessary since the human mental model might entail plans that are not even feasible for the robot or are prohibitively expensive, and can thus at best serve as a guide, and not an oracle, to the explicable plan generation process. In such scenarios, the robot can choose to provide explanations about its inexplicable behavior which help to communicate the model discrepancies to the perplexed human (Chakraborti et al. 2017; Langley 2016).

An important consideration in the computation of explicable behavior is access to the human mental model, $\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}$. In many domains, such as in household or factory scenarios), there is generally a clear expectation of how a task should be

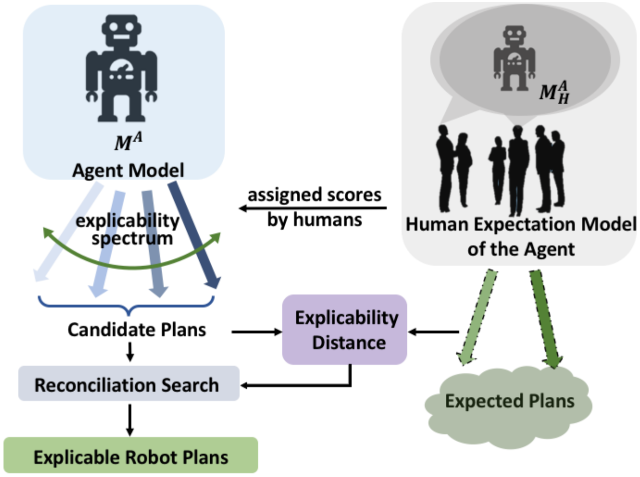


Figure 1: Schematic diagram of the first setting: Here a regression model called explicability distance is learned to fit plan scores assigned by humans to plan distances between robot's and human's expected plans. This gives heuristic for computing explicable plans.

performed. In such cases, $\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}$ can be constructed following the norms or protocols that are relevant to that domain. Most deployed products make use of inbuilt models of user expectations in some form or the other. In the first setting, we hypothesize that the plan distances (Srivastava et al. 2007; Nguyen et al. 2012) can quantify the distance between the robot plan $\pi_{\mathcal{M}^{\mathcal{R}}}$ and the expected plans $\pi_{\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}}$ from $\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}$. The domain modeler constructs $\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}$, which is then used to generate expected plans. Then we compute the distance between plans from $\mathcal{M}^{\mathcal{R}}$ and $\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}$. The test subjects provide explicability assessments (scores reflecting explicability) for robot plans. These scores are mapped to the precomputed distances to derive an explicability distance. Then the plan generation process uses explicability distance as a heuristic to guide the search. This approach is illustrated in Figure 1.

In the following sections, we provide detailed methodologies to compute explicable plans in each of the settings. We also present results on each setting using physical robot-based domains.

2 Explicable Planning Problem

2.1 Classical Planning

A classical planning problem can be defined as a tuple $\mathcal{P} = \langle \mathcal{M}, \mathcal{I}, \mathcal{G} \rangle$, where $\mathcal{M} = \langle \mathcal{F}, \mathcal{A} \rangle$ is the domain model (that consists of a finite set $\mathcal{F}$ of fluents that define the state of the world and a set of operators or actions $\mathcal{A}$), and $\mathcal{I} \subseteq \mathcal{F}$ and $\mathcal{G} \subseteq \mathcal{F}$ are the initial and goal states of the problem respectively. Each action $a \in \mathcal{A}$ is a tuple of the form $\langle pre(a), eff(a), c(a) \rangle$ where $c(a)$ denotes the cost of an action, $pre(a) \subseteq \mathcal{F}$ is the set of preconditions for the action a and $eff(a) \subseteq \mathcal{F}$ is the set of the effects. The solution to the planning problem is a *plan* or a sequence of actions $\pi = \langle a_1, a_2, \ldots, a_n \rangle$ such that starting from the initial state, sequentially executing the actions lands the agent in the goal state, i.e. $\Gamma_{\mathcal{M}}(\mathcal{I}, \pi) \models \mathcal{G}$ where $\Gamma_{\mathcal{M}}(\cdot)$ is the transition function defined for the domain. The cost of the plan, denoted as $c(\pi)$, is given by the summation of the cost of all the actions in the plan π, $c(\pi) = \sum_{a_i \in \pi} c(a_i)$.

2.2 Problem Definition

The problem of explicable planning arises when the robot plan, $\pi_{\mathcal{M}^{\mathcal{R}}}$, deviates from the human's expectation of that plan, $\pi_{\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}}$. Here $\pi_{\mathcal{M}^{\mathcal{R}}}$ is the robot's plan solution to the planning problem, $\mathcal{P}^{\mathcal{R}} = \langle \mathcal{M}^{\mathcal{R}}, \mathcal{I}^{\mathcal{R}}, \mathcal{G}^{\mathcal{R}} \rangle$, where $\mathcal{M}^{\mathcal{R}} = \langle \mathcal{F}^{\mathcal{R}}, \mathcal{A}^{\mathcal{R}} \rangle$; whereas, $\pi_{\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}}$ is the plan solution considering the human mental model, such that, $\mathcal{P}^{\mathcal{R}}_{\mathcal{H}} = \langle \mathcal{M}^{\mathcal{R}}_{\mathcal{H}}, \mathcal{I}^{\mathcal{R}}, \mathcal{G}^{\mathcal{R}} \rangle$, where $\mathcal{M}^{\mathcal{R}}_{\mathcal{H}} = \langle \mathcal{F}^{\mathcal{R}}, \mathcal{A}^{\mathcal{R}}_{\mathcal{H}} \rangle$. The differences in the human mental model can lead to different plan solutions. We now formally define the explicable planning problem and the solution to it.

Definition 1. *The **explicable planning problem** is defined as a tuple $\mathcal{P}_{EPP} = \langle \mathcal{M}^{\mathcal{R}}, \mathcal{M}^{\mathcal{R}}_{\mathcal{H}}, \mathcal{I}^{\mathcal{R}}, \mathcal{G}^{\mathcal{R}} \rangle$, where,*

- *$\mathcal{M}^{\mathcal{R}} = \langle \mathcal{F}^{\mathcal{R}}, \mathcal{A}^{\mathcal{R}} \rangle$ is the robot's domain model where $\mathcal{F}^{\mathcal{R}}$ and $\mathcal{A}^{\mathcal{R}}$ represent the actual set of fluents and actions available to the robot*
- *$\mathcal{M}^{\mathcal{R}}_{\mathcal{H}} = \langle \mathcal{F}^{\mathcal{R}}, \mathcal{A}^{\mathcal{R}}_{\mathcal{H}} \rangle$ is the human's mental model of the robot model where $\mathcal{F}^{\mathcal{R}}$ and $\mathcal{A}^{\mathcal{R}}_{\mathcal{H}}$ represent the set of fluents and actions that the human thinks are available to the robot*
- *$\mathcal{I}^{\mathcal{R}}$ is the initial state of the robot*
- *$\mathcal{G}^{\mathcal{R}}$ is the goal state of the robot*

Here $\mathcal{A}^{\mathcal{R}}$ and $\mathcal{A}^{\mathcal{R}}_{\mathcal{H}}$ represent that the action names, preconditions, effects and costs of the actions can be different. The initial state and the goal state are known to the human-in-the-loop. In order to compute the difference between the robot plan and the plan expected by a human, we need an evaluation function that computes the distance in terms of various aspects of the plan, like the action sequence, state sequence, etc. Let's denote such an evaluation function as δ^*. The solution to an explicable planning problem is an explicable plan that achieves the goal and minimizes the model differences while also minimizing the cost of the plan. We now formally define it as follows:

Definition 2. *(Zhang et al. 2017) A plan is an **explicable plan**, $\pi^*_{\mathcal{M}^{\mathcal{R}}}$, starting at $\mathcal{I}^{\mathcal{R}}$ that achieves the goal $\mathcal{G}^{\mathcal{R}}$, such that, $\underset{\pi_{\mathcal{M}^{\mathcal{R}}}}{\arg\min}\, c(\pi_{\mathcal{M}^{\mathcal{R}}}) + \delta^*(\pi_{\mathcal{M}^{\mathcal{R}}}, \pi_{\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}})$, where δ^* is an evaluation function.*

Since the evaluation function is not directly available to us, we learn an approximation of it using a combination of three plan distances measures. In the following section, we discuss our approach for learning the evaluation function and computing explicable plans.

3 Proposed Methodology

In this section, we quantify the explicability of the robot plans in terms of the *plan distance* between the robot plan $\pi_{\mathcal{M}^{\mathcal{R}}}$ and candidate expected plans $\pi_{\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}}$ from $\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}$. The outline of our approach is illustrated in Figure 1. Our first aim is to compute the plan distances between the robot plans and the expected plans using plan distance measures. Then we map the human explicability assessments (scores reflecting explicability) of candidate robot plans to the plan distance measures in form of regression model called explicability distance. The synthesis of explicable plans is achieved by modifying the `Fast-Downward` (Helmert 2006) planner to incorporate an anytime search with explicability distance as the heuristic. This process results in incrementally more explicable plans.

3.1 Background on Plan Distance Measures

We now briefly look at the three plan distance measures introduced and refined in (Srivastava et al. 2007; Nguyen et al. 2012), namely action, causal link and state sequence distances.

Action distance We denote the set of unique actions in a plan π as $A(\pi) = \{a \mid a \in \pi\}$. Given the action sets $A(\pi_{\mathcal{M}^{\mathcal{R}}})$ and $A(\pi^*_{\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}})$ of two plans $\pi_{\mathcal{M}^{\mathcal{R}}}$ and $\pi^*_{\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}}$ respectively, the action distance is, $\delta_A(\pi_{\mathcal{M}^{\mathcal{R}}}, \pi^*_{\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}}) = 1 -$

$\frac{|A(\pi_{\mathcal{M}^\mathcal{R}})\cap A(\pi^*_{\mathcal{M}^\mathcal{R}_\mathcal{H}})|}{|A(\pi_{\mathcal{M}^\mathcal{R}})\cup A(\pi^*_{\mathcal{M}^\mathcal{R}_\mathcal{H}})|}$. Here, two plans are similar (and hence their distance measure is smaller) if they contain same actions. Note that it does not consider the ordering of actions.

Causal link distance A causal link represents a tuple of the form $\langle a_i, p_i, a_{i+1}\rangle$, where p_i is a predicate variable that is produced as an effect of action a_i and used as a precondition for the next action a_{i+1}. The causal link distance measure is represented using the causal link sets $Cl(\pi_{\mathcal{M}^\mathcal{R}})$ and $Cl(\pi^*_{\mathcal{M}^\mathcal{R}_\mathcal{H}})$, $\delta_C(\pi_{\mathcal{M}^\mathcal{R}}, \pi^*_{\mathcal{M}^\mathcal{R}_\mathcal{H}}) = 1 - \frac{|Cl(\pi_{\mathcal{M}^\mathcal{R}})\cap Cl(\pi^*_{\mathcal{M}^\mathcal{R}_\mathcal{H}})|}{|Cl(\pi_{\mathcal{M}^\mathcal{R}})\cup Cl(\pi^*_{\mathcal{M}^\mathcal{R}_\mathcal{H}})|}$.

State sequence distance This distance measure finds the difference between sequences of the states. Given two state sequences $(s_0^R, \ldots, s_n^R)$ and $(s_0^H, \ldots, s_{n'}^H)$ for $\pi_{\mathcal{M}^\mathcal{R}}$ and $\pi^*_{\mathcal{M}^\mathcal{R}_\mathcal{H}}$ respectively, where $n \geq n'$ are the lengths of the plans, the state sequence distance is, $\delta_S(\pi_{\mathcal{M}^\mathcal{R}}, \pi^*_{\mathcal{M}^\mathcal{R}_\mathcal{H}}) = \frac{1}{n}\Big[\sum_{k=1}^{n'} d(s_k^R, s_k^H) + n - n'\Big]$, where $d(s_k^R, s_k^H) = 1 - \frac{|s_k^R \cap s_k^H|}{|s_k^R \cup s_k^H|}$ represents the distance between two states (where s_k^R is overloaded to denote the set of predicate variables in state s_k^R). The first term measures the normalized difference between states up to the end of the shortest plan, while the second term, in the absence of a state to compare to, assigns maximum difference possible.

3.2 Explicability Distance

We start with a general formulation for capturing a measure of explicability of the robot's plans using plan distances. A set of robot plans are scored by humans such that each action that follows the human's expectation in the context of the plan is scored 1 if explicable (0 otherwise). The plan score is then computed as the ratio of the number of explicable actions to the total plan length.

Definition 3. *A set of plans is a* ***set of expected plans****, $\mathcal{E}(\mathcal{P}^\mathcal{R}_\mathcal{H})$, for the planning problem $\mathcal{P}^\mathcal{R}_\mathcal{H} = \langle \mathcal{M}^\mathcal{R}_\mathcal{H}, \mathcal{I}^\mathcal{R}, \mathcal{G}^\mathcal{R}\rangle$, is a set of optimal cost plans that solve $\mathcal{P}^\mathcal{R}_\mathcal{H}$, $\mathcal{E}(\mathcal{P}^\mathcal{R}_\mathcal{H}) = \{\pi^{(i)}_{\mathcal{M}^\mathcal{R}_\mathcal{H}} | i = 1, \ldots, n\}$.*

This set of expected plans consists of the plan solutions that the human expects the robot to compute. But these plans are not necessarily feasible in the robot model, $\mathcal{M}^\mathcal{R}$.

Definition 4. *A* ***composite distance****, δ_{exp} is a distance between pair of two plans $\langle \pi_{\mathcal{M}^\mathcal{R}}, \pi_{\mathcal{M}^\mathcal{R}_\mathcal{H}}\rangle$, such that, $\delta_{exp}(\pi_{\mathcal{M}^\mathcal{R}}, \pi_{\mathcal{M}^\mathcal{R}_\mathcal{H}}) = ||\delta_A(\pi_{\mathcal{M}^\mathcal{R}}, \pi_{\mathcal{M}^\mathcal{R}_\mathcal{H}}) + \delta_C(\pi_{\mathcal{M}^\mathcal{R}}, \pi_{\mathcal{M}^\mathcal{R}_\mathcal{H}}) + \delta_S(\pi_{\mathcal{M}^\mathcal{R}}, \pi_{\mathcal{M}^\mathcal{R}_\mathcal{H}})||^2$.*

Among the set of expected plans, we compute a distance minimizing plan as follows:

Definition 5. *A* ***distance minimizing plan****, $\pi^*_{\mathcal{M}^\mathcal{R}_\mathcal{H}}$, is a plan in $\mathcal{E}(\mathcal{P}^\mathcal{R}_\mathcal{H})$, such that for a robot plan, $\pi_{\mathcal{M}^\mathcal{R}}$, the composite distance is minimized, $\pi^*_{\mathcal{M}^\mathcal{R}_\mathcal{H}} = \{\pi_{\mathcal{M}^\mathcal{R}_\mathcal{H}} | \operatorname*{arg\,min}_{\pi_{\mathcal{M}^\mathcal{R}_\mathcal{H}}} \delta_{exp}(\pi_{\mathcal{M}^\mathcal{R}}, \pi_{\mathcal{M}^\mathcal{R}_\mathcal{H}})\}$.*

Algorithm 1 Reconciliation Search

```
Input: $\mathcal{P}_{\mathcal{EPP}} = \langle \mathcal{M}^\mathcal{R}, \mathcal{M}^\mathcal{R}_\mathcal{H}, \mathcal{I}^\mathcal{R}, \mathcal{G}^\mathcal{R}\rangle$, $max\_cost$, and explicability distance function Exp(. , .)
Output: $\mathcal{E}_{\mathcal{EPP}}$
$\mathcal{E}_{\mathcal{EPP}} \leftarrow \emptyset$                          ▷ Explicable plan set
open ← $\emptyset$                               ▷ Open list
closed ← $\emptyset$                             ▷ Closed list
open.insert($\mathcal{I}, 0, \inf$)
while open $\neq \emptyset$ do
    n ← open.remove()                   ▷ Node with highest $h(\cdot)$
    if n $\models \mathcal{G}$ then
        $\mathcal{E}_{\mathcal{EPP}}$.insert($\pi$ s.t. $\Gamma_{\mathcal{M}^\mathcal{R}}(\mathcal{I}, \pi) \models n$)
    end if
    closed.insert(n)
    for each v $\in$ successors(n) do
        if v $\notin$ closed then
            if g(n) + cost(n, v) $\leq$ max_cost then
                open.insert(v, g(n) + cost(n, v), h(v))
            end if
        else
            if h(n) < h(v) then
                closed.remove(v)
                open.insert(v, g(n) + cost(n, v), h(v))
            end if
        end if
    end for
end while
return $\mathcal{E}_{\mathcal{EPP}}$
```

Definition 6. *An* ***explicability feature vector****, Δ, is a three-dimensional vector, which is given with respect to a distance minimizing plan pair, $\langle \pi_{\mathcal{M}^\mathcal{R}}, \pi^*_{\mathcal{M}^\mathcal{R}_\mathcal{H}}\rangle$, such that, $\Delta = \langle \delta_A(\pi_{\mathcal{M}^\mathcal{R}}, \pi^*_{\mathcal{M}^\mathcal{R}_\mathcal{H}}), \delta_C(\pi_{\mathcal{M}^\mathcal{R}}, \pi^*_{\mathcal{M}^\mathcal{R}_\mathcal{H}}), \delta_S(\pi_{\mathcal{M}^\mathcal{R}}, \pi^*_{\mathcal{M}^\mathcal{R}_\mathcal{H}})\rangle^T$.*

We approximate the evaluation function, δ^*, using a combination of the three plan distance measures as follows:

Definition 7. *The* ***explicability distance function*** *Exp($\pi_{\mathcal{M}^\mathcal{R}}$ / $\pi^*_{\mathcal{M}^\mathcal{R}_\mathcal{H}}$), is a regression function, f, that fits the three plan distances to the total plan scores, with b as the parameter vector, and Δ as the explicability feature vector, such that, Exp($\pi_{\mathcal{M}^\mathcal{R}}$ / $\pi^*_{\mathcal{M}^\mathcal{R}_\mathcal{H}}$) $\approx$ f(Δ, b).*

A regression model is trained to learn the explicability assessment (total plan scores) of the users by mapping this assessment to the explicability feature vector which consists of plan distances for corresponding plans.

3.3 Plan Generation

In this section, we present the details of our plan generation phase. We use the learned explicability distance function as a heuristic to guide our search towards explicable plans.

Reconciliation Search

Given an explicable planning problem $\mathcal{P}_{\mathcal{EPP}} = \langle \mathcal{M}^\mathcal{R}, \mathcal{M}^\mathcal{R}_\mathcal{H}, \mathcal{I}^\mathcal{R}, \mathcal{G}^\mathcal{R}\rangle$, the set of explicable plans in $\mathcal{M}^\mathcal{R}$ are defined as, $\mathcal{E}_{\mathcal{EPP}} = \{\pi^{*(i)}_{\mathcal{M}^\mathcal{R}} | i = 1, \ldots, m\}$, such that, each $\pi^{*(i)}_{\mathcal{M}^{\mathcal{R}}}$ is found by performing reconciliation search. Here each $\pi^{*(i)}_{\mathcal{M}^{\mathcal{R}}}$ may have different degree of explicability.

Non-Monotonicity Since plan score is the fraction of explicable actions in a plan, it exhibits non-monotonicity. As a partial plan grows, a new action may contribute either positively or negatively to the plan score, thus making the explicability distance function non-monotonic. Consider that the goal of an autonomous car is to park itself in a parking spot on its left side. The car takes the left turn, parks and turns on its left indicator. Here the turning on of the left tail light after having parked is an inexplicable action. The first two actions are explicable to the human drivers and contribute positively to the explicability score of the plan but the last action has a negative impact and decreases the score.

Due to non-monotonic nature of explicability distance, we cannot stop the search process after finding the first solution. Consider the following: if e_1 is explicability distance of the first plan, then a node may exist in open list (set of unexpanded nodes) whose explicability distance is less than e_1, which when expanded may result in a solution plan with explicability distance higher than e_1. A greedy method that expands a node with the highest explicability score of the corresponding partial plan at each step is not guaranteed to find an optimal explicable plan (one of the plans with the highest explicability score) as its first solution. Therefore, to handle the non-monotonic nature, we present a cost-bounded anytime greedy search algorithm called *reconciliation search* that generates all the valid loopless candidate plans up to a given cost bound, and then progressively searches for plans with better explicability scores. The value of the heuristic *h(v)* in a particular state *v* encountered during search is based entirely on the explicability distance of the agent plan prefix $\pi_{\mathcal{M}^{\mathcal{R}}}$ up to that state, *h(v)* $= Exp(\pi_{\mathcal{M}^{\mathcal{R}}}/\pi'_{\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}})$ s.t. $\Gamma_{\mathcal{M}^{\mathcal{R}}}(\mathcal{I}, \pi_{\mathcal{M}^{\mathcal{R}}}) \models v$and $\Gamma_{\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}}(\mathcal{I}, \pi'_{\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}}) \models v$.

We implement this search in the `Fast-Downward` planner. The approach is described in detail in Algorithm 1. At each iteration of the algorithm, the plan prefix of the agent model is compared with the explicable trace $\pi'_{\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}}$ (these are the plans generated using $\mathcal{M}^{\mathcal{R}}_{\mathcal{H}}$ up to the current state in the search process) for the given problem. Using the computed distances, the explicability score for the candidate agent plans is predicted. The search algorithm then makes a locally optimal choice of states. We do not stop the search after generating the first solution, but instead, continue to find all the valid loopless candidate solutions within the given cost bound or until the state space is completely explored.

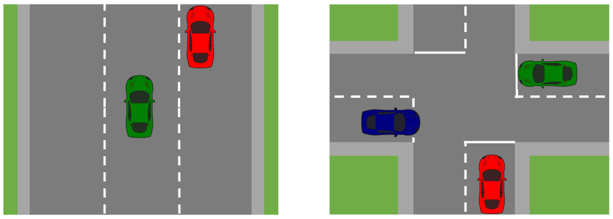
4 Experimental Analysis

Here we present the evaluation of our system in simulated autonomous car domain and physical robot delivery domain.

4.1 Autonomous Car Domain

No.	Questions	Yes	No
1	Autonomous car should always maintain itself in the center of the lane unless it is changing lanes	16	4
2	The car should automatically turn signal lights on before lane change.	20	0
3	The car should automatically turn signal lights on when the car starts swerving away from the center of the lane.	9	11
4	At a four-way stop, it should automatically provide turn signal lights when it is taking left or right turns.	18	2
5	It should slow down automatically when it is swerving off the center of the lane.	15	5
6	It should slow down automatically when there is an obstacle (pedestrian or parked vehicle) ahead in the lane.	17	3
7	Check one: When an emergency vehicle is parked on the rightmost lane, it should (1) automatically follow the move over maneuver, (2) whenever possible follow the move over maneuver.	9	11
8	Check one: At a four-way stop, it should (1) wait for the intersection to be clear and to be extra safe. (2) wait for the other cars unless it is its turn to cross over.	15	5

Table 1: The questionnaire used in the human study for Car Domain, and the tally of answers given by participants. For the last two questions, the participants were asked to choose one of the two options, and the "yes" tally corresponds to the first answer, "no" to the second.



(a) (b)

Figure 2: Simulated Autonomous Car Domain. Here only the red car is autonomous. (a) It is having difficulty merging to the middle lane and is confusing the human driver with its signals. (b) It is waiting at a 4-way stop even though it is its turn to cross.

Domain model There are a lot of social norms followed by human drivers which are usually above explicit laws. This can include normative behavior while changing lanes or during turn-taking at intersections. As such this car domain has emerged as a vibrant testbed for research (Sadigh et al. 2016; Kyle Hollins Wray 2017) in the HRI/AI community in recent times. In this work, we explore the topic of explicable behavior of an autonomous car in this domain, especially as it related to mental modeling of the humans in the loop.

Algorithm	Car R^2 %	Delivery R^2 %
Ridge Regression	53.66	31.42
AdaBoost Regression	61.31	66.99
Decision Tree Regression	74.79	39.61
Random Forest Regression	90.45	75.29

Table 2: Accuracy for car and delivery domain.

In our autonomous car domain (modeled in PDDL), the autonomous car model $\mathcal{M}^{\mathcal{A}}$ consists of lane and car objects as shown in Figure 2. The red car is the autonomous car in the experiments and all other cars are assumed to have human drivers. The car objects are associated with predicates defining the location of the car on a lane segment, status of left and right turn lights, whether the car is within the speed limit, the presence of a parked police car, and so on. The actions possible in the domain are with respect to the autonomous car. These actions are *Accelerate*, *Decelerate*, *LeftSqueeze*, *RightSqueeze*, *LeftLight* $\{$*On, Off*$\}$, *RightLight* $\{$*On, Off*$\}$, *SlowDown* and *WaitAtStopSign*. To change a lane, three consecutive actions of $\{$*Left, Right*$\}$ *Squeeze* are required.

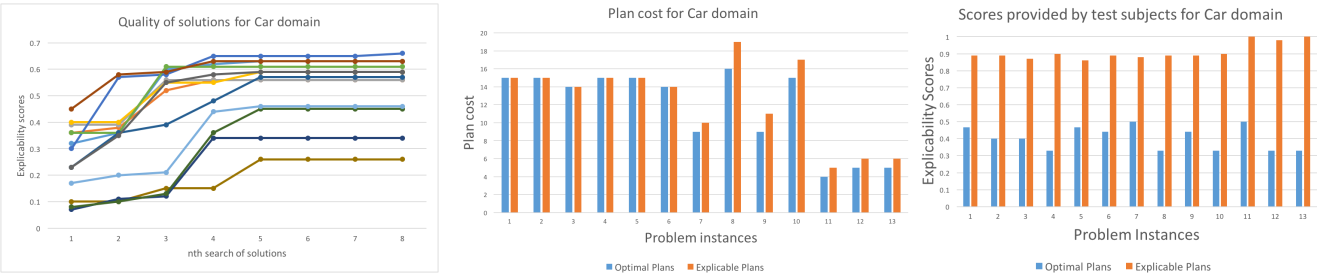
From $\mathcal{M}^{\mathcal{A}}$ we generated a total of 40 plans (consisting of both explicable and inexplicable behaviors) for 16 different planning problems. These plans were assessed by 20 human subjects, with each subject evaluating 8 plans (apart from 1 subject who evaluated 7 plans). Also, each plan was evaluated by 4 different subjects. The overall number of training samples was 159. The test subjects were required to have a state driving license. The subjects were provided with the initial state and goal of the car. After seeing the simulation the plan, they had to record whether they found each action explicable or not. The assessment had two parts: one part involved scoring each autonomous car action with 1, if explicable, and 0 otherwise (plan score was calculated as the fraction of actions in the plan that were labeled as explicable); the other part involved answering a questionnaire on the preconditions, effects of the agent actions. It consisted of 8 questions with *yes/no* answers. The questions used for constructing the domain and the corresponding answers are provided in Table 1. For each question, the answers with votes were used by the domain modeler to construct $\mathcal{M}^{\mathcal{A}}_{\mathcal{H}}$. The questions with divided opinions (3 and 7) were not included in the models as some found that behavior explicable while some others did not. The $\mathcal{M}^{\mathcal{A}}_{\mathcal{H}}$ domain consists of the same state predicates but ended up with different action definitions with respect to preconditions, effects, and action-costs.

Following are two examples of how the feedback was interpreted by the domain modeler. For question 1, the majority of answers agreed with the statement. Therefore for actions Accelerate and SlowDown, additional preconditions like *(not (squeezingLeft ?x)), (not (squeezingLeft2 ?x)), (not (squeezingRight ?x)), (not (squeezingRight2 ?x))* were added, where x stands for car. For question 8, since the answers agreed with choice 2, actions *waitAtStopSign1*, *waitAtStopSign2*, *waitAtStopSign3* were replaced by a new general action *waitAtStopSign*. This action removed predicates *waiting1, waiting2, waiting3* from the action definition. Also, actions *atStopSignAccelerate*, *atStopSignLeft*, *atStopSignRight* were changed to remove the precondition *waiting3* (these actions thus had two definitions in $\mathcal{M}^{\mathcal{A}}$ to allow for explicable behavior, one with higher cost).

Defining the explicability distance For the training problems, explicable plans were generated using the model $\mathcal{M}^{\mathcal{A}}_{\mathcal{H}}$. Since some actions names were not common to both the domains, an explicit mapping was defined between the actions over the two domains. This mapping was done in order to support plan distance operations performed between plans in the two domains (for the plan distances to be used effectively common action names are required).

As noted in Definition 6, features of the regression model are the three plan distances and the target is the score associated with the plans. We tune the hyperparameters by performing grid search over parameters like the number of trees, depth of tree and the minimum number of nodes to perform sample split. The results for different learning models are as shown in Table 2. We tried several ensemble learning algorithms to improve the accuracy of our model, out of which random forest regression gave the best performance. Random forests allow selection of a random subset of features while splitting the decision node. We evaluate the goodness of fit of the model, using the coefficient of determination or R^2. After training process, the new regression model was found to have 0.9045 R^2 value. That is to say, 90% of the variations in the features can be explained by our model. Our model predicts the explicability distance between the agent plans and human mental model plans, with a high accuracy.

Evaluation We evaluated our approach on 13 different planning problems. We ran the algorithm with a high cost bound, in order to cover the most explicable candidate plans for all the problems. The Figure 3a reports the explicability scores of the first 8 solutions generated by our algorithm for 13 test problems. From this graph, we note that the reconciliation search is able to develop plans with incrementally better explicability scores. These are the internal explicability scores (produced by the explicability distance function). From Figure 3b, we see for the last 7 problems, our planner generated explicable plans with a cost higher than that of optimal plans; a planner insensitive to explicability would not have been able to find these expensive but explicable plans. This additional cost can be seen as the price the agent pays to make its behavior explicable to the human. For the first 6 problems, even though the cost is same as that of the optimal plans, a planner insensitive to explicability cannot guarantee the generation of explicable plans. From Figure 3c, the test subjects provided higher explicability scores for the explicable plans than the optimal plans for all 13 problems. The plan scores given by 10 test subjects were computed as the ratio of explicable actions in the plans to the total plan length. Testing phase protocol was same as that



(a) (b) (c)

Figure 3: Car domain: (a) The graph shows how the search process finds plans with incrementally better explicable scores. Each color line represents one of the 13 different test problems. The markers on the lines represent a plan solution for that problem. The y-axis gives the explicability scores of the plans and the x-axis gives the solution number. The optimal plans generated using the `Fast-Downward` planner were compared for (b) plan costs (c) explicability scores provided by test subjects.

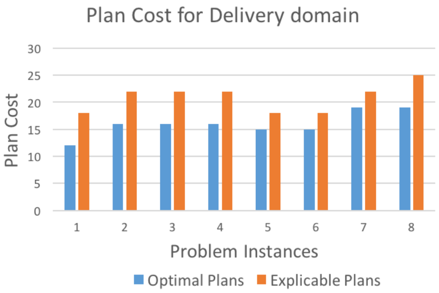


Figure 4: For delivery domain test problem instances, the optimal and explicable plans were compared for their plan costs.

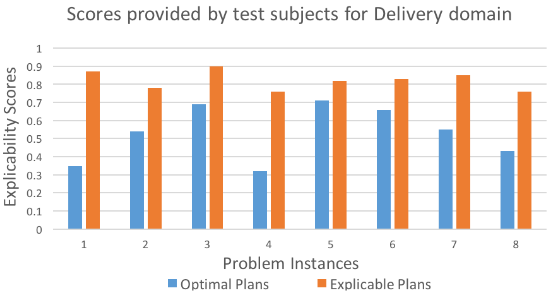


Figure 5: For delivery domain test problem instances, the optimal and explicable plans were compared for explicability scores provided by test subjects.

of the training phase, except for the questionnaire. The plan scores were averaged over the number of test subjects. In this domain, an inexplicable plan for changing lanes from *l2* to *l1* can be *LeftSqueeze-l2-l1, LeftSqueeze-l2-l1, LeftSqueeze-l2-l1, LeftLightOn* whereas an explicable plan would be *LeftLightOn, LeftSqueeze-l2-l1, LeftSqueeze-l2-l1, LeftSqueeze-l2-l1*. The ordering of *LeftLightOn* action decides whether the plan is explicable.

4.2 Robot based Delivery Domain

This domain is designed to demonstrate inexplicable behaviors of a delivery robot. The robot can deliver parcels/electronic devices and serve beverages to the humans. It has access to three regions namely kitchen, reception and employee desk. For the evaluation, we used a Fetch robot, which has a single arm with parallel grippers for grasping objects. It delivers beverages, parcels, and devices using a tray. Whenever the robot carries the beverage cup there is some risk that the cup may tip over and spill the contents all over the electronic items on the tray. Here the robot has to learn the context of carrying devices and beverages separately even if it results in an expensive plan (in terms of cost or time) for it. A sample plan in this domain with explicable and inexplicable plans is illustrated in Figure 6. Here, in the inexplicable version, the robot delivers device and beverage together. Although it optimizes the plan cost, the robot may tip the beverage over the device. Whereas, in the explicable version robot delivers the device and beverage cup separately, resulting in an expensive plan due to multiple trips back and forth. [1].

Domain and explicability distance This domain is also represented in PDDL. Here both the models were provided by the domain expert. The robot model has the following actions available: `pickup`, `putdown`, `stack`, `unstack` and `move`. The domain modeler provided $\mathcal{M}_{\mathcal{H}}^{\mathcal{A}}$ based on usual expectations of robots with a similar form factor. For example, in $\mathcal{M}_{\mathcal{H}}^{\mathcal{A}}$, certain actions which could be perceived riskier (like, carrying the device and cup in the same tray) had a higher cost due to the possibility of damaging the items. Thus $\mathcal{M}_{\mathcal{H}}^{\mathcal{A}}$ incentivizes the planner to choose safer actions. Both models have same state space and action representation. The model differences lie in the action-costs as well as preconditions and effects of actions. There were 20 plan instances created for each of the 13 planning problems. Each of the plans was labeled by 2 human subjects, which resulted in 40 different training samples (some problems have multiple solution plans). The performance of different ensemble learning techniques is as shown in Table 2. We again

[1] A video demonstration can be viewed at `https://bit.ly/2JweeYk`

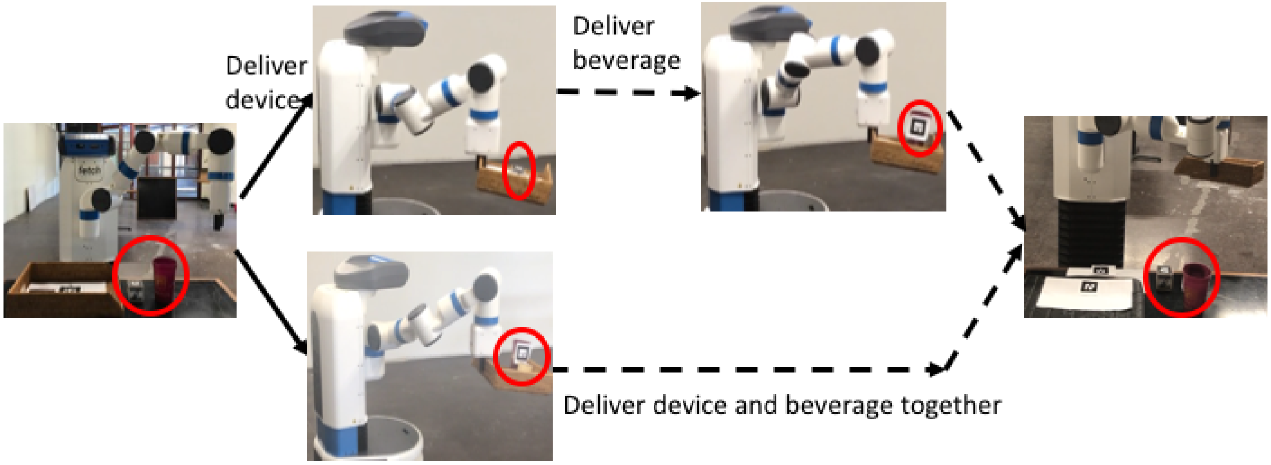


Figure 6: The goal of the agent is to deliver the device and beverage cup to the destination. In the cost-optimal plan, agent delivers both the items together, whereas in the explicable plan the agent delivers the items separately. A video demonstration can be viewed at `https://bit.ly/2JweeYk`

use the random forest regression model with an accuracy of 75%.

Evaluation For evaluation of this domain, 8 new planning problems (similar to the one shown in Figure 6) were used. These plan instances with a pictorial representation of intermediate behavioral states were labeled by 9 test subjects. For testing phase, the same protocol was followed as that in training phase. Figure 4 shows the comparison between the plan costs. Whenever the items consist of beverage cups, the robot has to do multiple trips, therefore all the explicable plans are more expensive than the optimal plans. For such scenarios, if a robot uses a cost-optimal planner, the plans chosen will always be inexplicable with respect to the plan context. In Figure 5, we compare the explicability scores provided by test subjects. The explicability scores provided by the subjects are higher for explicable plans. Some plans involved the robot stacking cups over devices to generate cost-optimal plans. These plans ended up receiving least scores.

5 Related Work

An important requirement for an AI agent collaborating with a human is the ability to infer human's goals and intents and use this information to inform its planning process. There have been various efforts in the direction of human aware planning to account for human's beliefs and goals (Sisbot et al. 2007; Cirillo, Karlsson, and Saffiotti 2010; Mainprice et al. 2011; Chakraborti et al. 2015; 2016) to encourage human agent interactions. Human-AI collaboration can involve both purely virtual tasks (Allen 1994; Ai-Chang et al. 2004; Sengupta et al. 2017), purely physical tasks (Alami et al. 2005; Montreuil et al. 2007) or their combination.

HRI community has focused on some related issues in the context of the human-AI collaboration, but they have mainly considered purely physical tasks involving manipulation and motion. Our multi-model explicability framework can be adapted to the HRI scenarios too. Indeed, there has been some work on explicability of motion planning that goes under the name of *predictability* (Dragan and Srinivasa 2013; Fisac et al. 2016; Knepper et al. 2017). Predictability assumes the goal is known and deals with generating likely (straightforward) motion towards it (when the goal is unknown, legibility is defined in this context, which conveys with a motion which goal the robot is reaching for). These can be understood in terms of $\mathcal{M}^{\mathcal{A}}$ - $\mathcal{M}^{\mathcal{A}}_{\mathcal{H}}$ models. The difference is that for motion tasks, the advantage is that humans have well understood intuitive/common-sense models (e.g. move in straight lines as much as possible). This reduces the difficulty of generating explicable plans considerably, as the human model is fixed, and the *distance* between the expected and observed trajectories can be easily measured. However, in many general domains that may involve both physical and virtual tasks (or purely virtual tasks), there are no such widely accepted intuitive human models. In such cases, the full generality of our approach will be necessary.

Previously, (Zhang et al. 2017) showed the generation of explicable plans by approximating $\mathcal{M}^{\mathcal{A}}_{\mathcal{H}}$ to a labeling scheme given that was learned using conditional random fields. In this work, we study how the problem changes when the human model is known and is represented in terms of an action model similar in representation to the agent's model. In many domains (household/factory scenarios), there is a clear expectation of how a task should be performed and M^{A}_{H} can be constructed using relevant norms. Most deployed products make use of inbuilt models of user expectations. This allows for a more informed plan generation process, which we refer to as *reconciliation search*. This assumption on model representation also allows us to investigate the relationship between the traditional measures of plan distances studied in the planning community and *explicability measure* implicitly used by the humans to compare the agent plans with the expected plans from their model.

6 Conclusion

In conclusion, we showed how the problem of explicable plan generation can be modeled in terms of plan distance measures studied in existing literature. We also demonstrated how a regression model on these distance measures can be learned from human feedback and used to guide an agent's plan generation process to produce increasingly explicable plans in the course of an anytime search process. We demonstrated the effectiveness of this approach in a simulated car domain and a physical robot delivery domain.

Going forward, two interesting avenues of future work include (1) learning to model plan similarities when domains are represented at different levels of abstraction (e.g. with experts versus non-experts in the loop); and (2) exploring interesting modes of behavior that can build upon the developed understanding of human expectations (e.g. by being inexplicable to draw attention to an event or, in the extreme case, being deceptive by conforming to or reinforcing wrong human expectations). The former may require a rethink of existing plan distance metrics to account for the effects of model difference in the presence of lossy abstractions.

Acknowledgements

This research is supported in part by the ONR grants N00014161-2892, N00014-13-1-0176, N00014-13-1-0519, N00014-15-1-2027, and the NASA grant NNX17AD06G.

References

Ai-Chang, M.; Bresina, J.; Charest, L.; Chase, A.; Hsu, J.-J.; Jonsson, A.; Kanefsky, B.; Morris, P.; Rajan, K.; Yglesias, J.; et al. 2004. Mapgen: mixed-initiative planning and scheduling for the mars exploration rover mission. *IEEE Intelligent Systems* 19(1):8–12.

Alami, R.; Clodic, A.; Montreuil, V.; Sisbot, E. A.; and Chatila, R. 2005. Task planning for human-robot interaction. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, 81–85. ACM.

Allen, J. F. 1994. Mixed initiative planning: Position paper. In *ARPA/Rome Labs Planning Initiative Workshop*.

Chakraborti, T.; Briggs, G.; Talamadupula, K.; Zhang, Y.; Scheutz, M.; Smith, D.; and Kambhampati, S. 2015. Planning for serendipity. In *IROS*.

Chakraborti, T.; Zhang, Y.; Smith, D. E.; and Kambhampati, S. 2016. Planning with resource conflicts in human-robot cohabitation. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*.

Chakraborti, T.; Sreedharan, S.; Zhang, Y.; and Kambhampati, S. 2017. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. In *IJCAI*.

Cirillo, M.; Karlsson, L.; and Saffiotti, A. 2010. Human-aware task planning: An application to mobile robots. *ACM Transactions on Intelligent Systems and Technology (TIST)* 1(2):15.

Dragan, A., and Srinivasa, S. 2013. Generating legible motion. In *Proceedings of Robotics: Science and Systems*.

Fan, X.; Oh, S.; McNeese, M.; Yen, J.; Cuevas, H.; Strater, L.; and Endsley, M. R. 2008. The influence of agent reliability on trust in human-agent collaboration. In *Proceedings of the 15th European conference on Cognitive ergonomics: the ergonomics of cool interaction*, 7. ACM.

Fisac, J.; Liu, C.; Hamrick, J. B.; Sastry, S.; Hedrick, J. K.; Griffiths, T. L.; and Dragan, A. D. 2016. Generating plans that predict themselves. In *Workshop on the Algorithmic Foundations of Robotics (WAFR)*.

Helmert, M. 2006. The fast downward planning system. *J. Artif. Intell. Res.(JAIR)* 26:191–246.

Knepper, R. A.; Mavrogiannis, C. I.; Proft, J.; and Liang, C. 2017. Implicit communication in a joint action. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, 283–292. ACM.

Kyle Hollins Wray, Stefan J. Witwicki, S. Z. 2017. Online decision-making for scalable autonomous systems. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 4768–4774.

Langley, P. 2016. Explainable agency in human-robot interaction. In *AAAI Fall Symposium Series*.

Mainprice, J.; Sisbot, E. A.; Jaillet, L.; Cortés, J.; Alami, R.; and Siméon, T. 2011. Planning human-aware motions using a sampling-based costmap planner. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 5012–5017. IEEE.

Montreuil, V.; Clodic, A.; Ransan, M.; and Alami, R. 2007. Planning human centered robot activities. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, 2618–2623. IEEE.

Nguyen, T. A.; Do, M.; Gerevini, A. E.; Serina, I.; Srivastava, B.; and Kambhampati, S. 2012. Generating diverse plans to handle unknown and partially known user preferences. *Artificial Intelligence* 190(0):1 – 31.

Sadigh, D.; Sastry, S.; Seshia, S. A.; and Dragan, A. D. 2016. Planning for autonomous cars that leverage effects on human actions. In *Robotics: Science and Systems*.

Sengupta, S.; Chakraborti, T.; Sreedharan, S.; Vadlamudi, S. G.; and Kambhampati, S. 2017. Radar-a proactive decision support system for human-in-the-loop planning. *AAAI Fall Symposium on Human-Agent Groups: Studies, Algorithms and Challenges*.

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

Srivastava, B.; Nguyen, T. A.; Gerevini, A.; Kambhampati, S.; Do, M. B.; and Serina, I. 2007. Domain independent approaches for finding diverse plans. In *IJCAI*, 2016–2022.

Zhang, Y.; Sreedharan, S.; Kulkarni, A.; Chakraborti, T.; Zhuo, H. H.; and Kambhampati, S. 2017. Plan explicability and predictability for robot task planning. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*.