Artificial Intelligence 335 (2024) 104181

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

Planning with mental models – Balancing explanations and explicability

Sarath Sreedharan [a,*], Tathagata Chakraborti [b], Christian Muise [c,1], Subbarao Kambhampati [d,2]

[a] Colorado State University, Fort Collins, CO 80524, USA
[b] IBM Research, Cambridge, MA 02142, USA
[c] Queen's University, Kingston, ON K7L 3N6, Canada
[d] Arizona State University, Tempe, AZ 85281, USA

ARTICLE INFO

Keywords:
Explainable AI
Automated planning
Plan explanations
Explicable planning
Interpretability
Mental models

ABSTRACT

Human-aware planning involves generating plans that are explicable, i.e. conform to user expectations, as well as providing explanations when such plans cannot be found. In this paper, we bring these two concepts together and show how an agent can achieve a trade-off between these two competing characteristics of a plan. To achieve this, we conceive a first-of-its-kind planner MEGA that can reason about the possibility of explaining a plan *in the plan generation process itself*. We will also explore how solutions to such problems can be expressed as "self-explaining plans" – and show how this representation allows us to leverage classical planning compilations of epistemic planning to reason about this trade-off at plan generation time without having to incur the computational burden of having to search in the space of differences between the agent model and the mental model of the human in the loop in order to come up with the optimal trade-off. We will illustrate these concepts in two well-known planning domains, as well as with a robot in a typical search and reconnaissance task. Human factor studies in the latter highlight the usefulness of the proposed approach.

1. Introduction

1.1. Planning with mental models

It is often useful for an agent while interacting with a human to use, in the process of its deliberation, not only its own model $\mathcal{M}^R$ of the task but also the model $\mathcal{M}_h^R$[3] that the human thinks it has (as shown in Fig. 1). This mental model [1] is in addition to the task model of the human M_r^H (denoting their beliefs, intentions, and capabilities). This is, in essence, the fundamental thesis of the recent works on **plan explanations** as model reconciliation [2] and **explicable planning** [3] and is in addition to the originally

[*] Corresponding author.
E-mail address: sarath.sreedharan@colostate.edu (S. Sreedharan).

[1] Work done while at IBM Research (Cambridge).

[2] Work done while at ASU.

[3] Please note that in practice, the robot would only have access to an approximation of the true model $\mathcal{M}_h^R$. However, through this paper, we will generally assume that the robot's approximation of the model is the same as the true model $\mathcal{M}_h^R$.

https://doi.org/10.1016/j.artint.2024.104181
Received 8 February 2023; Received in revised form 20 May 2024; Accepted 10 July 2024

Available online 18 July 2024
0004-3702/© 2024 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

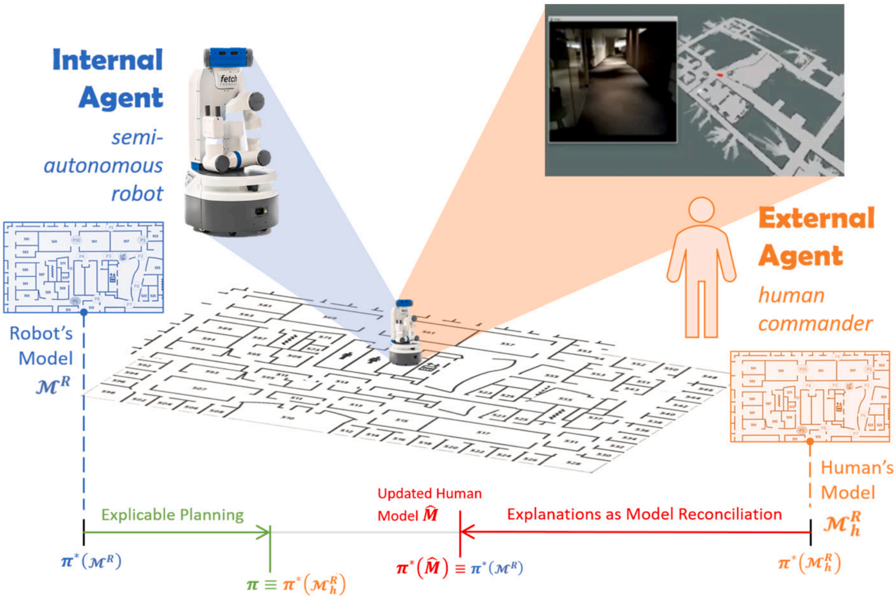

Internal Agent
semi-autonomous robot
Robot's Model $\mathcal{M}^R$
External Agent
human commander
Human's Model $\mathcal{M}^R_h$
Explicable Planning
Updated Human Model $\widehat{\boldsymbol{M}}$
Explanations as Model Reconciliation
$\pi^*(\mathcal{M}^R)$
$\pi \equiv \pi^*(\mathcal{M}^R_h)$
$\pi^*(\widehat{\boldsymbol{M}}) \equiv \pi^*(\mathcal{M}^R)$
$\pi^*(\mathcal{M}^R_h)$

Fig. 1. The explicability versus explanation trade-off in planning with expectations of the human in the loop. The robot accounts for the human's (mental) model of itself in addition to its own model – it can either bring the former closer to its own using explanations via the model reconciliation process so that an otherwise inexplicable plan now makes sense in the human's updated model and/or it can produce explicable plans which are closer to the human's expectation of optimality.

studied *human-aware planning* (HAP) problems where actions of the human (i.e. the *human task model* and a robot's belief of it) are involved in the planning process. The need for explicability and explanations occur when these two models – $\mathcal{M}^R$ and $\mathcal{M}^R_h$ – diverge. This means that the optimal plans in the respective models – $\pi^*_{\mathcal{M}^R}$ and $\pi^*_{\mathcal{M}^R_h}$ – may not be the same and hence optimal behavior of the robot in its own model may seem inexplicable to the human.

- In **explicable planning**, the robot produces a plan $\bar{\pi}$ that is closer to the human's expected plan, i.e. $\bar{\pi} \approx \pi^*_{\mathcal{M}^R_h}$.
- During **plan explanation** (as model reconciliation), it updates the mental model to an intermediate model $\bar{\mathcal{M}}^R_h$ in which the robot's original plan is *equivalent* (with respect to a metric such as cost or similarity) to the optimal one and hence explicable, i.e. $\pi^*_{\mathcal{M}^R} \equiv \pi^*_{\bar{\mathcal{M}}^R_h}$.

Until now, these two processes of plan explanations and explicability have remained separate in so far as their role in an agent's deliberative process is considered - i.e. a planner either generates an explicable plan to the best of its ability or produces explanations of its plans where they required. However, there are situations where a combination of both provides a much better course of action – if the expected human plan is too costly in the planner's model (e.g. the human might not be aware of some safety constraints) or the cost of communication overhead for explanations is too high (e.g. limited communication bandwidth). Consider, for example, a human working with a robot that has just received a software update allowing it to perform new complex maneuvers. Instead of directly trying to conceive all sorts of new interactions right away that might end up spooking the user, the robot could instead reveal only certain parts of the new model while still using its older model (even though suboptimal) for the rest of the interactions so as to slowly reconcile the drifted model of the user. This is the focus of the current paper where we try to attain the sweet spot between plan explanations and explicability during the planning process.

1.2. Contributions and outline

Section 3 We develop a first-of-its-kind planner that can envisage possible explanations required of its plans and incorporate these considerations in the planning process itself, thereby trading off the cost of conforming to human expectations (explicability) with the cost of explanations.

Section 3.1 Since the explicability problem has been studied in plan space while explanation generation works in model space, a viable solution to the balancing act cannot be a simple combination of the two. Our planner not only computes a plan given a model but also *what model to plan in* given the human mental model. This also means that, in contrast to explicability-only approaches, we can deal with situations where an explicable plan does not exist by being able to reconcile model differences in the same planning framework.

Section 3.2 We show how this allows us to generate explanations that are even shorter than the previously proposed shortest possible or "minimally complete" explanations, given a plan, in [2].

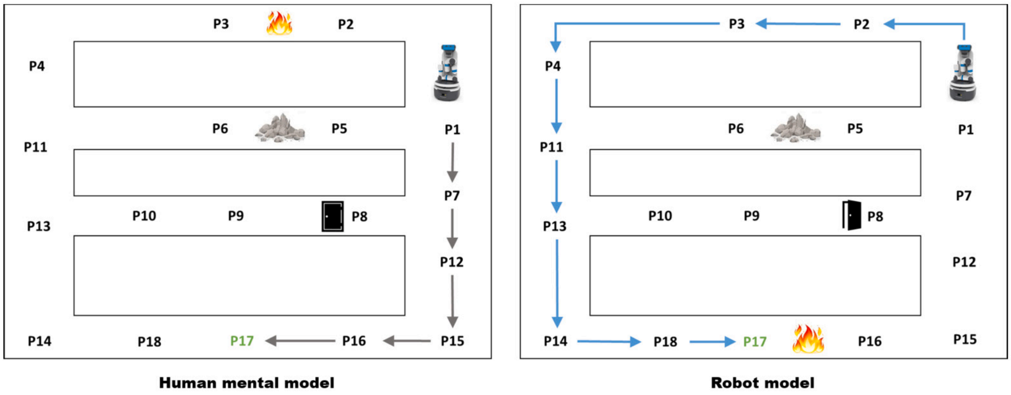


Fig. 2. The robot starts at P1 and needs to go to P17. The human incorrectly believes that the path from P16 to P17 is clear and the one from P2 to P3 is blocked due to fire. Both agents know that there is movable rubble between P5 and P6 which can be cleared with a costly clear_passage action. Finally, in the mental model, the door at P8 is locked while it is unlocked in the model for the robot which cannot open unlocked doors.

Section 4 We then present the first unification of various threads of planning with differing human expectation: including acting in-accordance with the human expectation (explicability), bridging model asymmetry through implicit (epistemic effects of plan execution on the mental model) and explicit communication (explanations).

Section 4.1 We show how this formulation is complete (as compared to the model space search approach) and also lends itself to compilation to classical planning which provides significant computational advantage.

Section 4.2 - 4.3 We show how this unified framework allows us to extend the scope of novel balancing behaviors, such as being able to model epistemic effects of observed execution.

Section 5 - 6 Finally, we illustrate the salient features of the approach in two well-known planning domains and in a human factors study in a mock search and rescue domain. The empirical evaluations demonstrate the effectiveness of the approach from the robot's perspective, while the study highlights its usefulness in being able to conform to expected normative behavior.

1.3. Running example: the USAR domain

We will use as a running example, a robot performing an Urban Search And Reconnaissance (USAR) task – here a remote robot is put into disaster response operation controlled partly or fully by an external human commander, as seen in Fig. 1. This is a typical USAR setup [4] where the robot's job is to infiltrate areas that may be otherwise harmful to humans, and report on its surroundings as and when required or instructed by the external. The external usually has a map of the environment, but this map is no longer accurate in a disaster setting – e.g. new paths may have opened up or older paths may no longer be available due to rubble from collapsed structures like walls and doors. The robot however may not need to inform the external of all these changes so as not to cause information overload on the commander who may be otherwise engaged in orchestrating the entire operation. This requires the robot to reason about the model differences due to changes in the map, i.e. the initial state of the planning problem (the human model has the original unaffected model of the world).

Fig. 2 shows a relevant section of the map of the environment where this whole scenario plays out. In this case, the robot's path could be blocked by various obstacles. Some of which are visualized in the map include rubble (which can be moved), locked doors, and some of the paths are even blocked by fire. The robot starts at P1 and needs to travel to its goal at P17. The optimal plan expected by the commander is highlighted in grey in their map and involves the robot moving through waypoint P7 and follow that corridor to go to P15 and then finally to P16. The robot knows that it should in fact be moving to P2 – its optimal plan is highlighted in blue.[4] This disagreement arises from the fact that the human incorrectly believes that the path from P16 to P17 is clear while that from P2 to P3 is blocked. In this case, the robot could follow the plan that takes the robot through P5 and P6 with a clear_passage_P5_P6. The robot needs to only explain a single initial state change to make it optimal in the updated map of the commander.

```
Explanation >> remove-has-initial-state-clear P16 P17
```

This is an instance where the plan closest to the human expectation, i.e. the most explicable plan, still requires an explanation, which previous approaches in the literature cannot provide. Moreover, in order to follow this plan, the robot must perform the costly `clear_passage` action to traverse the corridor between `P5` and `P6`, which it could have avoided in its optimal path (through P2 highlighted in blue). Indeed, the robot's optimal plan requires the following explanation:

```
Explanation >> add-has-initial-state-clear P2 P3
Explanation >> remove-has-initial-state-clear P16 P17
```

[4] For interpretation of the references to color please refer to the web version of this article.

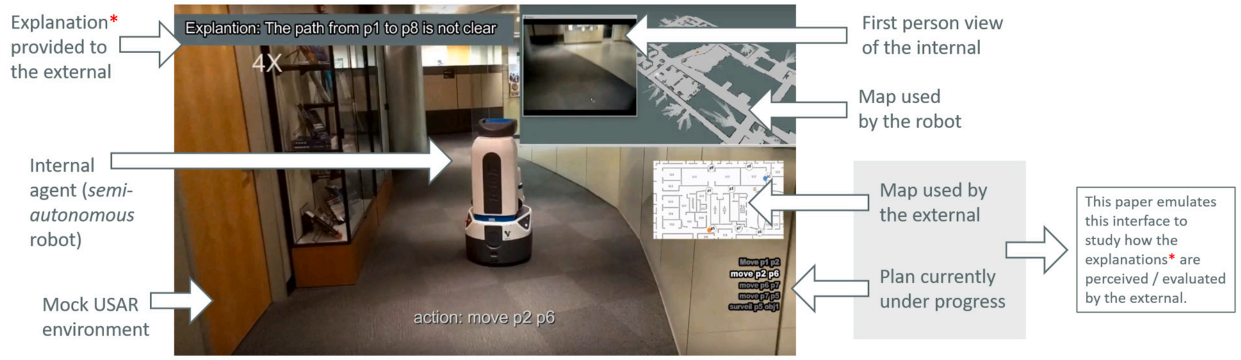

Explanation* provided to the external
Explantion: The path from p1 to p8 is not clear
4X
First person view of the internal
Map used by the robot
Internal agent (*semi-autonomous* robot)
Map used by the external
Move p1 p2
move p2 p6
move p6 p7
move p7 p5
surveil p5 obj1
Plan currently under progress
Mock USAR environment
action: move p2 p6
This paper emulates this interface to study how the explanations* are perceived / evaluated by the external.

Fig. 3. Illustration of the simulated USAR setting. We expose a mock interface to the external agent (right inset) on the browser to study the properties of different explanations afforded by the model reconciliation framework. (Section 6 / Fig. 6).

By providing this explanation, the robot can convey to the human the optimality of the current plan as well as the infeasibility of the human's expected plan.

For the user studies later in Section 6, we will expose to participants a toy version of the interface for the external. (See Fig. 3.)

2. Expectation-aware planning

2.1. Background

Definition 1. A Classical Planning Problem is a tuple $\mathcal{M} = \langle D, I, G\rangle$ with domain $D = \langle F, A\rangle$ - where F is a set of propositional fluents that define a state space associated with the planning problem, such that for every state s, we have $s \subseteq F$, and A is the set of actions, and I and G are the initial and goal states, such that $I, G \subseteq F$. Action $a \in A$ is a tuple $\langle c_a, pre(a), eff^{\pm}(a)\rangle$ where c_a is the cost, and $pre(a), eff^{\pm}(a)$ are the preconditions and add/delete effects [5].

Note that the above formulation uses the popular 'set notation' to capture states, whereby each state is uniquely identified by the propositions that are true in a given state. However, when we use a state along with a logical operator such as entailment (⊨), we will equivalently treat it as a logical formula composed of a conjunction of fluents that are part of the state and the negation of fluents that are absent from the state, i.e., a state s corresponds to a logical formula

$$\bigwedge_{f \in S} f \wedge \bigwedge_{f' \in F \setminus s} \neg f'.$$

The precondition is a propositional formula defined over state fluents such that an action a can only be executed in a state s if $s \models \text{pre}(a)$. As with the state, we can also use a set notation for preconditions if the precondition is purely a conjunction over a set of positive literals. The effects are generally of the form $c \rightarrow e$, where the antecedent represents the condition under which the effect $e \in F$ should be applied (where the fluent corresponding to e is set to true in the state if $c \rightarrow e$ is part of the add effects, and if it is part of the delete it is set to false). In general, we can define a transition function as follows:

Definition 2. For a given planning problem $\mathcal{M}$, the transition function is defined as $\delta_{\mathcal{M}} : 2^F \times A \rightarrow 2^F$, such that

$$\delta_{\mathcal{M}}(s, a) = \begin{cases} (s \setminus \text{del\_set}(s, a)) \cup \text{add\_set}(s, a), & \text{if } s \models pre(a) \\ \textit{undefined} & \text{otherwise} \end{cases}$$

Where del_set(s, a) and add_set(s, a) contains the delete and add effects where the conditions are satisfied in the current state s, i.e.,

$$\text{del\_set}(s, a) = \{e \mid c \rightarrow e \in eff^{-}(a) \wedge s \models c\},$$
$$\text{add\_set}(s, a) = \{e \mid c \rightarrow e \in eff^{+}(a) \wedge s \models c\}.$$

We will overload the transition function notation to also apply to action sequences, where the execution of sequences of actions is defined by subsequent action application. Also, when the effect is not conditional, which is equivalent to saying c is True, we will replace $c \rightarrow e$, with just the effect fluent e. In general, most starting models we will consider will be free of conditional effects and focus on actions with conjunctive preconditions (with only positive literals). We will primarily rely on conditional effects to support the planning compilations that we will introduce in Section 4.1.

Note that the "model" $\mathcal{M}$ of a planning problem includes the action model *as well as the initial and goal states*. The solution to $\mathcal{M}$ is a sequence of actions or a (satisficing) *plan* $\pi = \langle a_1, a_2, \ldots, a_n\rangle$ such that $\delta_{\mathcal{M}}(I, \pi) \models G$. The cost of a plan π is $C(\pi, \mathcal{M}) = \sum_{a \in \pi} c_a$ if $\delta_{\mathcal{M}}(I, \pi) \models G$; ∞ otherwise. The optimal plan has cost $C^*_{\mathcal{M}}$. We will use $\Pi^*_{\mathcal{M}}$ to represent the set of all optimal plans for $\mathcal{M}$.

We rely on a model parameterization function to represent a given model by using a set of propositional features. This set of features will become the basis of defining the model update operations. These model updates will become the basis of the explanation formulation we will be using in this paper. Following conventions used by [6], we will start by defining the feature space by using the fluent names and the action labels.

Definition 3. For a given pair of condition effect free models $\mathcal{M}^1 = \langle D^1, I^1, G^1 \rangle$ and $\mathcal{M}^2 = \langle D^2, I^2, G^2 \rangle$, where $D^1 = \langle F^1, A^1 \rangle$ and $D^2 = \langle F^2, A^2 \rangle$, the *set of model space propositions* for these two seed models, $\mathcal{F}$, is defined as follows:

$$\begin{aligned}\mathcal{F} = \{&\text{“\textit{init-has-}”} \cdot f \mid f \in F^1 \cup F^2\} \cup \{\text{“\textit{goal-has-}”} \cdot f \mid f \in F^1 \cup F^2\}\\ &\bigcup_{a \in A^1 \cup A^2} \{a \cdot \text{“\textit{-has-precondition-}”} \cdot f, a \cdot \text{“\textit{-has-add-effect-}”} \cdot f,\\ &\qquad\qquad a \cdot \text{“\textit{-has-del-effect-}”} \cdot f \mid f \in F^1 \cup F^2\}\\ &\cup \{a \cdot \text{“\textit{-has-cost-}”} \cdot c_a \mid a \in A^1\} \cup \{a \cdot \text{“\textit{-has-cost-}”} \cdot c_a \mid a \in A^2\}.\end{aligned}$$

Here, the label of each new model space proposition is formed by concatenating the model-component string (*init-has-*, *goal-has-*, *-has-precondition-*, etc.), with the original proposition labels (and action label when applicable).

These model space propositions are sufficient to uniquely identify any models that share fluents, action labels, and cost values with the seed models. We will use the notation $\mathbb{M}$ to represent the models that meet these criteria. Note that our notations ($\mathcal{F}$ and $\mathbb{M}$) do not specify the seed models since our paper will focus on cases where the seed models are always fixed.

With the basic definition in place, our next step will be to define a model parameterization function $\Gamma : \mathbb{M} \rightarrow 2^{\mathcal{F}}$ that can translate any model into a representation in terms of these model space propositions.

We will define the application of the function on a model $\mathcal{M} = \langle D, I, G \rangle$, where $D = \langle F, A \rangle$, as follows:

$$\begin{aligned}\tau_I^{\mathcal{M}} &= \{\text{“\textit{init-has}”} \cdot f \mid f \in I\}\\ \tau_G^{\mathcal{M}} &= \{\text{“\textit{goal-has-}”} \cdot g \mid g \in G\}\\ \tau_{pre(a)}^{\mathcal{M}} &= \{a \cdot \text{“\textit{-has-precondition-}”} \cdot f \mid f \in pre(a)\}\\ \tau_{eff^+(a)}^{\mathcal{M}} &= \{a \cdot \text{“\textit{-has-add-effect-}”} \cdot f \mid \mid f \in eff^+(a)\}\\ \tau_{eff^-(a)}^{\mathcal{M}} &= \{a \cdot \text{“\textit{-has-del-effect-}”} \cdot f \mid \mid f \in eff^-(a)\}\\ \tau_{c_a}^{\mathcal{M}} &= \{a \cdot \text{“\textit{-has-cost-}”} \cdot c_a\}\\ \tau_a^{\mathcal{M}} &= \tau_{pre(a)}^{\mathcal{M}} \cup \tau_{eff^+(a)}^{\mathcal{M}} \cup \tau_{eff^-(a)}^{\mathcal{M}} \cup \tau_{c_a}^{\mathcal{M}}\\ \tau_A^{\mathcal{M}} &= \bigcup_{a \in A^{\mathcal{M}}} \tau_a^{\mathcal{M}}\\ \Gamma(\mathcal{M}) &= \tau_I^{\mathcal{M}} \cup \tau_G^{\mathcal{M}} \cup \tau_A^{\mathcal{M}}\end{aligned}$$

We will use the function Γ^{-1} to map back from parameterized representations into models and τ^{-1} to map a model-space feature into the original fluent. Additionally, we will use the operator '+' to denote the application of a set of model updates. In particular, a specific set of model updates is captured by a tuple $\mathcal{E} = \langle \mathcal{E}^+, \mathcal{E}^- \rangle$, where $\mathcal{E}^+ \subseteq \mathcal{F}$ represents the set of new additions to the model and $\mathcal{E}^- \subseteq \mathcal{F}$ represent the set of components will be removed. In particular, we define the application of model updates as

$$\mathcal{M} + \mathcal{E} = \Gamma^{-1}((\Gamma(\mathcal{M}) \setminus \mathcal{E}^-) \cup \mathcal{E}^+))$$

We will use the term *Model-Space Search* to refer to search over the space of models defined by $\mathcal{F}$. In some of our discussions, we will talk about cases where we would want to minimize the cost of the model updates. While one can explore more general notions of costs in the context of model updates, we will mostly limit ourselves to assuming that each model update has a unit cost.

2.2. Expectations and mental models in planning

Interfacing with humans adds a new component to single-agent planning – the mental model of the human. This manifests itself in the form of expectations that the human has of the agent. Such a mental model can be represented as a version of the problem at hand which the agent believes the human is operating with[5] [2]. This brings us to an extension of the classical planning paradigm that accepts as inputs the agent model as well as a symbolic estimate of the mental model of the human in order to account for the expectations of the human in the planning process. Note that access to a symbolic model of the human's belief about the robot, **does**

[5] The actual decision-making problem may be over a graph, a planning problem, a logic program, etc. Many of the concepts discussed in the paper, though confined to automated planning, do in fact carry over beyond the actual representation.

not assume that humans maintain an explicit symbolic representation of their beliefs about the task as a classical planning model. The robot only uses this to represent the information content of human belief. As discussed before, the robot does not have direct access to it. Throughout this work, we will assume that we have access to an exact estimate of human belief and use this to generate both the explanation and explicable behavior. However, it is worth noting that there exist works that we could leverage to extend our method to support scenarios where this assumption may not be met. This includes methods to learn an estimate of the human mental models (cf. [3,7]), which our algorithms can use. There are also works that show how we can use these model estimates even when they are incomplete or noisy.

Another assumption we will make throughout this work is that the robot model reflects the ground truth and the human's mental model is wrong. This is the case in many tasks where the model is constantly being updated or refined based on newly received data, as in the case of USAR. Additionally, systems like RADAR decision-support system [8] have shown that, even in cases where this assumption is not met, one can still produce useful explanations by acting as if the assumption holds. In such cases, the robot may generate information that the human knows to be false, and they can ask the robot to update its model. However, such interaction occurs on top of the explanation/behavior generation process and is thus outside the scope of this work.

Definition 4. An Expectation-Aware Planning Problem (EAP) is the tuple $\Psi = \langle \mathcal{M}^R, \mathcal{M}_h^R \rangle$ where $\mathcal{M}^R = \langle D^R, I^R, G^R \rangle$ and $\mathcal{M}_h^R = \langle D_h^R, I_h^R, G_h^R \rangle$ are the agent's model and the human's understanding of the same.

While the human is under the assumption that $\mathcal{M}_h^R$ is an accurate representation of the task at hand, the model could be different from $\mathcal{M}^R$ in terms of action definitions, the initial state, and the goal. This difference means that plans generated for the model $\mathcal{M}^R$ may have different properties in the mental model $\mathcal{M}_h^R$. For example, a plan π^* that is optimal in $\mathcal{M}^R$ may be considered suboptimal or even un-executable by the human. Existing literature has explored two kinds of solutions to EAP problems, as discussed below.

2.3. Explicable plans

An explicable solution to an EAP is a plan π that is (1) executable in the robot's model and (2) closest to the expected (optimal) plan in the human's mental model of the robot:

(1) $\delta_{\mathcal{M}^R}(I^R, \pi) \models G^R$; and
(2) $C(\pi, \mathcal{M}_h^R) \approx C^*_{\mathcal{M}_h^R}$.

"Closeness" or distance to the expected plan is modeled here in terms of cost optimality, but in general, this can be any metric such as plan similarity. However, in this work, we chose to focus on cost as the primary metric since ample psychological evidence suggests that perceived cost is one of the primary factors influencing human decision-making [9]. In fact, many widely used models of human decision-making, such as the noisy rational model [10], use cost as the only plan-specific feature differentiating between the various decisions. Additionally, providing optimal plans in the resulting model allows us to further leverage other explanatory techniques after the fact to justify further a selected plan (cf. [11,12]). Something that may not be possible under other measures of similarity when dealing with rational agents. After all, choosing an optimal plan can be justified by citing that the agent is trying to minimize the cost. Similarly, humans can easily verify that the current plan is better than any alternative they may have expected. In existing literature [3,13] this has been achieved by modifying the heuristic that guides the search process to be driven by the robot's knowledge of the human mental model. Such a heuristic can be either derived directly [13] from the mental model or *learned* [3] through interactions in the form of affinity functions between plans and their purported goals.

2.4. Plan explanations

The other approach would be to (1) compute optimal plans in the planner's model as usual, but also provide an explanation (2) in the form of a model update to the human so that (3) the same plan is now also optimal in the updated mental model. Thus, a solution for a plan π, involves an explanation $\mathcal{E} = \langle \mathcal{E}^+, \mathcal{E}^- \rangle$ consisting of a set of model updates (as discussed in Section 2.1) defined over model space features for the seed models $\mathcal{M}^R$ and $\mathcal{M}_h^R$[6] such that:

(1) $\mathcal{E}^+ \subseteq \Gamma(\mathcal{M}^R) \setminus \Gamma(\mathcal{M}_h^R)$, and $\mathcal{E}^- \subseteq \Gamma(\mathcal{M}_h^R) \setminus \Gamma(\mathcal{M}^R)$.
(2) $C(\pi, \mathcal{M}^R) = C^*_{\mathcal{M}^R}$;
(3) $\bar{\mathcal{M}}_h^R \longleftarrow \mathcal{M}_h^R + \mathcal{E}$; and
(4) $C(\pi, \bar{\mathcal{M}}_h^R) = C^*_{\bar{\mathcal{M}}_h^R}$.

As discussed, a model update may include a correction to the belief (goals or state information) as well as information pertaining to the action model itself, as illustrated in [2]. As a result of this explanation, the human and the agent both agree that the given

[6] Henceforth, any reference to a model space search or to model space features would be by default defined over seed models $\mathcal{M}^R$ and $\mathcal{M}_h^R$.

plan is the best possible the latter could have come up with. Note that whether there is no valid plan in the human model, or just a different one, does not make any difference. The solution is still an explanation so that the given plan is the best possible in the updated human model. On the other hand, if there is no plan in the robot model, the explanation ensures that there is no plan in the updated human model either.

In [2], we explored many such solutions – including ones that minimize length, called **minimally complete explanations** or MCEs:

$$\min |\mathcal{E}| \text{ such that } \pi \in \Pi^*_{\mathcal{M}_h^R+\mathcal{E}}$$

However, this was done post facto, i.e. the plan was already generated and it was just a matter of finding the best explanation for it. This not only ignores the possibility of finding better plans that are also optimal but with smaller explanations, but also misses avenues of compromise whereby the planner sacrifices its optimality to reduce the overhead of the explanation process. Our approach is capable of both enabling the agent to explain its plans and choosing plans that align with the user's expectations.

3. Balancing explanation and explicable behavior generation

In this paper, we propose a "balanced solution" to an EAP, with components of both explicability and explanation (we will succinctly refer to this as balanced planning). This means that a solution may consist of model information to be provided to the observer along with the plan that will be followed by the agent. By allowing the planner to reason about explanations relevant to a given plan, we will effectively create agents that are able to combine the complementary strengths of explicable planning and explanation generation. This method would even allow the agent to fall back to pure forms of these strategies when the situation demands it. We will call such plans *Balanced Plans* and represent them as a tuple of the form $(\pi, \mathcal{E})$, where π is the plan the agent will be following, and $\mathcal{E}$ is the explanatory information the robot will be providing the human.

Definition 5. A tuple $\langle \pi, \mathcal{E} \rangle$ is considered a balanced solution for an EAP problem $\Psi = \langle \mathcal{M}^R, \mathcal{M}_h^R \rangle$, if $\delta_{\bar{\mathcal{M}}^R}(\pi, I^R) \models G^R$ and $\delta_{\bar{\mathcal{M}}_h^R}(\pi, \bar{I}_h^R) \models \bar{G}_h^R$, where $\bar{\mathcal{M}}_h^R \longleftarrow \mathcal{M}_h^R + \mathcal{E}$, such that $\mathcal{E} = \langle \mathcal{E}^+, \mathcal{E}^- \rangle$, $\mathcal{E}^+ \subseteq \Gamma(\mathcal{M}^R) \setminus \Gamma(\mathcal{M}_h^R)$, and $\mathcal{E}^- \subseteq \Gamma(\mathcal{M}_h^R) \setminus \Gamma(\mathcal{M}^R)$.

However, the above-specified definition merely points to generating a 'valid' balanced solution. As with any planning problem, we would be interested in generating solutions that minimize the cost associated with solutions and potentially even identify optimal solutions. In its most general form, the problem of creating balanced plans involves optimizing for the following cost terms.

1. **Plan Cost** $C(\pi, \mathcal{M}^R)$: The first objective is the cost of the plan that the robot is going to follow.
2. **Communication Cost** $C(\mathcal{E})$: The next objective is the cost of communicating the explanation. Ideally, this cost should reflect both the cost accrued at the agent's end (corresponding to the cost of actions that need to be performed to communicate the information) and a cost relating to the difficulty faced by the human to understand the explanation. For the majority of this paper, we will use explanation length as a proxy for both aspects of explanation costs ($C(\mathcal{E}) = \mid \mathcal{E} \mid = |\mathcal{E}^+| + |\mathcal{E}^-|$): i.e. the larger an explanation, the harder it may be to understand for the human.
3. **Penalty of Inexplicability ($C_{IE}(\pi, \mathcal{M}_h^R + \mathcal{E})$)**: This corresponds to the cost the human attaches to the inexplicability of the generated plan in their updated mental model. We will generally assume this cost to be directly proportional to the inexplicability score related to the plan. As in the case of explicable planning, we could measure inexplicability in terms of several different metrics, but this paper will mostly focus on measuring inexplicability in terms of the difference between the cost of the current plan in the human's updated model and the cost of the expected plan. We will assume the penalty itself to be directly proportional to the absolute value of this difference.

While in the most general case generating a balanced plan would be a multi-objective optimization, for simplicity, we will assume that we have weight parameters that allow us to combine the individual costs into a single cost. Thus the cost of a balanced plan (which includes both an explanation and a plan), would be given as:

$$C((\mathcal{E}, \pi)) = C(\mathcal{E}) + \alpha * C(\pi, \mathcal{M}^R) + \beta * C_{IE}(\pi, \mathcal{M}_h^R + \mathcal{E}) \tag{1}$$

Proposition 1. *A balanced solution optimizing Equation* (1) *need not be unique.*

One can easily prove the validity of this proposition by constructing a simple example scenario where multiple balanced solutions return the same $C((\cdot, \cdot))$ value. For example, one could build a domain that only contains two copies of the same action, both of which have the goal fact as part of the add effect. Now, in the human belief, let the add effects of both actions be empty. Here, we can now build two balanced solutions with two sets of model update components and two different plans but will result in the same $C((\cdot, \cdot))$, regardless of the values α and β take.

This property is similar to standalone explicable plans and plan explanations. Interestingly, solutions that are equally good according to the cost model can turn out to be different in usefulness to the human, as investigated recently in [14] in the context of plan explanations. This can have similar implications to balanced solutions as well. Apart from Section 4.3.2, we will rarely look at optimizing this full cost term. Instead, we will look at some special cases of this general optimization problem. Below we will look at three classes of balanced planning problems, each looking at optimizing a successively more constrained version of the cost function defined above.

1. **Optimal Balanced Planning:** The first group obviously correspond to generating plans that optimizes for the cost function provided in Equation (1). Obviously the nature of the final solution still rely heavily on the values of the parameters α and β. We will take a look at the impact of these values on the nature of the solution generated in Section 4.3.2.
2. **Balanced Explicable Plans:** The first special case we could consider are the ones where we restrict ourselves to cases where the plan is perfectly explicable, i.e., optimal, in the resultant model. Therefore in our objective we can ignore the inexplicability penalty term and the optimization objective becomes.

$$\min_{(\pi,\mathcal{E})} \; C(\mathcal{E}) + \alpha * C(\pi, \mathcal{M}^R)$$
$$\text{subject to } IE(\pi, \mathcal{M}_h^R + \mathcal{E}) = 0$$

   Most of this paper will focus on generating this type of balanced plans. We will sometimes use the term optimal balanced explicable plans to refer to a plan that is a balanced explicable plan for a given α, with the lowest cost in the robot model. In other words, a balanced explicable plan π is an optimal balanced explicable plan, if there exists no other balanced explicable plan π', such that $C(\pi, \mathcal{M}^R) > C(\pi', \mathcal{M}^R)$. Note that while the plan may be optimal in the human's updated model, the plan need not be optimal for the robot. Which brings us to the next group of behavior.
3. **Balanced Optimal Plans:** In this case, we constrain ourselves to identifying not only plans and explanations that will ensure perfect explicability, but we also try to ensure that the plans are in fact optimal in the robot model (note that this carries an inherent bias that robot's task level actions are always more expensive than communication, which need not be true). Thus the objective in this case becomes just

$$\min_{(\pi,\mathcal{E})} \; C(\mathcal{E})$$
$$\text{subject to } IE(\pi, \mathcal{M}_h^R + \mathcal{E}) = 0$$
$$\text{and } C(\pi, \mathcal{M}^R) = C^*_{\mathcal{M}^R}$$

   Interestingly, this means identifying the optimal in the robot's model with the MCE with the minimal cost.

In the following sections, we will see how one could generate such balanced plans. For now, we will focus on identifying the model information and the plan that constitutes a balanced solution. We will overlook the question of 'how' to map it to a form so that it can be directly executed until Section 4.

3.1. The MEGA algorithm: model space search

We employ a *model space* A^* search[7] to compute the optimal Balanced Explicable Plan for a given α.[8] We call this novel planning technique MEGA (Multi-model Explanation Generation Algorithm). The set Λ of actions contains unit model change actions that make a single change to a domain at a time. In particular, we will focus on three main types of model updates:

1. Turn a fluent p true or false in the initial state – represented by the update $\mathcal{E} = \langle \{\text{"}\textit{init-has-}\text{"} \cdot p\}, \{\} \rangle$ and $\mathcal{E} = \langle \{\}, \{\text{"}\textit{init-has-}\text{"} \cdot p\} \rangle$, respectively.
2. Add or remove a fluent p from the precondition (or add or delete effect) list of an action a – represented by the update $\mathcal{E} = \langle \{a \cdot \text{"}\textit{-has-\{prec/adds/dels\}-}\text{"} \cdot p\}, \{\} \rangle$ and $\mathcal{E} = \langle \{\}, \{a \cdot \text{"}\textit{-has-\{prec/adds/dels\}-}\text{"} \cdot p\} \rangle$.
3. Add or remove a fluent p from the goal list – represented by the update $\mathcal{E} = \langle \{\text{"}\textit{goal-has-}\text{"} \cdot p\}, \{\} \rangle$ and $\mathcal{E} = \langle \{\}, \{\text{"}\textit{goal-has-}\text{"} \cdot p\} \rangle$, respectively.

In the search described in Algorithm 1, each search node corresponds to a tuple consisting of a model and a corresponding set of model updates (where one could obtain the model component by applying the model updates to $\mathcal{M}_h^R$). We will refer to the latter as the explanation component or just the explanation. The algorithm starts by initializing the min node tuple ($\mathcal{N}$) with the human mental model M_h^R and an empty explanation, and the same tuple is pushed into the fringe. At each point in the search, we expand the node that contains the explanation with minimum cost. For any node, the successor nodes correspond to the models that can be obtained by changing one model space feature in the current model, in line with their values in the model $\mathcal{M}^R$. We use the symbol λ to capture each such possible change. The cost associated with each model is then calculated to be $c + C(\lambda)$, where c is the cost associated with the previous node. For each new model $\bar{M}$ generated during model space search, we test if the objective value of the

[7] In this section, we will focus on primarily using the trivially admissible heuristic value of $h = 0$. It would be easy to include heuristics into the search as done in previous works (cf. [11]).

[8] As in [2] we assume that the mental model is known and has the same computation power: [2] also suggests possible ways to address this, the same discussions apply here as well.

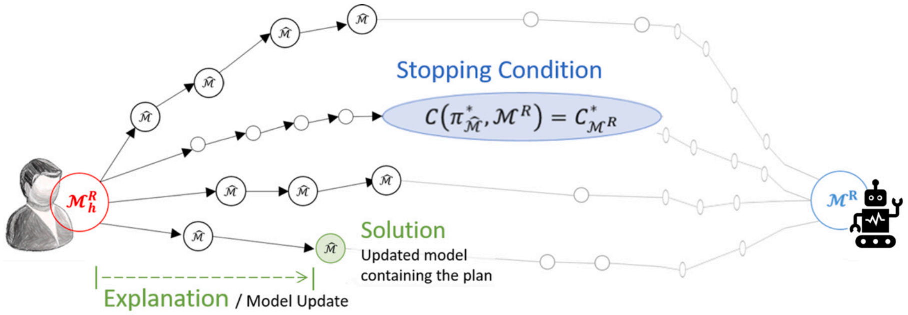


Fig. 4. Model space search to determine the best model to plan in w.r.t. the explanation versus explicability trade-off. The search stops at the blue node, which houses a model where the generated plan is optimal. The green node with the best value of the objective function is then selected as the solution.

Algorithm 1 MEGA.

```
procedure MEGA-Search
    Input: EAP $\Psi = \langle \mathcal{M}^R, \mathcal{M}^R_h \rangle$, $\alpha$
    Output: Plan $\pi$ and Explanation $\mathcal{E}$
    fringe ← Priority_Queue()
    c_list ← {}                                              ▷ Closed list
    $\mathcal{N}_{min} \leftarrow \langle \mathcal{M}^R_h, \{\} \rangle$                       ▷ Track node with min. value of obj.
    fringe.push($\langle \mathcal{M}^R_h, \{\} \rangle$, priority = 0)
    while True do
        $\langle \bar{\mathcal{M}}, \mathcal{E} \rangle, c \leftarrow$ fringe.pop()
        if OBJ_VAL($\langle \bar{\mathcal{M}}, \mathcal{E} \rangle$) ≤ OBJ_VAL($\mathcal{N}_{min}$) then
            $\mathcal{N}_{min} \leftarrow \langle \bar{\mathcal{M}}, \mathcal{E} \rangle$                ▷ Update min node
        for $\forall \pi^*_{\bar{\mathcal{M}}}$ do              ▷ We iterate over all optimal plans for the model $\bar{\mathcal{M}}$.
            if $C(\pi^*_{\bar{\mathcal{M}}}, \mathcal{M}^R) = C^*_{\mathcal{M}^R}$ then      ▷ Search is complete when $\pi^*_{\bar{\mathcal{M}}}$ is optimal in $\mathcal{M}^R$
                $\langle \mathcal{M}_{min}, \mathcal{E}_{min} \rangle \leftarrow \mathcal{N}_{min}$
                return $\langle \pi_{\mathcal{M}_{min}}, \mathcal{E}_{min} \rangle$
        c_list ← c_list $\cup\ \bar{\mathcal{M}}$
        for $f \in \Gamma(\bar{\mathcal{M}}) \setminus \Gamma(\mathcal{M}^R)$ do           ▷ Misconceptions in the mental model
            $\lambda \leftarrow \langle \{\}, \{f\} \rangle$                          ▷ Remove from $\bar{\mathcal{M}}$
            if $\bar{\mathcal{M}} + \lambda \notin$ c_list then
                fringe.push($\langle \bar{\mathcal{M}} + \lambda, \langle \mathcal{E}^+, \mathcal{E}^- \cup \{f\} \rangle \rangle$, $c + C(\lambda)$)
        for $f \in \Gamma(\mathcal{M}^R) \setminus \Gamma(\bar{\mathcal{M}})$ do           ▷ Missing conditions in the mental model
            $\lambda \leftarrow \langle \{f\}, \{\} \rangle$                          ▷ Add to $\bar{\mathcal{M}}$
            if $\bar{\mathcal{M}} + \lambda \notin$ c_list then
                fringe.push($\langle \bar{\mathcal{M}} + \lambda, \langle \mathcal{E}^+ \cup \{f\}, \mathcal{E}^- \rangle \rangle$, $c + C(\lambda)$)
    procedure OBJ_VAL($\langle \bar{\mathcal{M}}, \mathcal{E} \rangle$)
        return $C(\mathcal{E}) + \alpha \times \min_{\pi^*_{\bar{\mathcal{M}}}} C(\pi^*_{\bar{\mathcal{M}}}, \mathcal{M}^R)$
                                     ▷ Consider optimal plan in $\bar{\mathcal{M}}$ that is cheapest in $\mathcal{M}^R$
```

new node is smaller than the current min node by iterating over all possible optimal plans. We additionally track the visited nodes using the closed list c_list.

Note that Algorithm 1 considers a case where the explanation/model updates $\mathcal{E}$ could take arbitrary costs, represented by the function $C(\mathcal{E})$. The only assumption we place on the cost is that the costs are additive, i.e., if we combine two model updates $\mathcal{E}_1$ and $\mathcal{E}_2$, into a new model update set $\mathcal{E}_3$, then $C(\mathcal{E}_3) = C(\mathcal{E}_1) + C(\mathcal{E}_2)$. For cases where they have unit costs, the function maps into the cardinality of the model update set, i.e., $C(\mathcal{E}) = |\mathcal{E}^+| + |\mathcal{E}^-|$. We stop the search once we identify a model that is capable of producing a plan that is also optimal in the robot's own model. This is different from the original MCE-search [2] where we were trying to find the *first* node where a given plan is optimal. Finally, we select the node with the best objective value as the solution. Algorithm 1 provides details of the model space search process, while Fig. 4 provides a visual depiction of the same. The search stops upon reaching the blue node (which houses a possible model update where the generated plan is optimal). The green node with the best value of the objective function is then selected as the solution.

3.2. Properties of balanced explicable plans and MEGA algorithm

We will now compare and contrast some properties of the above-described algorithm and, in general, the problem of generating optimal balanced explicable plans. Note that the following propositions are proved by assuming that the explanation cost corresponds to the cardinality of the model update set.

Proposition 2. *For $\alpha \geq 0$, `MEGA` is guaranteed to find a balanced solution $\langle \mathcal{E}, \pi \rangle$ that minimizes the objective value $|\mathcal{E}| + \alpha * C(\pi, \mathcal{M}^R)$.*

Proof. The search space explores all possible sizes of model update sets until it reaches a set where an optimal robot plan is possible. So, if an optimal solution was possible under one of these model update sets, the algorithm should have identified it. Thus the only reason for possible reason for the algorithm to fail would be if the solution is outside of this set. We will show by contradiction that this is not possible for any $\alpha \geq 0$. To start with, let's assume that there exists a balanced solution $\langle \mathcal{E}', \pi' \rangle$ that was not part of the solutions that were considered, such that

$$|\mathcal{E}| + \alpha * C(\pi, \mathcal{M}^R) > |\mathcal{E}'| + \alpha * C(\pi', \mathcal{M}^R)$$

Let's assume the robot optimal plan π^* was optimal for the updated model obtained by applying the explanation set $\mathcal{E}^*$. Thus, the algorithm would exit once it evaluates $\mathcal{E}^*$. Now, we can assert that $|\mathcal{E}'| \geq \mathcal{E}^*$. This is because all explanations whose size is smaller than $|\mathcal{E}^*|$ are part of the sets that have been explored and, as such, don't contain the solution $\langle \mathcal{E}', \pi' \rangle$. For $\langle \mathcal{E}', \pi' \rangle$ to be better than any solution found, its objective value must be smaller than that corresponding to $\langle \mathcal{E}^*, \pi^* \rangle$. Given the fact that $|\mathcal{E}'| \geq \mathcal{E}^*$, this is only possible if $C(\pi', \mathcal{M}^R) < C(\pi^*, \mathcal{M}^R)$. This is impossible given the fact that π^* is optimal for $\mathcal{M}^R$. Hence, it contradicts our assumption and proves the proposition. □

Proposition 3. *`MEGA` yields a minimal explanation for any plan generated as part of the solution $(\pi, \mathcal{E})$.*

Proof. The proof for this proposition lies with the fact that the cost metric being optimized by `MEGA` corresponds to $|\mathcal{E}| + \alpha * C(\pi, \mathcal{M}^R)$. Any solution corresponds to the optimal value for this objective (by Proposition 2). Thus, for any plan selected part of the solution, if the explanation selected is not minimal, one could always select a solution with a lower objective value by using the smaller explanation set. Thus proving the proposition by contradiction. □

This means that with a high enough α the algorithm is guaranteed to compute the best possible plan for the robot as well as the smallest explanation associated with it. This is by the construction of the search process itself. This is beyond what is offered by the model reconciliation search in [2], which only computes the smallest explanation *given* a plan that is optimal in the planner's model.

Proposition 4. $\alpha = |\ \mathcal{M}^R\ \Delta\ \mathcal{M}^R_h\ |$ *(i.e. the total number of differences in the models) yields the most optimal plan in the planner's model along with the minimal explanation possible.*

Proof. The validity of this proposition is easy to see, since with $\forall \mathcal{E},\ |\mathcal{E}| \leq |\ \mathcal{M}^R\ \Delta\ \mathcal{M}^R_h\ |$, the latter being the total model difference, the penalty for departure from explicable plans is high enough that the search must choose from possible explanations only. In general, this works for any $\alpha \geq |MCE|$ but since an MCE will only be known retrospectively after the search is complete, the above condition suffices since the entire model difference is known up front and is the largest possible explanation. In other words, setting $\alpha = |\ \mathcal{M}^R\ \Delta\ \mathcal{M}^R_h\ |$ will force `MEGA` to generate Balanced optimal plans. □

Proposition 5. $\alpha = 0$ *yields the plan that requires the least explanation.*

Proof. Under this condition, the search minimizes the cost of explanations only – i.e. it will produce the plan that requires the shortest explanation. This brings us close to the original "explicability only" view of planning in [3,13], where the cost of robot plan is secondary to generating plans that are easier for the human to understand. Though unlike the earlier works, in this case, we limit ourselves to plans that are perfectly explicable, that is, plans that are optimal in the (possibly updated) human model. □

Remark. Note that the `MEGA`-search is required only once per problem and is independent of the hyperparameter α. The algorithm terminates only after all the nodes containing a minimally complete explanation have been explored. This means that for different values of α, the agent only needs to post-process the nodes with the new objective function in mind. Thus, a large part of the reasoning process for a particular problem can be pre-computed.

3.2.1. Approximate `MEGA`

`MEGA` evaluates executability (in the robot model) of all optimal plans within each intermediate model during the search. This is quite expensive. Instead, we implement `MEGA-approx` that does this check only for the first optimal plan that gets computed. This means that in Algorithm 1, we drop the loop (line 12) and only consider a single plan to calculate the objective value (line 26). This has the following consequences.

Proposition 6. *`MEGA-approx` is not complete.*

`MEGA-approx` is an optimistic version of `MEGA` and is not guaranteed to find all balanced solutions. This is because in each search node we are checking for whether an optimal plans – the first one that gets computed – is executable in the robot model, and moving on if not. In models where multiple optimal plans are possible, and some are executable in the robot model while others are not, this will result in MEGA-approx discarding certain models as viable solutions where a balanced plan was actually possible. The resulting incompleteness of the search also means we lose Propositions 2, 3 and 4.

Note that the above algorithm uses brute-force search to identify the required model updates. However, as we will see in the next section, we can leverage existing planning heuristics to guide this search, along with plan generation.

4. Self-explaining plans

Having established the salient features of balancing behavior and how to identify a valid balanced solution, we will now explore the question of how to capture it in a form such that these valid solutions could be carried out or 'executed'. In particular, we will consider translating balanced solutions to plans that contain explanatory actions as part of the planning model – hence, "self-explaining plans". Mapping balanced solutions into such self-explaining plans also allows us to leverage a compilation to classical planning that will get us out of having to perform an explicit model space search. At first glance, the need to keep track of both models and identify the model changes may make the problem of solving EAP planning problems considerably harder than the original decision-making problem. However, as we will see in Theorem 2, finding a valid solution in this setting is no harder than identifying valid plans for classical planning problems. This sets us off to the path for an efficient compilation.

One of the main challenges of compiling an EAP problem to traditional planning problems is to allow for a way to handle the identification of model updates and to account for the effect of these model updates on the user's expectation. A good way to go about this would be by acknowledging that if the observer is actually watching the agent executing a plan, these explanations can delivered through agent actions and hence modeled as communicative or *explanatory actions*. These actions can, in fact, be seen as actions with epistemic effects in as much as they are aimed towards modifying the human mental model (knowledge state). This means that a solution to an EAP planning problem can be seen as *self-explaining plans*, in the sense that some of the actions in the plan are aimed at helping people better understand the rest of it. Thus self-explaining plans can be seen are a different way to model balanced solutions. Any balanced solution can be represented as a self-explaining plan and vice-versa.

This puts EAP planning and balanced planning in general, squarely in the purview of epistemic planning [15], but the additional constraints enforced by the setting allow us to leverage relatively efficient methods to solve the problem at hand. These constraints include facts such as: all epistemic actions are public, modal depth is restricted to one, modal operators only applied to literals, for any literal the observer believes it to be true or false and the robot is fully aware of all of the observer beliefs [16].

Model updates in the form of epistemic effects of communication actions also open up the possibility of other actions having epistemic *side effects*. The definition of EAP puts no restriction on how the model update information is delivered. It is quite possible that actions that the agent is performing to achieve the goal (henceforth referred to as task-level actions to differentiate them from primary epistemic communication actions) could have epistemic side effects. This is something people leverage to simplify communication – e.g. one might avoid providing a prior description of some skill they are about to use when they can simply demonstrate it. So, one of our goals with the compilation is to allow for such epistemic side effects, a factor that has previously not been considered in any of the earlier works. This consideration also enables us to capture task-level constraints that may be imposed on the communication actions.

Now coming back to the MEGA algorithm. One way to convert the solution identified there into self-explaining plans would be to prepend the plan identified by MEGA with communication action corresponding to the model updates. To account for cases where the task-level actions may have epistemic side effects, we can update the MEGA algorithm to consider such side-effects when considering the final model over which the plan is evaluated. This primarily involves updating line 26 of Algorithm 1 to consider optimal plans in the model that results from both applying $\mathcal{E}$ and the side-effects of the actions in the plan. However, as discussed, in the absence of such epistemic side-effects, one could trivially convert the solutions generated by MEGA by replacing each piece of information with the corresponding communicative action.

However, instead of just providing us with an operationalizable representation of the balanced solutions, the self-explaining plans also provide us a way to map the problem of generating balanced solutions into classical planning. This allows us to fold the explicit model space search performed in MEGA algorithm into the planning process. This, in turn, allows us to leverage planning heuristics to direct the model space search.

4.1. Compilation to classical planning

Here, we adopt a formulation that is similar to the one introduced in [17] to compile reasoning about epistemic states into a classical planning problem. In our setting, each explanatory action can be viewed as an action with epistemic effects. One interesting distinction to make here is that the mental model now includes not only the human's belief about the task state but also their belief about the robot's model. This means that the planning model will need to separately keep track of (1) the current robot state, (2) the human's belief regarding the current state, (3) how actions would affect each of these (as humans may have differing expectations about the effects of each action) and (4) how those expectations change with explanations.

Given an EAP problem $\Psi = \langle \mathcal{M}^R, \mathcal{M}_h^R \rangle$, we will generate a new planning model $\mathcal{M}^\Psi = \langle D^\Psi, I^\Psi, G^\Psi, C_\Psi \rangle$ (where $D^\Psi = \langle F^\Psi, A^\Psi \rangle$) as follows $F^\Psi = F \cup F_\mathcal{B} \cup F_\mu \cup \{\mathcal{G}, \mathcal{I}\}$, where $F_\mathcal{B}$ is a set of new fluents that will be used to capture the human's belief about the task state and F_μ is a set of meta fluents that we will use to capture the effects of explanatory actions and $\mathcal{G}$ and $\mathcal{I}$ are special goal and initial state propositions. We will use the notation $\mathcal{B}(p)$ to capture the human's that the fluent p is true in a given state. We are able to use a single fluent to capture the human belief for each (as opposed to introducing two new fluents $\mathcal{B}(p)$ and $\mathcal{B}(\neg p)$ as used in previous work (cf. [17])) as we are specifically dealing with a scenario where (a) the human's belief about the robot model is known to the robot – i.e., the robot has no uncertainty about the human's estimate of the current state or their estimate of the robot model and (b) human either believes each of the fluent to be true or false in a given state – i.e., the human has no uncertainty regarding the initial state or what the robot's model maybe (though their estimate may differ from the robot's true model). In this case, we also do not require the additional rules in [17] to ensure that the state captures the deductive closure of the agent beliefs.

F_μ will contain an element for every part of the human model that the robot can change through explanations, i.e., $|F_\mu| = |\Gamma(\mathcal{M}^R)\Delta\Gamma(\mathcal{M}_h^R)|$. A meta fluent corresponding to a literal ϕ from the precondition of action a, which is currently missing from the human model, takes the form of $\mu^+_{\text{prec}}(\phi, a)$. The superscript "+" refers to the fact that the clause ϕ is part of the precondition of the action a in the robot model but not part of the human one, or more formally, $\tau(\phi) \in \Gamma(\mathcal{M}^R) \setminus \Gamma(\mathcal{M}_h^R)$. In other words, it captures the fact that the robot could add information about this model component to the human model through explanations. Similarly, for cases where the fluent represents an incorrect human belief (i.e., $\tau(\phi) \in \Gamma(\mathcal{M}_h^R) \setminus \Gamma(\mathcal{M}^R)$) we will be using the superscript "−".

For every action $a^R = \langle \text{prec}(a^R), \text{adds}(a^R), \text{dels}(a^R)\rangle \in A^R$ and its human counterpart $a_h^R = \langle \text{prec}(a_h^R), \text{adds}(a_h^R), \text{dels}(a_h^R)\rangle \in A_h^R$, we define a new action $a^\Psi = \langle \text{prec}(a^\Psi), \text{adds}(a^\Psi), \text{dels}(a^\Psi)\rangle \in A^\Psi \in \mathcal{M}^\Psi$ whose precondition is:

$$\begin{aligned}\text{prec}(a^\Psi) = \text{prec}(a^R) &\cup \{\mu^+(\phi, \text{prec}(a^R)) \rightarrow \mathcal{B}(\phi) | \phi \in \text{prec}(a^R) \setminus \text{prec}(a_h^R)\}\\ &\cup \{\mu^-(\phi, \text{prec}(a^R)) \rightarrow \mathcal{B}(\phi) | \phi \in \text{prec}(a_h^R) \setminus \text{prec}(a^R)\}\\ &\cup \{\mathcal{B}(\phi) | \phi \in \text{prec}(a_h^R) \cap \text{prec}(a^R)\}\end{aligned}$$

In any given state, an action in the augmented model is only applicable if the action is executable in the robot model and the human believes the action to be executable. Unlike the executability of the action in the robot model (captured through unconditional preconditions) the human's beliefs about the action executability can be manipulated by turning the meta fluents on and off. The effects of these actions can also be defined similarly by conditioning them on the relevant meta fluent. In addition to these task level actions (represented by the set A_τ), we can also define explanatory actions (A_μ) for each meta-fluent in the set F_μ. Now, for the meta fluents that correspond to adding new information to the human model (i.e., the $\mu^+(*)$ fluents), the fluent will be part of the add effects of the corresponding explanatory action in A_μ. Similarly, for $\mu^-(*)$ fluents, the meta fluent will be part of the delete effects. The explanatory action definition will be empty except for the effect corresponding to the meta-fluent. Special actions a_0 and a_∞ that are responsible for setting all the initial state conditions true and checking the goal conditions are also added to the domain model. These two actions will allow us to simplify notations and proofs by employing a uniform mechanism to capture the effect of all explanatory actions. a_0 has a single precondition that checks for $\mathcal{I}$ and has the following effects:

$$\text{adds}(a_0) = \{\bot \rightarrow p \mid p \in I^R\} \cup \{\bot \rightarrow \mathcal{B}(p) \mid p \in I_h^R\} \cup \{\bot \rightarrow p \mid p \in F_{\mu^-}\}$$

$$\text{dels}(a_0) = \{\mathcal{I}\}$$

where F_{μ^-} is the subset of F_μ that consists of all the fluents of the form $\mu^-(*)$. Similarly, the precondition of action a_∞ is set using the original goal and adds the special goal proposition $\mathcal{G}$.

$$\text{prec}^{a_\infty} = G^R \cup \{\mu^+(p^G) \rightarrow \mathcal{B}(p) \mid p \in G^R \setminus G_h^R\} \cup \{\mu^-(p^G) \rightarrow \mathcal{B}(p) \mid p \in G_h^R \setminus G^R\} \cup \{\mathcal{B}(p) \mid G_h^R \cap G^R\}$$

Finally the new initial state and the goal specification becomes $I_\mathcal{E} = \{\mathcal{I}\}$ and $G_\mathcal{E} = \{\mathcal{G}\}$ respectively. To see how such a compilation would look in practice, consider an action (move_from p1 p2) that allows the robot to move from p1 to p2 only if the path is clear. The action is defined as follows in the robot model:

```
(:action move_from_p1_p2
    :precondition (and (at_p1) (clear_p1_p2))
    :effect (and (not (at_p1)) (at_p2) ))
```

Let us assume the human is aware of this action but does not care about the status of the path (as they assume the robot can move through any debris-filled path). Thus the corresponding action definition in the human model would be

```
(:action move_from_p1_p2
    :precondition (and (at_p1))
    :effect (and (not (at_p1)) (at_p2) ))
```

In this case, the action in the augmented model and the relevant explanatory action will be:

```
(:action move_from_p1_p2
  :precondition (and (at_p1) (𝓑((at_p1))) (clear_p1_p2)
            (implies (μ+((clear_p1_p2),
                           prec(move_from_p1_p2)))
                              (𝓑((clear_p1_p2)))))
  :effect (and (not (at_p1)) (at_p2)
               (not 𝓑(at_p1)) 𝓑(at_p2))
)
```

```
(:action explain_$\mu^{+}_{prec}$_move_from_clear
   :precondition (and)
   :effect (and $\mu^{+}_{prec}$(clear_p1_p2, move_from_p1_p2)
   )
)
```

Note how, in the compiled model, we allow for the fact that human would consider `(clear_p1_p2)` to be part of the precondition (captured using the fluent $\mathcal{B}$`(clear_p1_p2)`), if and only if, the fluent μ^{+}_{prec}`(clear_p1_p2, move_from_p1_p2)` is true. This fluent is only made true by the specified explanatory action.

Finally, C_Ψ captures the cost of all explanatory and task-level actions. For now, the cost of task-level actions is set to the original action cost in the robot model, and the explanatory action costs are set according to C_E. Later, we will discuss how we can adjust the explanatory action costs to generate desired behavior. Additionally, we will set the cost of the actions a_0 and a_∞ to zero.

We will refer to an augmented model that contains an explanatory action for each possible model update and has no actions with effects on both the human's mental model and the task level states as the *canonical augmented model*. Given an augmented model, let $\pi_{\mathcal{E}}$ be a plan that is valid for this model ($\pi_{\mathcal{E}}(I^\Psi) \subseteq G^\Psi$). From $\pi_{\mathcal{E}}$, we extract two types of information – the model updates induced by the actions in the plan and the sequence of actions that have some effect on the task state. The former is, represented as $\mathcal{E}(\pi_{\mathcal{E}}) = \langle \mathcal{E}^{+}(\pi_{\mathcal{E}}), \mathcal{E}^{-}(\pi_{\mathcal{E}}) \rangle$, where $\mathcal{E}^{+}(\pi_{\mathcal{E}}) \subseteq \Gamma(\mathcal{M}^R)$ and $\mathcal{E}^{-}(\pi_{\mathcal{E}}) \subseteq \Gamma(\mathcal{M}_h^R)$. The latter is represented as $\mathcal{D}(\pi_{\mathcal{E}})$) and we refer to the output of $\mathcal{D}$ as the task level fragment of the original plan $\pi_{\mathcal{E}}$ (we assume that $\mathcal{D}(\pi_{\mathcal{E}})$ exclude a_0 and a_∞). $\mathcal{E}(\pi_{\mathcal{E}})$ can directly be identified from the explanatory actions that are part of the plan, but in more general cases it can also contain effects from action in $\mathcal{D}(\pi_{\mathcal{E}})$.

Under this compilation, the planner can automatically find positions of the explanatory actions, but to avoid any confusion that may arise from belief revisions on the users' end, we can enforce some common sense ordering, like making any explanation related to an action appear before the first instance of that action. This ordering will make sure that users are not confused about earlier action effects and also helps reduce branching, making planning more efficient. Such ordering can be enforced by adding some additional book-keeping variables. We also employ additional bookkeeping to ensure that a_0 is applied before any other actions (and is always required for an action sequence to be valid). We will avoid explicitly specifying such variables and bookkeeping in the compilation to simplify the notation. With this compilation, the execution of a valid action sequence π in the $\mathcal{M}^\Psi$ will result in a state that captures both the result of executing the action sequence in the robot and updated human models, more formally we can state it as a proposition of the form:

Proposition 7. *For a valid action sequence π for the augmented model $\mathcal{M}^\Psi$ of a given EAP problem $\Psi = \langle \mathcal{M}^R, \mathcal{M}_h^R \rangle$, we can see that*

1. $\delta_{\mathcal{M}^R}(I^R, \mathcal{D}(\pi)) = (\delta_{\mathcal{M}^\Psi}(I^\Psi, \pi) \cap F)$, *where F is the original set of fluents.*
2. *Let $\tilde{\mathcal{M}}_h^R = \mathcal{M}_h^R + \mathcal{E}(\pi)$, then for every $f \in \delta_{\tilde{\mathcal{M}}_h^R}(\tilde{I}_h^R, \mathcal{D}(\pi))$, there must exist a $\mathcal{B}(f) \in \delta_{\mathcal{M}^\Psi}(I^\Psi, \pi)$ and vice versa.*

Proof. The proof of the first statement of the proposition follows directly from the fact that the compilation copies all the action definitions from the robot model over to $\mathcal{M}^\Psi$ and the fluents that are part of F are only updated by those parts of the model. As such, the result of executing the task actions and a^0 capture the entirety of how the fluents in F will be updated by any valid action sequence. a^0 merely sets the initial state, and the task actions will make no updates to fluents in F that it would not have made in the robot model. Hence, the first statement $\delta_{\mathcal{M}^R}(I^R, \mathcal{D}(\pi)) = (\delta_{\mathcal{M}^\Psi}(I^\Psi, \pi) \cap F)$ holds.

Now, coming to the second statement. The first point we need to show is that if π is valid for $\mathcal{M}^\Psi$ then $\mathcal{D}(\pi)$ must be valid for $\tilde{\mathcal{M}}_h^R = \mathcal{M}_h^R + \mathcal{E}(\pi)$. Now, each action definition in the human model is mapped over to $\mathcal{M}^\Psi$, in terms of $\mathcal{B}(\cdot)$ variables. Firstly, there must be preconditions that are part of both models and hence are not conditioned on any meta fluents. Thus they had to be satisfied by $\mathcal{B}(\cdot)$ effects of the previous actions in the plan. Now the rest of the preconditions will need to be satisfied based on whether the corresponding $\mu^{+/-}(\cdot)$ variables are added or deleted. If they were, the corresponding changes would also be reflected in $\mathcal{E}(\pi)$. Thus, if $\mathcal{E}(\pi)$ adds a new precondition to an action that is part of $\mathcal{D}(\pi)$, then the corresponding $\mathcal{B}(\cdot)$ should have held as part of the execution of π in $\mathcal{M}^\Psi$. This also holds true for preconditions removed by the explanatory actions and hence also removed from $\tilde{\mathcal{M}}_h^R$. The validity of the plan also relies on a symmetric argument holding for the effects of these actions. This symmetric argument for the effects also implies that the effect of executing these actions with respect to the $\mathcal{B}$ fluents will mirror the execution of the actions in the model $\tilde{\mathcal{M}}_h^R$. This completes the proof for the second statement of the propositions. Which in turn proves the proposition as a whole. □

This brings us to our first theorem.

Theorem 1. *For a given EAP problem $\Psi = \langle \mathcal{M}^R, \mathcal{M}_h^R \rangle$ the corresponding augmented model $\mathcal{M}^\Psi$ is a sound and complete formulation, i.e.: (1) for every valid π for $\mathcal{M}^\Psi$ the tuple $\langle \mathcal{D}(\pi), \mathcal{E}(\pi) \rangle$ is a valid solution for Ψ and (2) for every valid solution $\langle \pi, \mathcal{E}^\Psi \rangle$, there exists a corresponding valid plan for π' for $\mathcal{M}^\Psi$ such that $\mathcal{D}(\pi') = \pi$ and $\mathcal{E}(\pi') = \mathcal{E}^\Psi$.*

Proof. The first condition, i.e., (1) for every valid π for $\mathcal{M}^\Psi$ the tuple $\langle \mathcal{D}(\pi), \mathcal{E}(\pi) \rangle$ is a valid solution for Ψ, corresponds to the soundness of the formulation. This condition directly follows from Proposition 7 and the definition of a balanced solution (i.e., Definition 5). The proposition ensures that $\mathcal{D}(\pi)$ is valid for the robot model and for the updated human model. This is, in fact, the requirements for a balanced solution laid out in Definition 5.

The second condition corresponds to the completeness of the compilation. To see why the formulation is complete, consider a solution $< \pi, \mathcal{E}^{\Psi} >$ for Ψ. Now we can directly construct a plan π for $\mathcal{M}^{\Psi}$. From the procedure for constructing $\mathcal{M}^{\Psi}$, we know that there must exist an explanatory action for each possible model difference. This means that there should exist a sequence of explanatory actions $\langle a_1 .., a_k \rangle$ that results in the same model updates captured by $\mathcal{E}^{\Psi}$. Now consider the plan $\pi^{\Psi} = a_0 + \langle a_1, .., a_k \rangle + \pi + a_{\infty}$. For $< \pi, \mathcal{E}^{\Psi} >$ to be a balanced solution for Ψ, then π must be executable in $\mathcal{M}^R$ and $\mathcal{M}^R_h + \mathcal{E}^{\Psi}$. For each action in $\mathcal{D}(\pi^{\Psi})$, the Proposition 7, tells us that the state obtained contains the union of the state resulting from executing the actions in model $\mathcal{M}^R$ and the copy of the state resulting from executing the plan in the $\mathcal{M}^R_h$ but expressed in terms of the belief fluents. Since the preconditions of the actions expressed in terms of fluents F are the same as that in $\mathcal{M}^R$, and the plan is valid in $\mathcal{M}^R$, the preconditions must hold in the resultant state for $\delta_{\mathcal{M}^{\Psi}}(I^{\Psi}, \pi^{\Psi})$. Similarly, for the preconditions of the actions expressed in terms of belief fluents, the compilation ensures that the precondition is only checked if it is part of the $\mathcal{M}^R_h + \mathcal{E}^{\Psi}$. As per, Proposition 7, there is a one-to-one correspondence between the valuation of belief fluents in $\delta_{\mathcal{M}^{\Psi}}(I^{\Psi}, \pi^{\Psi})$ and the execution of the plan π in $\mathcal{M}^R_h + \mathcal{E}^{\Psi}$. As such if the plan is valid in $\mathcal{M}^R_h + \mathcal{E}^{\Psi}$, every action in $\mathcal{D}(\pi^{\Psi})$ must be executable in $\mathcal{M}^{\Psi}$. We can extend this line of reasoning to goal conditions for $\mathcal{M}^R$ and $\mathcal{M}^R_h + \mathcal{E}^{\Psi}$ and see that π^{Ψ} is a valid plan for $\mathcal{M}^{\Psi}$. This proves both statements of our theorem, and hence we prove the theorem as a whole. □

With the compilation in place, we can now look at establishing the complexity of generating valid balanced plans $(\pi, \mathcal{E})$ for a given EAP problem such that the plan π is valid in both the robot model and the updated human model.

Theorem 2. *For a given EAP problem $\Psi = \langle \mathcal{M}^R, \mathcal{M}^R_h \rangle$, where both $\mathcal{M}^R$ and $\mathcal{M}^R_h$ are represented as classical planning problems, whether a valid balanced plan exists (BALANCED-EXIST) for Ψ is PSPACE-complete.*

Proof. We establish the PSPACE-hardness of BALANCED-EXIST by showing that the problem of whether a valid plan exists for a given model (PLAN-EXIST) can be transformed into an EAP. Specifically, consider a planning model $\mathcal{M}$ as defined in Section 2.1. Here, PLAN-EXIST is PSPACE-hard [18]. Now, let us create an EAP problem $\Psi = \langle \mathcal{M}, \mathcal{M} \rangle$. Per our definition, all valid balance solutions must have a model update part $\mathcal{E} = \langle \mathcal{E}^+, \mathcal{E}^- \rangle$, $\mathcal{E}^+ \subseteq \Gamma(\mathcal{M}) \setminus \Gamma(\mathcal{M})$, and $\mathcal{E}^- \subseteq \Gamma(\mathcal{M}) \setminus \Gamma(\mathcal{M})$. This, in turn, means that we can only have a balanced solution where the model information is empty. As such, any plan that is part of a valid balanced plan must be executable in $\mathcal{M}$ and any plan that is executable in $\mathcal{M}$ must correspond to a valid balance solution for Ψ. As such, PLAN-EXIST is true for the model $\mathcal{M}$ if and only if BALANCED-EXIST is true for Ψ. Additionally, the transformation from $\mathcal{M}$ to Ψ is a polynomial one. This completes the proof for PSPACE-hardness of BALANCED-EXIST.

We can establish membership in the PSPACE class by showing that there exists a sound and complete compilation from creating balanced plans for EAP to a planning problem with conditional effects and disjunctive/negative preconditions that are linear in the size of the original planning problems. Per the results established by [18], the problem of plan existence is still in PSPACE for this class of planning problems. This part is already proved by Theorem 1. Thus proving that BALANCED-EXIST is PSPACE complete. □

It is worth noting that in most cases, we need to go beyond just generating valid plans and talk about generating cost-minimizing and even optimal solutions.

4.2. Stage of interaction and epistemic side effects

One important aspect we have yet to discuss is whether the explanation is meant for a plan that is proposed by the system (i.e. the system presents a sequence of actions to the user) or are we explaining some plan that is being executed either in the real world or some simulation the user (observer) has access to. Even though the above formulation can be directly used for both scenarios, we can use the fact that the human is observing the execution of the plans to simplify the explanatory behavior by leveraging the fact that many of these actions may have epistemic side effects. This allows us to not explain any of the effects of the actions that the human can observe (for those effects we can directly update the believed value of the corresponding state fluent and the meta-fluent).[9] This is beyond the capability of any of the existing algorithms in this space of the explicability-explanation dichotomy.

This consideration also allows for the incorporation of more complicated epistemic side-effects wherein the user may infer facts about the task that may not be directly tied to the effects of actions. Such effects may be specified by domain experts or generated using heuristics. Once identified, adding them to the model is relatively straightforward as we can directly add the corresponding meta fluent into the effects of the relevant action. An example of a simple heuristic would be to assume that the firing of a conditional effect results in the human believing the condition to be true. For example, if we assume that the robot had an action (open_door_d1_p3) that had a conditional effect:

```
(when (and (unlocked_d1)) (open_d1))
```

Then, in the compiled model, we can add a new effect:

[9] This means that when the plan is being executed, the problem definition should include the observation model of the human (which we assume to be deterministic). To keep the formulation simple, we ignore this for now. Including this additional consideration is straightforward for deterministic sensor models.

```
(when (and (unlocked_d1))
      (and 𝓑(open_d1) 𝓑(unlocked_d1)))
```

Even in this simple case, it may be useful to restrict the rule to cases where the effect is conditioned on previously unused fluents so the robot does not expect the observer to be capable of regressing over the entire plan.

4.3. Optimality of the agent

The compilation explored so far only takes into consideration the expectations the agent has about the safety of the plans (i.e. the user would expect any plans generated to be valid and executable) and does not account for the user's expectation on whether the agent should act optimally. To better understand this, let us go back to the USAR domain (Fig. 2). The robot is now at P1, needing to travel to its goal at P17. The optimal plan expected by the commander is highlighted in grey, and the robot optimal plan is highlighted in blue. Here, if the agent just followed the plan that takes the robot through P5 and P6 with a clear_passage_P5_P6 action with no additional explanatory actions, then the user may still be confused as to why the agent does not just follow the grey plan that involves going through P16 to P17.

Even in cases where the action costs are the same for the agent and the human, we cannot account for such expectations by merely generating optimal plans in the augmented model. For example, the optimal plan (blue) in the augmented model would be the one through P2 and P3 with one extra explanatory action explain_μ_I^+_clear_P2_P3. While this provides an explanation to ensure validity, ensuring the optimality would require the agent to also explain that the passage from P16 to P17 is blocked, which would clearly be more expensive than choosing the valid plan for any non-zero cost for explanatory actions.

One approach to address this would be to prune all solutions where the task level fragment of the plan ($\mathcal{D}(\pi)$) is suboptimal in the updated human model. A simple way to enforce this would be to extend the planner to perform an optimality test for the current plan during the goal test (i.e., is the plan optimal in the updated model, thereby corresponding to a balanced explicable plan). It may be possible to use more intelligent pruning to reduce the number of goal tests (e.g. the optimality test never needs to be repeated for the same set of model updates) and we could design heuristics that take into account optimality aspects. In this paper, we adopt this simple approach as a first step toward modeling these novel behaviors. However, please note that the generation of balanced explicable plans does require us to extend traditional planners.

4.3.1. Balanced plans versus agent optimal plans

Even when generating plans that preserve the user's expectations about agent optimality, the agent could generate two types of plans: agent optimal balanced plans or balanced explicable plans. In the first scheme, the agent chooses to select self-explanatory plans whose task level fragment is going to be optimal in the original agent model and then choose the minimal explanations (MCEs) that justifies the plan that is optimal in the updated mental model. An example would be choosing the plan highlighted in blue in robot model and then explaining that the path from P2 to P3 is clear and P16 to P17 is blocked. In the latter scheme, the agent could choose plans that are easiest to explain (here again we need to ensure that after the explanation the plan is optimal in the updated model). For example, in the USAR scenario if communication is expensive, it may be easier to choose the plan to move through P5 and P6 with a clear passage action since we only need to explain that the passage P16 to P17 is blocked.

In the first case, the agent is effectively prioritizing any loss of optimality over any overhead accrued by communicating the explanation, while in the second case the agent accounts for the cost of both the plan it is performing and the explanation cost (the cost of communication and possibly the computational overhead experienced by the user on receiving the explanation). By assigning explanatory costs to explanatory actions we are essentially generating balanced plans but there may be scenarios where the agent needs to stick to its optimal plan. We can generate such agent optimal plans by setting lower explanatory action costs. Before we formally state the bounds for explanatory costs, let us define the concept of *optimality delta* (denoted as $\Delta\pi_{\mathcal{M}}$) for a planning model, which captures the cost difference between the optimal plan and the second most optimal plan. More formally $\Delta\pi_{\mathcal{M}}$ can be specified as:

$$\Delta\pi_{\mathcal{M}} = C^*_{\mathcal{M}} - C'_{\mathcal{M}}$$

Where $C'_{\mathcal{M}} = 0$, if $\forall \pi \in \Pi_{\mathcal{M}}, C(\pi) = C^*_{\mathcal{M}}$, else $C'_{\mathcal{M}} = C(\pi')$, where $C(\pi') > C^*_{\mathcal{M}}$ and there exists no plan $\hat{\pi}$, such that $C(\pi') < C(\hat{\pi}) < C^*_{\mathcal{M}}$.

Theorem 3. *In a canonical augmented model $\mathcal{M}^\Psi$ for an EAP planning problem Ψ, if the sum of costs of all explanatory actions is $< \Delta\pi_{\mathcal{M}^R}$ and if π is the cheapest valid plan for $\mathcal{M}^\Psi$ such that $\mathcal{D}(\pi) \in \Pi^*_{\mathcal{M}_h^R + \mathcal{E}(\pi)}$, then:*

(1) *$\mathcal{D}(\pi)$ is optimal for $\mathcal{M}^R$*
(2) *$\mathcal{E}(\pi)$ is the MCE for $\mathcal{D}(\pi)$*
(3) *There exists no plan $\hat{\pi} \in \Pi^*_{\mathcal{M}^\Psi}$ such that MCE for $\mathcal{D}(\hat{\pi})$ is cheaper than $\mathcal{E}(\pi)$, i.e., the search will find the plan with the smallest MCE.*

Proof. We start by observing that by the framing of our theorem statement, there exists no valid plan π' for the augmented model ($\mathcal{M}^\Psi$) with a cost lower than that of π and where the task level fragment ($\mathcal{D}(\pi')$) is optimal for the updated human model. We will prove condition one by contradiction; as such let's assume that the statement doesn't hold and thus $\mathcal{D}(\pi) \notin \Pi^*_{\mathcal{M}^R}$ (i.e., the current plan's task-level fragment is not optimal in the robot model) and let $\hat{\pi} \in \Pi^*_{\mathcal{M}^R}$. Let's construct a plan $\hat{\pi}_{\mathcal{E}}$ for the augmented model that corresponds to the plan $\hat{\pi}$, i.e., $\mathcal{E}(\hat{\pi}_{\mathcal{E}})$ is the MCE for the plan $\hat{\pi}$ and $\mathcal{D}(\hat{\pi}_{\mathcal{E}}) = \hat{\pi}$. The given augmented plan $\hat{\pi}_{\mathcal{E}}$ is a valid solution for our augmented planning problem $\mathcal{M}^\Psi$ that satisfies the condition that its task-level component is an optimal plan for $\mathcal{M}^R_h + \mathcal{E}(\pi)$. The former follows from Theorem 2, and the latter follows from the fact that $\hat{\pi}_{\mathcal{E}}$ consists of the MCE for $\hat{\pi}$, and by definition, MCEs ensure the optimality of the plan in the updated human model. Since $\hat{\pi}$ is optimal for $\mathcal{M}^R$ and $\mathcal{D}(\pi)$ is not, we have $C_{\mathcal{M}^R}(\hat{\pi}) < C_{\mathcal{M}^R}(\mathcal{D}(\pi))$. From the definition of optimality gap, we know $C_{\mathcal{M}^R}(\hat{\pi}) + \Delta\pi_{\mathcal{M}^R} \leq C_{\mathcal{M}^R}(\mathcal{D}(\pi))$. By leveraging the facts that the costs of task-level actions are the same as the robot model, and the total cost of explanatory actions in a plan cannot equal or exceed $\Delta\pi_{\mathcal{M}^R}$, we can see that $C_\Psi(\hat{\pi}_{\mathcal{E}}) < C_{\mathcal{M}^R}(\hat{\pi}) + \Delta\pi_{\mathcal{M}^R} \leq C_{\mathcal{M}^R}(\mathcal{D}(\pi)) \leq C_\Psi(\pi)$. This violates the premise of our theorem, hence proving the first statement by contradiction.

Moving to the second statement, we will leverage the fact that $\mathcal{D}(\pi)$ is an optimal plan in model $\mathcal{M}^R$. We know that per Theorem 2, one can construct a valid plan for $\mathcal{M}_\Psi$ by adding explanatory actions corresponding to the MCE for the given plan. Since MCE is the minimal cost explanation for $\mathcal{D}(\pi)$ if $\mathcal{E}(\pi)$ was not the MCE, the cost of the new plan would be lower than C_Ψ which would again violate the premise of our theorem. Thus proving our second statement.

For our final statement, we will follow a similar logic. Again, let's assume there is another plan, π', that is optimal for $\mathcal{M}^R$ but has an MCE $\mathcal{E}'$ with a lower cost. Let $\hat{\pi}'$, be a plan for $\mathcal{M}_\Psi$ such that $\mathcal{D}(\hat{\pi}') = \pi'$ and $\mathcal{E}(\hat{\pi}') = \mathcal{E}'$. This is a valid solution for $\mathcal{M}_\Psi$ per Theorem 2. $C_{\mathcal{M}^R}(\mathcal{D}(\hat{\pi}')) = C_{\mathcal{M}^R}(\mathcal{D}(\pi))$ (both are optimal for $\mathcal{M}^R$), and per our assumption the cost of explanatory actions in $\hat{\pi}'$ must be lower than the one for π. This means that $C_{\mathcal{M}_\Psi}(\hat{\pi}') < C_{\mathcal{M}_\Psi}(\pi)$. Which again violates our premise, hence proving our statement by contradiction. □

Note that while it is hard to find the exact value of the optimality $\Delta\pi_{\mathcal{M}}$, it is guaranteed to be ≥ 1 for domains with only unit cost actions or $\geq (C_2 - C_1)$, where C_1 is the cost of the cheapest action and C_2 is the cost of the second cheapest action, i.e. $\forall a(C_{\mathcal{M}}(a) < C_2 \rightarrow C_{\mathcal{M}}(a) = C_1)$. Thus allowing us to easily scale the cost of the explanatory actions to meet this criteria.

4.3.2. Disallowing explicable plans that are too costly

There could be scenarios where forcing the agent to always choose plans that are optimal in the human model may not be the best strategy. There may be cases where we would prefer the agent to deviate from the optimal expected plan if it results in significant gains. The penalty for deviating from the expected optimal plan should thus be another optimization criteria. This means allowing for the generation of optimal balanced plans. Which in this case corresponds to

$$C(\pi) + \beta * (C_{\mathcal{M}^R_h + \mathcal{E}(\pi)}(\pi) - C^*_{\mathcal{M}^R_h + \mathcal{E}(\pi)})$$

where the second term corresponds to the penalty associated with the agent following an inexplicable plan $C_{IE}(\pi, \mathcal{M}^R_h)$.[10]

Definition 6. The solution $\langle \mathcal{E}^\Psi, \pi^\Psi \rangle$ to Ψ is optimally balanced if –

1. $\pi^\Psi(I^R) \models_{\mathcal{M}^R} G^R$.
2. $\pi^\Psi(\bar{I}) \models_{\bar{\mathcal{M}}} G^R_h$, where $\bar{\mathcal{M}} \leftarrow \mathcal{M}^R + \mathcal{E}$.
3. $\nexists \langle \hat{\mathcal{E}}, \hat{\pi} \rangle$ satisfying (1) and (2) and $\alpha \times C(\hat{\mathcal{E}}) + C_{\mathcal{M}^R}(\hat{\pi}) + \beta \times (C_{\mathcal{M}^R_h + \hat{\mathcal{E}}}(\hat{\pi}) - C^*_{\mathcal{M}^R_h + \hat{\mathcal{E}}}) < \alpha \times C(\mathcal{E}^\Psi) + C_{\mathcal{M}^R}(\pi^\Psi) + \beta \times (C_{\mathcal{M}^R_h + \mathcal{E}^\Psi}(\pi^\Psi) - C^*_{\mathcal{M}^R_h + \mathcal{E}^\Psi})$.

To generate such optimal balanced plans, we need to relax the goal requirement that the final plan is optimal in the human model. We can incorporate the inexplicability penalty into the reasoning about the plan by assigning the cost of a_∞ (the goal action) to be β times the cost difference between the optimal plan in the human model and the current plan. When β is set to zero, the problem would just identify the cheapest plan in the original robot model that is executable in the human model. However, such dynamic action cost assignment again requires us to look beyond standard planners. We can also use this formulation to generate plans that are still guaranteed to be optimal in the human model by setting α higher than a threshold κ, where κ is some upper bound on plan length for the robot (that includes explanatory actions).

5. Empirical results

We will now provide evaluations of `MEGA-approx` (empirical evaluation will be exclusively focusing on `MEGA-approx` instead of `MEGA`) demonstrating the trade-off in the cost and computation time of plans for varying size of the model difference and the hyper-parameter α. We will then report on human factor studies on how users receive this trade-off. Note that the two flavors of evaluations are done with different motivations. The former evaluates the explicability-explanation trade-off from the robot's perspective, which can minimize communication and the penalty due to explicability. The user study instead evaluates the effect of this on the human. Experiments in Section 5.1 used the FastDownward [19] as the underlying planner (specifically we used A* search and LM-CUT

[10] The term $\alpha * \mathcal{E}(\pi)$ is already expected to be folded into the term $C(\pi)$ and reflected in the cost of the communication actions.

heuristic [20]). All other experiments used a planner developed by the authors that use hmax [21] as the underlying heuristic along with A* search.[11] We did not impose any memory limit on any of the instances. All results reported here are from experiments run on a 12-core Intel(R) Xeon(R) CPU with an E5-2643 v3@3.40 GHz processor and 64 GB RAM.

5.1. Illustration of the cost trade-off in balanced plans

The hyperparameter α determines how much an agent is willing to sacrifice optimality versus the explanation cost. We will illustrate this trade-off on modified versions of two popular IPC[12] domains. All problem instances were picked from the original problem set; the human domain corresponds to the original domain description, and the robot one has been updated as follows.

The rover (Meets a Martian) domain Here the IPC Mars Rover has a model as described in the IPC domain, but has undergone an update whereby it can carry the rock and soil samples needed for a mission at the same time. This means that it does not need to empty the store (via `drop_off` action) before collecting new rock and soil samples anymore so that the new action definitions for `sample_soil` and `sample_rock` no longer contain the precondition `(empty ?s)`.

During its mission it runs across a Martian, a poor astronaut who was earlier sent to start the colonization process (a la Matt Damon, from eponymous movie). Our Martian is unaware of the robot's expanded storage capacity, and has an older, extremely cautious, model of the rover it has learned while spying on it from its cave. They believe that any time the Rover collects a rock sample, it also needs to collect a soil sample and need to communicate this information to the lander. The Martian also believes that before the rover can perform `take_image` action, it needs to send the soil data and rock data of the waypoint from where it is taking the image. Clearly, if the rover was to follow this model, in order not to spook the Martian it will end up spending a lot of time performing unnecessary actions (like dropping old samples and collecting unnecessary samples) – e.g. if the rover is to communicate an image of an objective `objective2`, all it needs to do is move to a waypoint (`waypoint3`) from where `objective2` is visible and perform the action:

```
(take_image waypoint3 objective2 camera0 high_res)
```

If the rover was to produce a plan that better represents the Martian's expectations, it would look like:

```
(sample_soil store waypoint3)
(communicate_soil_data waypoint3 waypoint3 waypoint0)
(drop_off store)
(sample_rock store waypoint3)
(communicate_rock_data waypoint3 waypoint3 waypoint0)
(take_image waypoint3 objective1 camera0 high_res)
```

If the rover uses an MCE here, it ends up explaining 6 model differences. In some cases, this may be acceptable, but in others, it may make more sense for the rover to bear the extra cost rather than laboriously walk through all updates with an impatient Martian. Fig. 5 shows how the explicability and explanation costs vary for problem instances and α values in this domain. Here, the explicability cost refers to the extra cost the robot bears as compared to its optimal plan. The different α values we tested are marked on the graphs for each problem. The algorithm converges to the smallest possible MCE, when α is set to 1. For smaller α, `MEGA-approx` saves explanation cost by choosing more explicable (and expensive) plans.

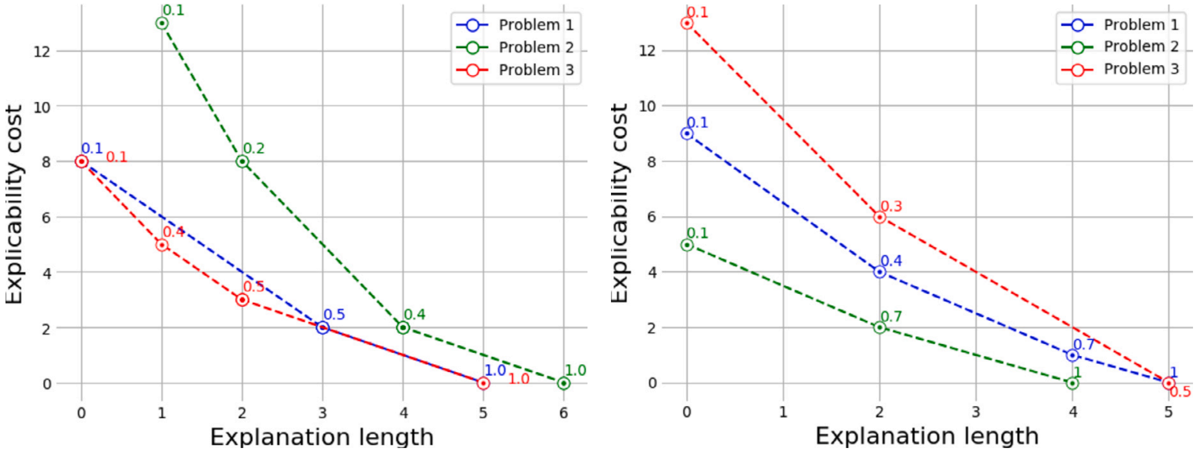
The Barman (in a Bar) domain The traditional Barman captures a robot preparing various drinks using glasses, shakers, and drink dispensers. Usually, the robot is constrained by the fact that the robot can only grasp one object at a time. Here, the brand new two-handed Barman robot is wowing onlookers with its single-handed skills, even as its admirers who may be unsure of its capabilities expect, much like the standard IPC domain, that it needs one hand free for actions like `fill-shot`, `refill-shot`, `shake` etc. This means that to make a single shot of a cocktail with two shots of the same ingredient with three shots and one shaker, the human expects the robot to:

```
(fill-shot shot2 ingredient2 left right dispenser2)
(pour-shot-to-used-shaker shot2 ingredient3 shaker1 left)
(refill-shot shot2 ingredient3 left right dispenser3)
(pour-shot-to-used-shaker shot2 ingredient3 shaker1 left)
(leave left shot2)
(grasp left shaker1)
```

The robot can, however, directly start by picking both the shot and the shaker and does not need to put either of them down while making the cocktail. Similar to the Rover domain, we again illustrate (Fig. 5) how at lower values of α the robot generates plans that

[11] Code can be found at http://bit.ly/2Xb7OCp.

[12] From the International Planning Competition (IPC) 2011. http://www.plg.inf.uc3m.es/ipc2011-learning/Domains.html.



(a) The Rover (Meets a Martian) Domain

(b) The Barman (in a Bar) Domain

Fig. 5. The explicability versus explanation costs with respect to α.

Table 1
Runtime (sec) versus the size of explanations $\mathcal{E}$ with respect to the size of the model difference $| \mathcal{M}^R \Delta \mathcal{M}_h^R |$.

		$\mid \mathcal{M}^R \Delta \mathcal{M}_h^R \mid = 2$		$\mid \mathcal{M}^R \Delta \mathcal{M}_h^R \mid = 7$		$\mid \mathcal{M}^R \Delta \mathcal{M}_h^R \mid = 10$	
		$\mid\mathcal{E}\mid$	Time	$\mid\mathcal{E}\mid$	Time	$\mid\mathcal{E}\mid$	Time
Rover	p1	0	1.22	1	5.83	3	143.84
	p2	1	1.79	5	125.64	6	1061.82
	p3	0	8.35	2	10.46	3	53.22
Barman	p1	2	18.70	6	163.94	6	5576.06
	p2	2	2.43	4	57.83	6	953.47
	p3	2	45.32	5	4183.55	6	5061.50

require less explanation. As α increases the algorithm produces plans that require larger explanations with the explanations finally converging at the smallest MCE required for that problem.

Table 1 illustrates how the length of explanations computed compares with the total model difference $| \mathcal{M}^R \Delta \mathcal{M}_h^R |$. The human models considered for this table were created by randomly deleting a subset of model components from the original domain problem files. We used an α value of 1. Clearly, there are significant gains to be had in terms of the minimality of explanations and the reduction in the cost of explicable plans as a result of it. This is something the robot trades off internally by considering its limits of communication, cost model, etc. We will discuss the external effect of this (on the human) later in the human factors study.

5.2. Illustration of balancing in self-explaining plans

We will now illustrate this balancing behavior in self-explaining plans. In Fig. 2, if the robot were to follow the explanation scheme established in [2], it would stick to its own plan and provide the following explanation:

```
remove-(clear p16 p17)-from-I <-- Path from P16 to P17 is blocked
add-(clear p2 p3)-to-I <-- i.e. Path from P2 to P3 is clear
```

If the robot were to stick to a purely explicable plan [3] then it can choose to use the passage through P5 and P6 after performing a costly `clear_passage` action (this plan is not optimal in either of the models).

In the epistemic formulation, the action for opening a door has an epistemic side effect that the observer would know that the door is unlocked. We start by assigning a cost of 10 to every robot action other than clear-rubble action (which is 50) and the move-through-door action (set to 20). We set the cost of communication action to 1 to start with. The solution produced corresponds to the blue plan in Fig. 2.

```
>> explains_$\mu^+_{init}$_clear_p2_p3
>> explains_$\mu^-_{init}$_clear_p16_p17
>> move_p1_p2
>> move_p2_p3
>> move_p3_p4
>> move_p4_p11
>> move_p11_p13
>> move_p13_p14
>> move_p14_p18
>> move_p18_p17
```

This plan includes the *optimal robot plan and corresponding minimal explanation*. Now if we were to set the cost of communication actions to 100 (C_E in the compilation), we see the agent deviating to plans which on their own may not be optimal – e.g. a plan that involves opening the door at P8:

```
>> explains_$\mu^{-}_{init}$_clear_p16_p17
>> move_p1_p7
>> move_p7_p8
>> opendoor_p8_d1
>> movethroughdoor_p8_p9_d1
>> move_p9_p10
>> move_p10_p13
>> move_p13_p14
>> move_p14_p18
>> move_p18_p17
```

Here the robot does not have to explicitly provide a separate explanation for the status of the door, but still needs to explain that the path from P15 to P16 is blocked. Note that this plan is an example of **a balanced plan** that leverages **epistemic side effects**. Now we go one step further and relax the need to assure optimality of the plan in the human model by changing it from a hard constraint to just a penalty. This gets us the exact same plan as above but without the explanation about the blocked corridor from P15 to P16, thus allowing a notion of soft explicability.

5.3. Computation time: MEGA-approx versus planning compilation

Contrary to classical notions of planning that occurs in state or plan space, we are now planning in the model space, i.e. every node in the search tree is a new planning problem. As seen in Table 1, this can be time consuming (even for the approximate version) with increasing number of model differences between the human and the robot, even as there are significant gains to be had in terms of minimality of explanations, and the reduction in cost of explicable plans as a result of it. MEGA-approx remains comparable with the original work on model reconciliation [2] which also employs model space search, though we are solving a harder problem (computing the plan in addition to its explanation). Interestingly, in contrast to [2], the time taken here (while still within the bounds of the IPC Optimal Track) is *conceded at planning time rather than at explanation time*, so the user does not have to actually ask for an explanation and wait.

An interested reader may also refer to existing works on model space search [2,22,23] which introduces heuristics and approximations which are equally applicable here and can considerably speed up the process. However, the focus of this paper is instead on the interesting behaviors that emerge from considering explanations during the plan generation process.

More importantly, in addition to providing a more general way to capture novel balancing behaviors, our unified approach also lends itself to reusing established heuristics for state space search from the compilation to classical planning. To recap, MEGA-approx identifies only one optimal plan per search node and the search ends as soon as it finds a node where the optimal plan produced has the same cost as the robot plan and is executable in the robot model. This means all the solutions we generate from the planning compilation are guaranteed to be better in terms of cost than that generated by the other. Moreover, there are computational gains to be had. For comparison, similar to the Rover and Barman examples used above for illustration, we selected five IPC domains and for each domain, we created three unique models by introducing 10 random updates in the model, except in the case of Gripper and Driverlog where only 5 were removed. Each of these five domains were paired with five problem instances and then tested on each of the possible configurations. Each instance was run with a limit of one hour; all explanatory actions were restricted to the beginning of the plan and the cost of explanatory actions were set to be twice the cost of original action. For MEGA-approx we used an α value of one.

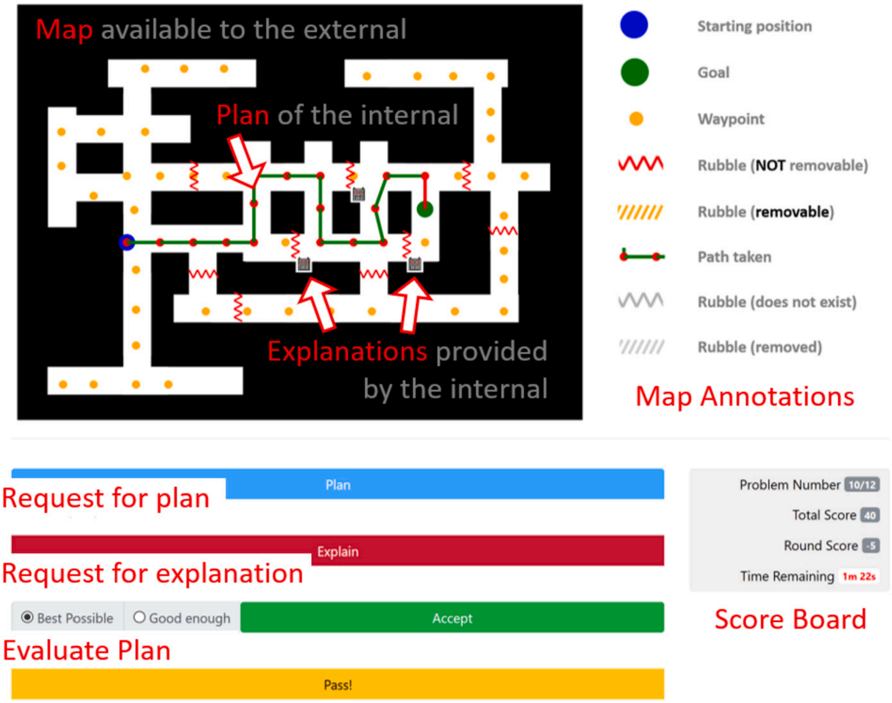
Table 2 lists the time taken to solve each of these problems. For calculating the average runtime, we used 3600 secs as the stand in for the runtime of all the instances that timed out. We used *h_max* (admissible) as the heuristic for all the configurations. The table shows that the compilation does better than the model space search for generating balanced plans for most of the domains. Gripper seems to be the only domain, where model search seems to perform slightly better but this is also a domain that had the smallest number of model differences. This indicates that the ability to leverage planning heuristics can make a marked difference in domains with a large number of explanatory actions.

6. Results from a human subject study

In this section, we will use the USAR domain introduced before to analyze how participants in a user study responded to the explicability versus explanations trade-off. The experimental setup (reproduced here in part of clarity) derives from those used to study the broader details of the model reconciliation process in [24]. Specifically, we set out to test two key hypothesis –

Table 2
Coverage and average runtime (sec) for the planning compilation versus `MEGA-approx`.

	New Compilation		Model Space Search	
	coverage	**runtime**	**coverage**	**runtime**
Blocksworld	13/15	**569.38**	13/15	2318.73
Elevator	**15**/15	**59.20**	1/15	3382.462
Gripper	5/15	2301.90	**6**/15	**2093.54**
Driverlog	**4**/15	**2740.38**	2/15	3158.59
Satellite	**2**/15	**3186.93**	0/15	3600



Problem Number 10/12
Total Score 40
Round Score -5
Time Remaining 1m 22s

Fig. 6. Interface for the external commander in the USAR domain.

H1. Subjects would require explanations when the robot comes up with suboptimal plans in their mental model.

H1a. Response to balanced explicable plans should be indistinguishable from inexplicable / robot optimal plans.

H2. Subjects would require less explanations for explicable plans as opposed to balanced or robot optimal plans.

As we explained before, **H1** is the key thesis of our recent works on explanations [2] that we build on here; it formulates the process of explanation as one of *model reconciliation* to achieve common grounds with respect to a plan's optimality. This forms the basis of incorporating considerations of explanations in the plan generation process as well, as done in the paper, in the event of model differences with the human in the loop. **H2** forms the other side of this coin and completes the motivation of computing balanced plans. Note that balanced plans would still appear suboptimal (and hence inexplicable) to the human even though they afford opportunities to the robot to explain less or perform a more optimal plan. Thus, we expect (**H1a**) their behavior to be identical in case of both robot optimal and balanced plans.

6.1. Experimental setup

The experimental setup exposes the external commander's interface to participants who get to analyze plans in a mock USAR scenario. The participants were incentivized to make sure that the explanation does indeed help them understand the optimality of the plans in question by formulating the interaction in the form of a game. This is to make sure that participants were sufficiently invested in the outcome as well as mimic the high-stakes nature of USAR settings to accurately evaluate the explanations. Fig. 6 shows a screenshot of the interface which displays to each participant an initial map (which they are told may differ from the robot's actual map), the starting point and the goal. A plan is illustrated in the form of a series of paths through various waypoints highlighted on the map. The participant had to identify if the plan shown is optimal. If unsure, they could ask for an explanation. The explanation was provided in the form of a set of changes to the player's map. The player was awarded 50 points for correctly identifying the plan

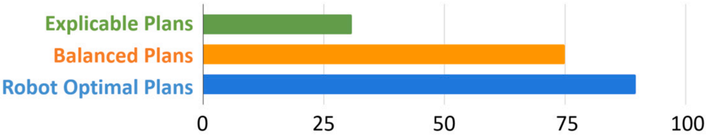


Fig. 7. Percentage of times participants in the study asked for explanations for different types of plans, illustrating reduced demand for explanations for explicable plans with no significant difference for robot optimal and balanced plans.

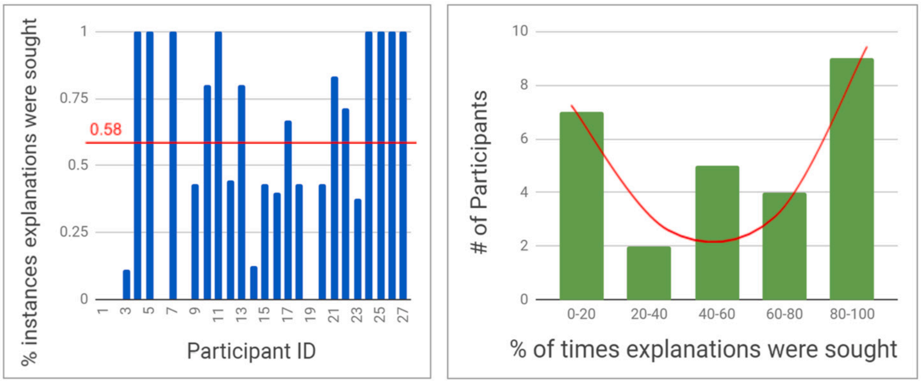


Fig. 8. Click-through rates for explanations.

Table 3
Statistics of explicability vs. explanation trade-off.

Optimal Plan		Balanced Plan		Explicable Plan	
$\lvert\mathcal{E}\rvert$	$C(\pi, \mathcal{M}^R)$	$\lvert\mathcal{E}\rvert$	$C(\pi, \mathcal{M}^R)$	$\lvert\mathcal{E}\rvert$	$C(\pi, \mathcal{M}^R)$
2.5	5.5	1	8.5	-	16

as either optimal or satisficing. Incorrect identification cost them 20 points. Every request for explanation further cost them 5 points, while skipping a map did not result in any penalty. Even though *there were no incorrect plans in the dataset*, the participants were told that selecting an inexecutable plan as either feasible or optimal would result in a penalty of 400 points, in order to deter them from guessing when they were unsure.

Each subject was paid $10 as compensation for their participation and received additional bonuses depending on how well the performed (≤ 240 to ≥ 540 points). This was done to ensure that participants only ask for an explanation when they are unsure about the quality of the plan (due to small negative points on explanations) while they are also incentivized to identify the feasibility and optimality of the given plan correctly, with a reward for identifying the correct properties and a penalty for doing this wrongly.

Each participant was shown 12 maps. For 6 of them, they were shown the optimal robot plan, and when they asked for an explanation, they were randomly shown different types of explanations from [2]. For the rest, they were either shown a (explicable) plan that is optimal in their model with no explanation or a balanced plan with a shorter explanation (the choice is again randomized). In other words, for the same map different participants may have seen optimal, explicable or balanced plans. We had 27 participants, 4 female and 22 male of age 19-31 (1 participant did not reveal their demographic) with a total of 382 responses across all maps.

6.2. *Salient findings*

Fig. 7 shows how people responded to different kinds of plans and their associated explanations. These results are from the two problem instances that included both a balanced and a fully explicable plan. Out of 54 user responses to these, 13 were for explicable plans and 12 for the balanced ones. From the perspective of the human, the balanced plan and the robot optimal plan do not make any difference since both of them appear suboptimal. This is evident from the fact that the click-through rate for explanations in these two conditions are similar (**H1a**) – the high click-through rates for perceived suboptimality conform to the expectations of **H1a**. Furthermore, the rate of explanations is much less for explicable plans as desired (**H2**).

Table 3 shows the statistics of the explanations / plans. These results are from 124 problem instances that required MCEs as per Section 2.4, and 25 and 40 instances that contained balanced and explicable plans respectively. As desired, the robot gains in the reduced length of explanations but loses out in the cost of plans produced as it progresses along the spectrum of optimal to explicable plans. Thus, while Table 3 demonstrates the explanation versus explicability trade-off from the robot's point of view, Fig. 7 shows how this trade-off is perceived from the human's perspective.

It is interesting to see that in Fig. 7 almost a third of the time, participants still asked for explanations even when the plan was explicable, i.e. optimal in their map. This may be an artifact of the risk-averse behavior incentivized by the gamification of the explanation process as well as an indication of the cognitive burden on the humans who are never (cost) optimal planners. Furthermore, the participants also did *not* ask for explanations around 20-25% of the time when they "should have" (i.e. suboptimal plan in the human model). There was no clear trend here (e.g. decreasing rate for explanations asked due to, perhaps, increasing trust during the course of the experiment) and was most likely due to limitations of the inferential capability of humans. Thus, going forward, the objective function must look to incorporate the cost or difficulty of analyzing the plans and explanations from the point of view of the human in addition to that in MEGA(4) and Table 3 modeled from the perspective of the robot.

Finally, in Fig. 8, we show how the participants responded to inexplicable plans, in terms of their click-through rate on the explanation request button. Fig. 8(left) shows the % of times subjects asked for explanations while Fig. 8(right) shows the same w.r.t. the number of participants. They indicate the variance of human response to the explicability-explanations trade-off. Such information can be used to model the α parameter to situate the explicability versus explanation trade-off according to preferences of individual users. It is interesting to see that the distribution of participants (right) seem to be bimodal indicating that subjects are either particularly skewed towards risk-averse behavior or not, rather than a normal distribution of responses to the explanation-explicability trade-off. This is somewhat counter-intuitive and against expectations (**H1**) and further motivates the need for learning α interactively.

7. Related work

Human-aware planning Efforts to make planning more "human-aware" have largely focused on incorporating an agent's understanding $\mathcal{M}_r^H$ of the human model $\mathcal{M}^H$ into its decision making. In addition, the need for human-aware agents to be able to explain their behavior has received increased attention in recent times [25]. This is highlighted in the success of recent workshops on explainable AI [26] and planning in particular [27]. Since then the importance of considering the human's understanding $\mathcal{M}_h^R$ of the agent's actual model $\mathcal{M}^R$ in the planning process has also been acknowledged, sometimes implicitly [28] and later explicitly [3,2]. These considerations allow a human-aware agent to conceive novel and interesting behaviors by reasoning both in the space of plans as well as models. This work is the first of its kind, folding in model space considerations of explanations as studied in human-aware literature into the agent's plan generation process itself.

In the model space, modifications to the human mental model $\mathcal{M}_h^R$ may be used for explanations [2] while reasoning over the actual task model $\mathcal{M}^H$ can reveal interesting behavior by affecting the state of the human, such as in [29]. In the plan space, a human-aware agent can use $\mathcal{M}^H$ and $\mathcal{M}_h^R$ to compute joint plans for teamwork [30] or generate behavior that conforms to the human's preferences [31,28,32–35] and expectations [36,3,13] and create plans that help the human understand the robot's objectives [37].

In general, preference modeling looks at constraints on plan generation if the robot wants to contribute to the human utility, while explicability addresses how the robot can adapt its behavior to human expectation (as required by the human mental model). For a detailed discussion of these distinctions, we refer the reader to [38]. In the following, We will instead review the existing literature and emphasize the key differentiators for our unified framework.

Epistemic planning It is well understood in social sciences that explanations must be generated while keeping in mind the beliefs of the agent receiving the explanation [39]. As such, epistemic planning makes for an excellent framework for studying the problem of generating these explanations. While the most general formulation of epistemic planning has been shown to be undecidable, many simpler fragments have been identified [40]. In human-aware planning settings too, there is wide consensus that epistemic planning could be an extremely useful tool. Readers can refer to [41] for an overview of works done in employing epistemic planning for "social planning". Recently, there has been a lot of interest in developing efficient methods for planning in such settings [17,42–45]. Previous works (cf. [46]) have also investigated the use of speech acts in planning problems. Following the conventions set by these works, the explanatory actions studied within this paper can be viewed as INFORM acts.

Clearly, a traditional epistemic planning framework could capture many aspects of balanced planning. Our work was the first to make this observation and leveraged existing planning compilations for epistemic planning for these purposes. However, there is one aspect of existing explanation generation that is not currently studied within extant epistemic planning literature, namely, the optimality of the plan. Existing explanation work, particularly model-reconciliation ones, focuses on choosing model updates that ensure that the plan is perceived as optimal in the updated human model. Even if the human is not necessarily capable of coming up with optimal plans on their own, ensuring optimality in the updated model will ensure that the humans won't consider alternative plans they might incorrectly think are better than the proposed one.

Communicative actions Our work also looks at the use of explanatory actions as a means of communicating information to the human observer. The most obvious types of such explanatory action includes purely communicative actions such as speech [47] or the use of mixed reality projections [48,49]. Recent works have shown that physical agents could also use movements to relay information such as intention [50,51] and incapability [52]. Our framework allows for a natural trade-off between these different types of communication.

Explanations for planning problems Many recent works dealing with explanation generation for planning have looked at characterizing explanations in terms of the types of questions they answer [53,54]. This characterization is orthogonal to the question of what type of information constitutes valid explanations. Putting aside questions regarding observability, the reason why a user may request an explanation is either due to knowledge mismatch (incomplete or incorrect knowledge of the task) or due to limitations of their inferential capabilities. The answer to any of these questions would require correcting the human's model of the task and/or

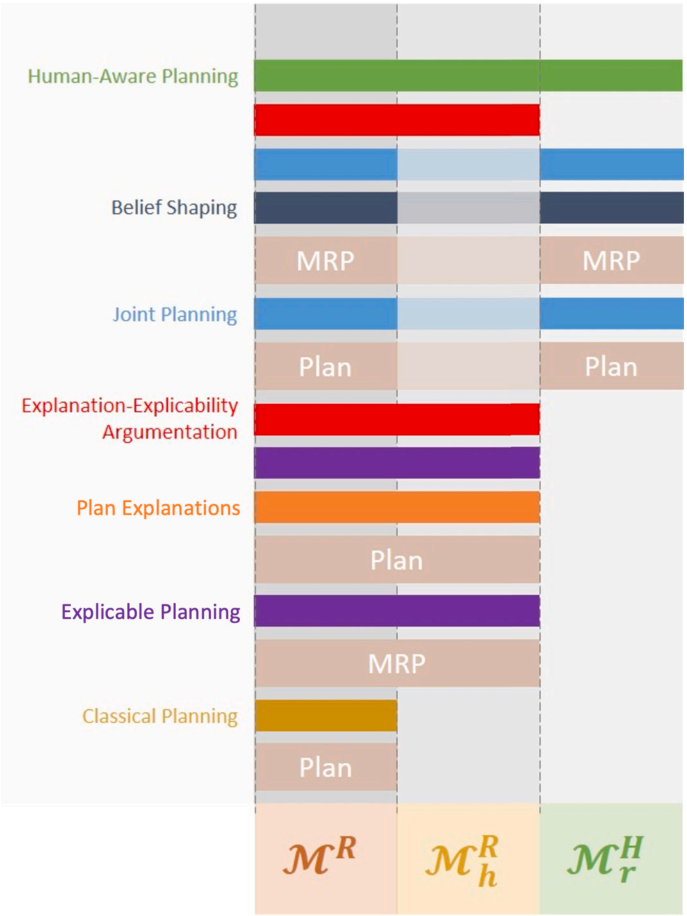


Fig. 9. A "Subsumption Architecture" for Human-Aware Planning.

providing inferential assistance. Works that have looked at model reconciliation explanations have mostly focused on the former. Explanations discussed in this paper can be viewed as an answer to the question *"Why this plan?"* (which can also be viewed as a contrastive question of the form *"Why this plan and not any other plan?"*). This is not to say that in complex scenarios just the model reconciliation information would suffice, but it would need to be supplemented with information internal to the model that can address the differences in inferential capabilities. Use of abstractions [12], providing refutation of specific foils [12] and providing causal explanations [55] could be used to augment model reconciliation.

Explicable planning Explicable planning looks at cases where the agent is incapable of updating the users' expectations and can choose to following the plan that best matches the user expectations and is valid in the robot model. Two representative works in this direction are [3] and [13]. [3] investigates scenarios where the human model may be unknown while [13] proposes an iterative planning formalism that tries to find the most explicable plan by generating all possible valid solutions of a given cost threshold and then tries to find the most explicable plan from that set. Unlike the work presented here they look at more general distance measures for explicability, some of which are based on global plan properties. We can extend our current formulation to take into account such scores by turning these distances into the cost of an extra GOAL action (similar to the balancing formalism that allows for sub-optimality).

8. Concluding remarks

We saw how an agent can be human-aware by balancing the cost of departure from optimality (in order to conform to human expectations) versus the cost of explaining away causes of *perceived* suboptimality. We showed how this can be achieved by a novel model-space search algorithm and evaluated different properties of the approach on well-known IPC domains as well as through human factors experiments in a mock USAR domain. We then presented a unifying formulation for the task of planning in the presence of users with misaligned mental models. This formulation allows us to unify, for the first time, explanatory and explicable paradigms into a single framework that is also compilable to classical planning thereby being computationally more efficient than methods that rely on direct model space search.

8.1. Future work

One of the exciting features of our work is that we are able to place the Expectation-Aware Planning paradigm within the realm of epistemic planning, thereby laying the ground work to study more complex interaction scenarios including cases with more levels of nesting, uncertainty about mental models, more expressive models, incorporating non-deterministic effects, and so on. It would also be worth investigating specific considerations for choosing heuristics or formulating new ones for such problems. It is worth noting that the advantage the planning formalism gets over one that employs a naive search over the possible set of model updates is the fact that in the former case, we can directly employ heuristics and search methods developed for planning. However, model-space search as a problem is relatively under-studied, and developments in heuristics for model-space search could make algorithms like MEGA potentially competitive to the compilation method discussed here.

It is well known how humans make better decisions when they have to explain them [56]. In this work, in being able to reason about the explainability of its decisions, an AI planning agent was similarly able to make better decisions by explicitly considering the implications of its behavior on the mental model of the human in the loop. The work leaves open several intriguing avenues of further research, including expanding on this unifying thread to capture the many flavors of human-aware behavior, which also includes consideration of the human capability model as well. We thus end with the intriguing prospect of a hierarchy of human-aware behaviors that can inform the effective design of planning agents that work with end users, going forward.

8.2. A “subsumption architecture” for human-aware planning

The different behaviors engendered by this multi-model argumentation can be composed to form more and more sophisticated forms of human-aware behavior. We thus conclude this discussion with a hierarchical composition of behaviors in the form of a “subsumption architecture” for human-aware planning, inspired by [57]. This is illustrated in Fig. 9. The basic reasoning engines are the Plan and MRP (Model Reconciliation) modules – the former accepts a model of a planning problem (and optionally the model of the user) and produces a plan while the latter accepts the two models and produces a new model where some given properties hold. The former operates in the space of plans and gives rise to single agent, joint, and explicable planning depending on the models it is operating on. The latter operates in the space of models to produce explanations and belief-shaping behavior. These get expanded onto joint planning once the (agent’s estimation of the) human capability model $\mathcal{M}_r^H$ is factored into the reasoning modules. These can then be composed to form argumentation modules for trading off explanations and explicability in human-aware planning.

CRediT authorship contribution statement

Sarath Sreedharan: Writing – review & editing, Writing – original draft, Conceptualization. **Tathagata Chakraborti:** Writing – review & editing, Writing – original draft, Conceptualization. **Christian Muise:** Writing – review & editing, Writing – original draft, Supervision, Conceptualization. **Subbarao Kambhampati:** Writing – review & editing, Writing – original draft, Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

Kambhampati’s research is supported in part by ONR grants N00014-16-1-2892, N00014-18-1-2442, N00014-18-1-2840, N00014-19-1-2119, AFOSR grant FA9550-18-1-0067, DARPA SAIL-ON grant W911NF-19-2-0006, NSF grants 1936997 (C-ACCEL), 1844325, and a NASA grant NNX17AD06G. Muise’s research is supported in part by funding from Natural Sciences and Engineering Research Council of Canada (NSERC) RGPIN-2020-05501. Chakraborti was also supported by the IBM Ph.D. Fellowship during the formative years of the project. We would also like to thank all the reviewers for their detailed and constructive feedback. A special word of thanks to Sachin Grover (Arizona State University) for his help in setting up the user studies.

References

[1] T. Chakraborti, S. Kambhampati, M. Scheutz, Y. Zhang, AI challenges in human-robot cognitive teaming, arXiv preprint, arXiv:1707.04775, 2017.
[2] T. Chakraborti, S. Sreedharan, Y. Zhang, S. Kambhampati, Plan explanations as model reconciliation: moving beyond explanation as soliloquy, in: IJCAI, 2017.
[3] Y. Zhang, S. Sreedharan, A. Kulkarni, T. Chakraborti, H.H. Zhuo, S. Kambhampati, Plan explicability and predictability for robot task planning, in: ICRA, 2017.
[4] C.E. Bartlett, Communication between Teammates in Urban Search and Rescue, Thesis, Arizona State University, 2015.
[5] H. Geffner, B. Bonet, A concise introduction to models and methods for automated planning, in: Synthesis Lectures on Artificial Intelligence and Machine Learning, 2013.

[6] S. Sreedharan, P. Bercher, S. Kambhampati, On the computational complexity of model reconciliations, in: IJCAI, 2022, pp. 4657–4664.

[7] S. Sreedharan, A.O. Hernandez, A.P. Mishra, S. Kambhampati, Model-free model reconciliation, in: IJCAI, 2019.

[8] S. Sengupta, T. Chakraborti, S. Sreedharan, S. Kambhampati, RADAR - a proactive decision support system for human-in-the-loop planning, in: ICAPS Workshop on User Interfaces for Scheduling and Planning, 2017.

[9] P.A. Ortega, A.A. Stocker, Human decision-making under limited time, Adv. Neural Inf. Process. Syst. 29 (2016).

[10] H.J. Jeon, S. Milli, A.D. Dragan, Reward-rational (implicit) choice: a unifying formalism for reward learning, in: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual, Curran Associates, Inc., Virtual, 2020, pp. 4415–4426.

[11] S. Sreedharan, T. Chakraborti, S. Kambhampati, Foundations of explanations as model reconciliation, Artif. Intell. 301 (2021) 103558.

[12] S. Sreedharan, S. Srivastava, S. Kambhampati, Hierarchical expertise-level modeling for user specific contrastive explanations, in: IJCAI, 2018.

[13] A. Kulkarni, T. Chakraborti, Y. Zha, S.G. Vadlamudi, Y. Zhang, S. Kambhampati, Explicable robot planning as minimizing distance from expected behavior, in: AAMAS Extended Abstract, 2019.

[14] Z. Zahedi, A. Olmo, T. Chakraborti, S. Sreedharan, S. Kambhampati, Towards understanding user preferences for explanation types in explanation as model reconciliation, in: HRI Late Breaking Report, 2019.

[15] B. Löwe, E. Pacuit, A. Witzel, Del planning and some tractable cases, in: Logic, Rationality, and Interaction: Third International Workshop, Proceedings 3, LORI 2011, Guangzhou, China, October 10-13, 2011, Springer, 2011, pp. 179–192.

[16] C.J. Muise, P. Felli, T. Miller, A.R. Pearce, L. Sonenberg, Planning for a single agent in a multi-agent environment using fond, in: IJCAI, 2016.

[17] C.J. Muise, V. Belle, P. Felli, S.A. McIlraith, T. Miller, A.R. Pearce, L. Sonenberg, Planning over multi-agent epistemic states: a classical planning approach, in: AAAI, 2015.

[18] T. Bylander, The computational complexity of propositional STRIPS planning, Artif. Intell. (1994).

[19] M. Helmert, The Fast Downward Planning System, JAIR, 2006.

[20] M. Helmert, C. Domshlak, Lm-cut: optimal planning with the landmark-cut heuristic, in: Seventh International Planning Competition (IPC 2011), Deterministic Part, 2011, pp. 103–105.

[21] B. Bonet, H. Geffner, Planning as heuristic search, Artif. Intell. 129 (2001) 5–33.

[22] C. Wayllace, P. Hou, W. Yeoh, T.C. Son, Goal recognition design with stochastic agent action outcomes, in: IJCAI, 2016.

[23] S. Keren, L. Pineda, A. Gal, E. Karpas, S. Zilberstein, Equi-reward utility maximizing design in stochastic environments, in: IJCAI, 2017.

[24] T. Chakraborti, S. Sreedharan, S. Grover, S. Kambhampati, Plan explanations as model reconciliation – an empirical study, in: HRI, 2019.

[25] T. Chakraborti, S. Sreedharan, S. Kambhampati, The emerging landscape of explainable AI planning and decision making, in: IJCAI, 2020.

[26] XAI, Workshop Series, Proceedings of the IJCAI-ECAI Workshop on Explainable AI (XAI), 2018–2020.

[27] XAI, Workshop Series, Proceedings of the ICAPS Workshop on Explainable AI Planning, (XAIP), 2018–2020.

[28] R. Alami, M. Gharbi, B. Vadant, R. Lallement, A. Suarez, On human-aware task and motion planning abilities for a teammate robot, in: RSS Workshop on Human-Robot Collaboration for Industrial Manufacturing Workshop, 2014.

[29] T. Chakraborti, G. Briggs, K. Talamadupula, Y. Zhang, M. Scheutz, D. Smith, S. Kambhampati, Planning for serendipity, in: IROS, 2015.

[30] K. Talamadupula, G. Briggs, T. Chakraborti, M. Scheutz, S. Kambhampati, Coordination in human-robot teams using mental modeling and plan recognition, in: IROS, 2014.

[31] R. Alami, A. Clodic, V. Montreuil, E.A. Sisbot, R. Chatila, Toward human-aware robot task planning, in: AAAI Spring Symposium: To Boldly Go Where No Human-Robot Team Has Gone Before, 2006.

[32] M. Cirillo, L. Karlsson, A. Saffiotti, Human-aware task planning: an application to mobile robots, ACM Trans. Intell. Syst. Technol. (2010).

[33] U. Koeckemann, F. Pecora, L. Karlsson, Grandpa hates robots - interaction constraints for planning in inhabited environments, in: AAAI, 2014.

[34] S. Tomic, F. Pecora, A. Saffiotti, Too cool for school — Adding social constraints in human aware planning, in: Workshop on Cognitive Robotics (CogRob), 2014.

[35] T. Chakraborti, Y. Zhang, D. Smith, S. Kambhampati, Planning with resource conflicts in human-robot cohabitation, in: AAMAS, 2016.

[36] A. Dragan, K. Lee, S. Srinivasa, Legibility and predictability of robot motion, in: HRI, 2013.

[37] D. Sadigh, S. Sastry, S.A. Seshia, A.D. Dragan, Planning for autonomous cars that leverage effects on human actions, in: Robotics: Science and Systems, 2016.

[38] T. Chakraborti, A. Kulkarni, S. Sreedharan, D.E. Smith, S. Kambhampati, Explicability? Legibility? Predictability? Transparency? Privacy? Security? The emerging landscape of interpretable agent behavior, in: ICAPS, 2019.

[39] T. Miller, Explanation in artificial intelligence: insights from the social sciences, Artif. Intell. J. (2017).

[40] T. Bolander, M.H. Jensen, F. Schwarzentruber, Complexity results in epistemic planning, in: IJCAI, 2015.

[41] T. Miller, Social planning – reasoning with and about others, in: Invited Talk at IJCAI Workshop on Impedance Matching in Cognitive Partnerships, 2017.

[42] F. Kominis, H. Geffner, Beliefs in multiagent planning: from one agent to many, in: ICAPS, 2015.

[43] F. Kominis, H. Geffner, Multiagent online planning with nested beliefs and dialogue, in: ICAPS, 2017.

[44] T. Le, F. Fabiano, T.C. Son, E. Pontelli, EFP and PG-EFP: epistemic forward search planners in multi-agent domains, in: ICAPS, 2018.

[45] X. Huang, B. Fang, H. Wan, Y. Liu, A general multi-agent epistemic planner based on higher-order belief change, in: IJCAI, 2018.

[46] P.R. Cohen, C.R. Perrault, Elements of a plan-based theory of speech acts, Cogn. Sci. (1979).

[47] S. Tellex, R. Knepper, A. Li, D. Rus, N. Roy, Asking for help using inverse semantics, in: RSS, 2014.

[48] T. Chakraborti, S. Sreedharan, S. Kambhampati, Projection-aware task planning and execution for human-in-the-loop operation of robots in a mixed-reality workspace, in: IROS, 2018.

[49] R.K. Ganesan, Mediating Human-Robot Collaboration through Mixed Reality Cues, Ph.D. thesis, Arizona State University, 2017.

[50] A.M. MacNally, N. Lipovetzky, M. Ramirez, A.R. Pearce, Action selection for transparent planning, in: AAMAS, 2018.

[51] A. Dragan, K. Lee, S. Srinivasa, Legibility and predictability of robot motion, in: HRI, 2013.

[52] M. Kwon, S. Huang, A. Dragan, Expressing robot incapability, in: HRI, 2018.

[53] M. Fox, D. Long, D. Magazzeni, Explainable planning, in: IJCAI Workshop on Explainable AI (XAI), 2017.

[54] D.E. Smith, Planning as an iterative process, in: AAAI, 2012.

[55] B. Seegebarth, F. Müller, B. Schattenberg, S. Biundo, Making hybrid plans more clear to human users – a formal approach for generating sound explanations, in: ICAPS, 2012.

[56] H. Mercier, D. Sperber, Why do humans reason? Arguments for an argumentative theory, Behav. Brain Sci. (2011).

[57] R. Brooks, A robust layered control system for a mobile robot, IEEE J. Robot. Autom. (1986).