Towards Human-Machine Symbiosis: Design for Effective AI Facilitation

by

Mehrdad Zakershahrak

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved June 2021 by the
Graduate Supervisory Committee:

Nancy Cooke, Co-Chair
Yu Zhang, Co-Chair
Hani Ben Amor
Siddharth Srivastava
Sharon Hsiao

ARIZONA STATE UNIVERSITY

August 2021

ABSTRACT

The rapid increase in the volume and complexity of data lead to accelerated Artificial Intelligence (AI) applications, primarily as intelligent machines, in everyday life. Providing explanations is considered an imperative ability for an AI agent in a human-robot teaming framework, which provides the rationale behind an AI agent's decision-making. Therefore, the validity of the AI models is constrained based on their ability to explain their decision-making rationale. On the other hand, AI agents cannot perceive the social situation that human experts may recognize using their background knowledge, specifically in cybersecurity and the military. Social behavior depends on situation awareness, and it relies on interpretability, transparency, and fairness when we envision efficient Human-AI collaboration. Consequently, the human remains an essential element for planning, especially when the problem's constraints are difficult to express for an agent in a dynamic setting. This dissertation will first develop different model-based explanation generation approaches to predict where the human teammate would misunderstand the plan and, therefore, generate an explanation accordingly. The robot's generated explanation or interactive explicable behavior maintains the human teammate's cognitive workload and increases the overall team situation awareness throughout human-robot interaction. Further, it will focus on a rule-based model to preserve the collaborative engagement of the team by exploring essential aspects of the facilitator agent design. In addition to recognizing wherein the plan might be discrepancies, focusing on the decision-making process provides insight into the reason behind the conflict between the human expectation and the robot's behavior. Employing a rule-based framework will shift the focus from assisting an individual (human) teammate to helping the team interactively while maintaining collaboration. Hence, concentrating on teaming provides the opportunity to recognize some cognitive biases that skew the teammate's expectations and affect interaction

behavior. This dissertation investigates how to maintain collaboration engagement or cognitive readiness for collaborative planning tasks. Moreover, this dissertation aims to lay out a planning framework focusing on the human teammate's cognitive abilities to understand the machine-provided explanations while collaborating on a planning task. Consequently, this dissertation explored the design for AI facilitator, helping a team tasked with a challenging task to plan collaboratively, mitigating the teaming biases, and communicate effectively. This dissertation investigates the effect of some cognitive biases on the task outcome and shapes the utility function. The facilitator's role is to facilitate goal alignment, the consensus of planning strategies, utility management, effective communication, and mitigate biases.

ACKNOWLEDGMENTS

I would like to take this opportunity to acknowledge those who have impacted my PhD accomplishment and doctorate journey.

First, I express my deep gratitude and sincere appreciation to my supervisors, Prof. Nancy Cooke, and Dr. Yu Zhang for their guidance, encouragement and support throughout my PhD program. Their expertise and insightful feedback pushed me to sharpen my thinking and elevated my work to a higher level.

My sincere appreciation also goes to Ira A. Fulton Schools of Engineering of the ASU, School of Computing, Informatics, and Decision Systems Engineering (CIDSE), and specially Center for Human, Artificial Intelligence, and Robot Teaming (CERT) for generously funding this PhD research. I extend further thanks to NASA, U.S. Air Force, National Science Foundation (NSF), Center for Human, Artificial Intelligence, and Robot Teaming (CHART), and Defense Advanced Research Projects Agency (DARPA) for their generous support of this line of research to push the boundaries of human understanding.

I am thankful to my parents who have been an inspiration to me throughout my life. My father is the symbol of perseverance to me. He always inspired me to believe in my abilities and never settle to achieve my dreams. My mom, who has always been there for me, even when I didn't know, and looked after for me. Thank you for all of your kindness, support and patience.

Finally, special thanks to my wife, Samira, who supported me in every possible way. We have been through a lot together, and it amazes me how she somehow kept it all together. Everyday, I increasingly realize her love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

PREFACE

My research originally stemmed from my passion for understanding how our brain, incredibly conscientious, functions while interacting with the world.

As the world moves further into the artificial intelligence (AI) age, we mimic consciousness's functionalities and attributes into robots. Thus, learning how the robots can be engaged in our activities is an approach to discern our cognitive abilities. Further, it sheds light on how we model others to predict and adjust our behaviors every day in complex teaming scenarios.

My passion is to find out and develop tools to forecast sizable teams' rationality, such as societies, utilizing AI and planning, interacting under complicated situations, cognitive biases, specific expectations, etc.

Chapter 1

INTRODUCTION

As human-agent collaboration becomes more ubiquitous in our everyday life, intelligent machines are increasingly taking on more crucial roles in human-AI teams. Consequently, the validity of the intelligent machines is constrained by their ability to facilitate efficient collaboration. In this dissertation, I will study two distinct roles that AI agents are expected to play in human-AI interaction, one being an one-on-one relationship (i.e., a teammate) and one being one-to-many (i.e., a facilitator). In the role of a teammate, the AI agent focuses on establishing a mutual understanding between the human and AI peer. As a facilitator, the AI agent provides directions and suggestions to maintain collaborative engagement or cognitive readiness within the team.

AI Teammate

As intelligent robots become more prevalent in our lives, the interaction of such AI agents with humans becomes more frequent and essential. Similar to a human teammate, a robotic agent is required to not only understand its human peers, but also explain its own decision or behaviors when necessary. Explanations in a teaming context provide the rationale behind an individual agent's decision making, and help with building a shared situation awareness and maintaining trust between teammates Cooke (2015); Zakershahrak *et al.* (2019).

Prior work on generating explanations has been focused on providing the rationale behind the robot's decision making Chakraborti *et al.* (2017b, 2019). Therefore, the human is expected to understand an explanation regardless of how much information it contains. Although these approaches provide the right explanations from the explainer's perspective, they fail to heed the cognitive requirement of understanding an explanation from the explainee's perspective. Little discussion has been given on the ways of presenting such information Chakraborti *et al.* (2017a). One remaining challenge in explanation generation, is the consideration of the cognitive capabilities and preferences of the human to understand an explanation provided by a robot teammate Zakershahrak *et al.* (2020).

The main motivation of my research towards explanation generation is to maintain/reduce the cognitive workload of the human when sharing explanations. Explanation generation in human-robot teaming and explainable decision making can be categorized into the following dimensions:

1. *Cognitive* model explanation: how to explain the decision-making process of the robot
2. Explain domain model (dynamics), including the initial state and goal
3. Behavior explanation: directly explain a behavior

In chapters 2, 3, and 4, explanation is defined as any information that is provided to any human or agent, in terms of model features. The aim of providing explanations is to establish situation awareness in teaming. However, in chapter 5, I use facilitation interventions which is a rule-based approach to preserve the collaborative engagement of the team. The facilitation intervention can be viewed as implicit coaching behavior.

Another approach to keep the human in the loop for effective human-robot teaming is to provide explicable behavior. For effective teaming, a robot must maintain a behavioral model of its human teammates to project the team status and be aware of its human teammates' expectations of itself. Being aware of the human teammates' expectations leads to robot behaviors that align with the human expectation, thus facilitating more efficient and potentially safer teams. Chapter 2 addresses the problem of human-robot interaction with the consideration of such teammate models in sequential domains by leveraging the concept of plan explicability. This means the robot needs to work with (possibly incorrect) human planning preferences and learn human preconceptions about its model. To achieve this, I assumed that humans understand the agent's intention by attaching abstract tasks with the agent's actions, which I acknowledge as a labeling process. Then, I learn the human preferences using conditional random fields (CRFs). Then I use the learned model to label a new team plan (comprised of human and robot actions) to compute its explicability and predictability score. Having these measures helps the robot to dynamically synthesize plans that are more explainable based on human preferences.

Furthermore, throughout my research, I have explored the effect of explanation generation approaches on domain models: In Chapter 3, I argue that explanations, especially complex ones, should be provided in an online fashion, such that each explanation is broken into multiple parts, which are then communicated separately and intertwined with plan execution. One of the main challenges here is that the different parts of an explanation could be dependent on each other, which must be taken into account when generating online explanations Zakershahrak *et al.* (2019).

In Chapter 4, I focus on the influence of the order of information on the cognitive effort of the explainee in planning tasks. Considering that making an explanation is normally not an instantaneous effort; instead, information must be conveyed in a sequential order; furthermore, given the characterizations of our cognitive systems Ericsson and Smith (1991); Kahneman (2011), we often could not (or would not) wait until all the information has been conveyed before processing it. As a result, the order of presenting information matters. Hence, one of the keys to reducing cognitive effort is to minimize the cumulative effort required for processing all the information progressively in an explanation. Consequently, I term my approach *progressive explanation generation*, to capture that aspect Zakershahrak *et al.* (2020).

AI Facilitator

In Chapter 5, I design an online planning framework to study the effects of facilitation behavior throughout Human-AI teaming context. My approach to addressing this urgent need is to develop an AI capability to facilitate the commander's planning team's discussion. This rule-based behavior, in addition to recognizing wherein the plan might be discrepancies, focusing on the decision-making process provides insight into the reason behind the conflict between the human and the robot. Also, employing a rule-based framework, I shift the focus in this chapter from assisting an individual human teammate to helping the team interactively by focusing on the team decision processes.

The AI facilitation capability ensures that the planning team is always operating at a high level of decision effectiveness, ensuring that all team members are in harmonization. I expect these teams to be dynamic since as new, novel options are considered, it will require adding additional expertise to the team. I expect that the facilitation agent will enable the team to generate and analyze more prospects in less time. And the facilitation agent, as it is closely following the entire planning discussion, will be able to synthesize the plan and the plan rationale, which will be very useful for both a more detailed analysis of the selected plan and the execution of the plan.

In Chapter 6, I employ the findings of the approaches introduced in this dissertation to highlight my most important findings on human-robot team design, towards human-machine symbiosis.

Chapter 7 discusses the future directions of this line of research both in terms of efficient agent design and human-agent collaboration studies.

1.1 Overview of Contributions

Throughout this dissertation, I have studied two distinct AI roles: (1) AI as a teammate; (2) AI as a team facilitator. AI as a teammate aims to take over the roles in the team that is hard, dangerous, or even boring for the human. Therefore, for this role, the impact of explanation generation is to increase the situation awareness, trust, and transparency in human teammates throughout the interaction. Hence, AI design focuses on reducing/maintaining the required human mental workload to understand AI reasoning. On the other hand, AI as a team facilitator focuses on the overall team behavior such as goal alignment, planning process strategies, effective communication, utility alignment, bias mitigation, etc. Therefore, I explored AI facilitator design to answer (1) how a team articulates the given information and (2) distinguishes the critical information to fulfill a task. I elaborated my contributions toward each AI goal in the following subsections.

1.1.1 Chapter 2

My contribution in this chapter includes generating joint plan for human and robot which is more explainable, and implementing the ideas in a human-robot first respondent task scenario. I extend the plan explicability to consider interactive settings in which the human and robot's behaviors can influence each other. I term this new measure Interactive Plan Explicability (IPE). I compare the joint plan generated by our approach with the consideration of this measure using the fast forward (FF) planner, with the plan generated by FF without such consideration and the plan created with human subjects interacting with a robot running an FF planner. Because the human teammate is expected to adapt to the robot's behavior dynamically when it deviates from her expectation, the plan created with human subjects is expected to be more explicable than the FF plan and comparable to the explicable plan generated by our approach. Results indicate that the explicability score of plans generated by our algorithm is indeed closer to the human interactive plan than the plan developed by FF, implying that the plans developed by our algorithms align better with the expected plans of the human during execution. This can lead to more efficient collaboration in practice.

1.1.2 Chapter 3

In this chapter, I provide a new (online) approach to generate explanations given both the domain models of the robot and the human. Here, I break down a complex explanation and calculate the optimal time to share sub-explanations intertwined with the robot plan execution to increase the team situation awareness and trust. For a robotic teammate, the ability to generate explanations to justify its behavior is one of an explainable agency's essential requirements. Prior work on explanation generation has been focused on supporting the rationale behind the robot's decision or behavior. These approaches, however, fail to consider the mental demand for understanding the received explanation. In other words, the human teammate is expected to understand an explanation no matter how much information is presented. In this chapter, I argue that explanations, especially those of a complex nature, should be made in an online fashion during the execution. This helps spread out the information to be explained and thus reduces humans' mental workload in highly cognitive demanding tasks. However, a challenge here is that the different parts of an explanation may be dependent on each other, which must be taken into account when generating online explanations. To this end, a general formulation of online explanation generation is presented with three variations satisfying different *"online"* properties. The new explanation generation methods are based on a model reconciliation setting introduced in our prior work. I evaluated these methods both with human subjects in a simulated rover domain, using NASA Task Load Index (TLX), and synthetically with ten different problems across two standard IPC domains. Results strongly suggest that my methods generate explanations that are perceived as less cognitively demanding and much preferred over the baselines and are computationally efficient.

1.1.3 Chapter 4

In this chapter, I provide a learning approach to understand how humans progressively perceive a complex explanation. The output of my approach is to provide the optimal sequence of sub-explanation that achieves the least cognitive demand to provide the maximum possible situation awareness. Prior work on generating explanations in a planning and decision-making context has focused on providing the rationale behind an AI agent's decision-making. Although these methods offer the proper explanations from the explainer's perspective, they fail to heed the cognitive requirement of understanding an explanation from the explainee's (the human's) perspective. In this chapter, I set out to address this issue by first considering the influence of information order in an explanation, or the ***progressiveness of explanations***. Intuitively, progression builds later concepts on previous ones and is known to contribute to better learning. In this work, I aim to investigate similar effects during explanation generation when an explanation is broken into multiple parts that are communicated sequentially. The challenge here lies in modeling the humans' preferences for information order in receiving such explanations to assist understanding. Given this sequential process, a formulation based on goal-based MDP for generating progressive explanations is presented. The reward function of this

MDP is learned via inverse reinforcement learning based on explanations that are retrieved via human subject studies. I first evaluated our approach on a scavenger-hunt domain to demonstrate its effectiveness in capturing the humans' preferences. Upon analyzing the results, it revealed something more fundamental: the preferences arise strongly from both domain-dependent and independence features. The correlation with domain-independent features pushed us to verify this result further in an escape room domain. Results confirmed our hypothesis that the process of understanding an explanation was dynamic. The human preference that reflected this aspect corresponded precisely to the progression for knowledge assimilation hidden deeper in our cognitive function. I showed that progressive explanations achieved better task performance and reduced cognitive load. These results shed light on designing explainable robots across various domains. Robotic applications are strongly correlated with independent domain features, which I further verified using an escape-room domain. Results strongly confirmed that order matters when making explanations and sometimes even plays a dominant role. My method is evaluated on two domains: escape-room and scavenger-hunt domain employing human subjects and IPC rover domain for simulation. The results show that the progressive explanation generation method reduces the cognitive load over two baselines.

1.1.4 Chapter 5

To better understand the team processes to decide on complex tasks, in chapter 5, I focused on team dynamics during the planning. Hence, one question I am answering is: *How teammates articulate the inputs to the planning problem?* This question consists of two parts, *what to solve?*, and *How to solve it?* Since I concentrate on cognitive demanding collaborative tasks, the most important factors toward answering these questions are: the agreed utility function, mitigating team cognitive biases, and a heuristic to determine effective communication. Finally, the result of this chapter will elucidates the important factors in designing an AI facilitator agent, as well as the facilitator effect on overall collaboration and team communication.

1.2 Challenges

One of the main challenges in this work is the distinction and identification of human mental models. In particular, the identification of the human's mental model (dynamics model) is a general problem across all the chapters. In this dissertation, I have utilized two common approaches:

1. I assume/impose a human model, i.e.,I assume the initial human model is known and know how it may be changed by the agent's explanation/facilitation behaviors, or pre-train the human's initial model in terms of what is know and unknown;

2. Learn the human model.

The first approach is functional when studying the overall team interaction pattern, specifically the impact of robots' actions on humans teaming behavior and vice versa throughout a teaming scheme. This approach is helpful when we design or test interaction strategies for human-robot teams. On the other hand, the second approach is beneficial when focusing on how each explanation is perceived from an individual team member's perspective. For instance, the second approach is appropriate for understanding each new sub-explanation's effect on the previously generated sub-explanations.

Chapter 2 used a labeling approach to learn about the human model (second approach). The initial model associates a label to each action/behavior of each teammate. These labels then will be used to predict the human expectation in new scenarios based on the calculated similarity of learned actions-label pairs.

Chapter 3, 4 & 5: I employed the first approach to know the human's model or pre-train it. The mental states and the rewards associated with them are captured with respect to the task. Hence, the reward is defined as a linear weighted sum of features categorized as domain-dependent and independent, calculated from states in the domain. This definition affects many design choices of the task and the environment in which humans and robots interact. One interesting future work in this direction is extracting features and applying them towards the reward definition, given the task and environment as input.

Chapter 2

INTERACTIVE PLAN EXPLICABILITY

There is an ever-growing number of robotic applications, among which many depend on the ability of the robot being an effective teammate. The team effectiveness is a compound metric that captures how team members consistently act according to the expectation of the team Cooke *et al.* (2015). Effective teaming consists of (1) situation awareness in terms of recognizing the status of the team tasks and teammates' states, (2) shared mental model to predict or foresee the next action of the team under the current context, (3) direct and indirect interaction between the teammates, and (4) taking proactive actions considering other team members' subgoals to support their achievement Cooke (2015); Cooke *et al.* (2015).

Hence, to achieve a comparable level of efficiency as in human teams, a key challenge in human-robot teaming is to ensure that the robot always assists humans in an expected and understandable fashion that is consistent with the teaming context. To do this, a robotic teammate must first be able to recognize the intent of its human teammates and then coordinate with them in a way that is expected. This argument has been more prominently made recently in human-robot teaming research Chakraborti *et al.* (2017a). It is argued that the robot must maintain mental models of its human teammates. These mental models not only include human intent and other mental states, but also their expectations of the robot. The ability to accommodate such expectations can lead to more fluent teaming, even though it often leads to suboptimal plans due to the differences between the robot's plan and that of the human expectation.

There are many reasons why the robot's plan would differ from that of the human expectation. For example, humans may misunderstand the abilities of their robotic teammates, resulting in inconsistencies between the robot's domain model and the human's interpretation of this model. Modeling the human expectation is particularly challenging since the robot often does not have direct access to it and it is difficult to be learned. To address this issue, we use the notion of plan explicability discussed in Zhang *et al.* (2017), where an approach was proposed to learn the model of expectation based on a labeling process. That work, however, focused on the human being an observer. In this work, we extend the notion of plan explicability to an interactive setting where the human is cooperating with the robot.

In an interactive teaming setting, the behaviors of the human and robot can influence each other. For instance, consider a scenario where a human is assigned to a first-response task with a robotic teammate after a disaster occurred. Due to the hazardous situation in the environment, the human stays at the command center and the robot enters the environment to provide medical assistance at the locations where injured people are likely to be present. The team's goal is to provide medical assistance as quickly as possible. However, due to damages incurred by the disaster, some paths may be blocked which is unknown to the human and only perceivable by the robot teammate that is working at the disaster scene. Hence, the situation may happen that the human would command the robot to visit a room and expects the robot to follow the shortest path in her view, but the robot would take a longer route due to obstacles that the human is unaware of. This robot behavior from the human's perspective is inexplicable. In an interactive setting, this may trigger the human to more closely monitor the robot's behavior and command the robot more frequently. These interactions directly influence the mental models of the human and hence can change her teaming behavior, which would in turn affect the robot, thus forming a tight interaction. In such a case, a plan is comprised of both human and robot actions, and the influence of the agent's behavior on each other must be explicitly considered.

In such teaming scenarios, the ability of the robot to predict the joint team behavior hence can increase the team effectiveness since the robot can now anticipate the human's response and how it should react accordingly. This in turn allows the robot to choose plans that are the least interruptive to the human thus improving teaming fluency. To achieve this, similar to plan explicability Zhang *et al.* (2017), we assume that humans interpret robot plans by attaching abstract task labels to robot actions as a labeling process. The difference here is that the plan contains not only robot actions but also human actions. The human actions provide the teaming context for the labeling process, which is modeled using Conditional Random Fields (CRFs). The learned model can be used to label a new team plan to compute its interactive plan explicability score, similar to the explicability score in Zhang *et al.* (2017). Having these measures allows the robot to synthesize plans that are more explicable to the human. Our contribution in this work includes extending plan explicability to interactive teaming scenarios, implementing a plan monitoring and replanning process during actual human-robot interaction, as well as evaluating this approach using a synthetic first response domain.

2.1 Related Work

The notion of robotic teammate, or that using robots to complement humans in various tasks, has attracted lots of research interest. At the same time, however, the realization of this notion is challenging due to the human-aware aspect Chakraborti *et al.* (2017a), or that the robot must consider the human in the loop, both in terms of physical and mental models while planning to achieve the team goal. In such cases, it is no longer sufficient to model humans as parts of the environment Chakraborti *et al.*

(2015). Instead, human-robot teaming applications require the robot to be proactive in assisting humans Fern *et al.* (2007).

There are different aspects to be considered for human-robot teaming. First, the robot must take the human's intent into account. Various plan recognition algorithms Kautz and Allen (1986); Ramırez and Geffner (2010) can be applied to perform plan recognition based on a given set of observations. The challenge is how the robot can utilize this information to synthesize a plan while avoiding conflicts or providing proactive assistance Chakraborti *et al.* (2016); Cirillo *et al.* (2009). There are different approaches to planning with such consideration Chakraborti *et al.* (2015, 2017b).

A more challenging aspect, for the robot to be considered as a teammate, is to be socially acceptable, where the robot must be aware of the expectation of the human teammates and act accordingly. The challenge is to model the human's expectation of the robot and align the robot's behavior with this expectation. In Dragan and Srinivasa (2013), the approach is to generate "legible" motions that show the robot's intent implicitly Knepper *et al.* (2017). Another approach is to train the team sufficiently so that each team member would maintain a good prediction model of each other's behavior Nikolaidis *et al.* (2014). These approaches, however, work only in relatively simple and repetitive domains. For more complex domains, the robot is required to learn and model the human expectation from interactions Chakraborti *et al.* (2017a); Zhang *et al.* (2016). Using these models, the robot will be able to anticipate human expectations in order to remain comprehensible to the human, or to choose a behavior that is the least interruptive when it does not match perfectly with the expectation. This ability is well known to promote sustainability of teaming situation awareness Cooke (2015) in human-human teams. While this work is inspired by Zhang *et al.* (2017), we significantly extend the framework to consider interactive human-robot teaming instead of having the human being merely an observer.

2.2 Background

2.2.1 Planning

A planning problem can be formulated as a tuple $\mathrm{P} = \langle F, M, I, G \rangle$, where F is a set of fluents, M is the domain model which consists of a set of actions A and a cost function C. $I \subseteq F$ is the initial state and $G \subseteq F$ is the goal state. Each action in A is a tuple consists of preconditions and effects. C assigns a non-negative cost to each action. Given a planning problem with I and G, the objective is to synthesize a plan $\pi = \langle a_1, a_2, ..., a_n \rangle$ which consists of a sequence of actions that lead to the goal state from the initial state. The cost $c(\pi)$ is the sum of the costs of all the actions in the plan π.

2.2.2 Plan Explicability

The explicability of a plan Zhang *et al.* (2016) is correlated with a mapping of high-level tasks (as interpreted by humans) to the actions performed by the robotic agent. The demand for generating explicable plans is due to the inconsistencies between the robot's model and the human's interpretation of the robot model (which captures the human's expectation of the robot). To formalize the explicable planning problem, consider the setting with two models where M_R is the robot model and $\widetilde{M_R}$ is the human's interpretation of M_R. For a given initial and goal state pair, $\langle I, G \rangle$, let π_{M_R} be a plan generated by the robot using M_R, and $\pi_{\widetilde{M_R}}$ be the plan of the human's expectation using $\widetilde{M_R}$. An explicable plan in M_R is a plan π_{M_R} that minimizes the weighted sum of plan cost of π_{M_R} and the plan distance between π_{M_R} and $\pi_{\widetilde{M_R}}$. It can be written as:

$$argmin_{\pi_{M_R}} cost(\pi_{M_R}) + \alpha \cdot dist(\pi_{M_R}, \pi_{\widetilde{M_R}}) \tag{2.1}$$

where *cost* returns the cost of a plan, *dist* computes the distance between two plans, and α denotes the relative weight. In Eq. (2.1), $\widetilde{M_R}$ is often unknown and *dist* needs to be specified. To deal with this, in Zhang *et al.* (2016), the distance between the two plans is approximated using a CRF model, where a labeling scheme is used to map the human interpretations of the robot's actions as task labels to the robot actions in π_{M_R}. Then the *dist* function is defined as a composition of two functions as shown in Eq. (2.2), where F is a domain independent function that takes plan labels as its input and $L^\star(\pi_{M_R})$ is a labeling scheme that maps task labels to the actions in M_R.

$$dist(\pi_{M_R}, \pi_{\widetilde{M_R}}) = F \circ L^\star(\pi_{M_R}) \tag{2.2}$$

Using a CRF model to learn the labeling scheme $L^\star$, Eq. (2.2) becomes:

$$\begin{aligned} &argmin_{\pi_{M_R}} cost(\pi_{M_R}) + \\ &\qquad \alpha \cdot F \circ L_{CRF}(\pi_{M_R} \mid \{S_i \mid S_i = L^\star(\pi^i_{M_R})\}) \end{aligned} \tag{2.3}$$

where $L_{CRF}(\pi_{M_R})$ is the learned CRF model of $L^\star$ and $\{S_i\}$ is the training data.

Plan Explicability: Given a robot plan π in M_R

$$\pi = \langle a_0, a_1, a_2, ..., a_N \rangle \tag{2.4}$$

where a_0 is the starting action and there are N actions in π, and a set of action labels T given by

$$T = \{T_1, T_2, ..., T_M\} \tag{2.5}$$

where M is the number of labels, we can first apply $L^\star_{CRF}$ to obtain the label sequence, L_π. The explicability score of π is computed based on L_π. The explicability measure as in Zhang *et al.* (2017) is defined as follows:

$$F_\theta(L_\pi) = \frac{\sum_{i \in [1,N]} 1_{L(a_i) \neq \emptyset}}{N} \tag{2.6}$$

where $F_\theta(L_\pi) : L_\pi \rightarrow [0, 1]$ (with 1 being the most explicable), $\mathbb{1}$ is an indicator function, and F_θ is the domain independent function that converts plan labels to the final score. When the labeling process can't assign a label to an action a_i, its label $L(a_i)$ will be the empty set (implemented as a special label).

2.3 INTERACTIVE PLAN EXPLICABILITY

In our work, the robot creates composite plans for both the human and robot using an estimated human model and the robot's model, which can be considered as its prediction of the joint plan that the team is going to perform. At the same time, however, the human would also anticipate such a plan to achieve the same task, except with an estimated robot model and the human's own model.

Each problem in this domain can be expressed as a tuple $P_T = \langle I, M_R, \widetilde{M_H}, \Pi_C, G\rangle$. In this tuple, I denotes the initial state of the planning problem, while G represents the shared goal of the team. M_R represents the actual robot model and $\widetilde{M_H}$ denotes the approximate human model provided to the robot, which may also be learned Zhang *et al.* (2015a). The actual human model M_H could be quite different from $\widetilde{M_H}$ provided to the robot. Similarly, the approximate robot model from the human $\widetilde{M_R}$ may be different from the actual robot model M_R. See an illustration of the problem setting in Fig. 2.1. Finally Π_C represents a set of annotated plans that are provided as the training set for the CRF model.

2.3.1 Problem Formulation

In this work, the plan for the team will be represented by a composite plan, which is defined as follows:

Definition 2.3.1 (Composite Plan). *A composite plan π_c captures the actions performed by both the human and robot to achieve the goal and is represented as $\pi_c =$*

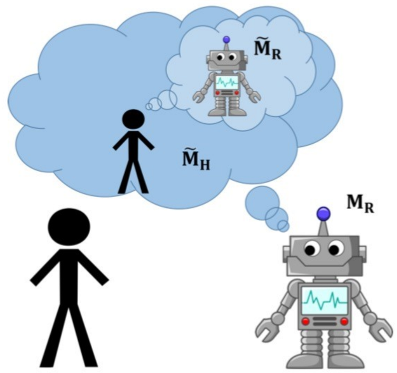


Figure 2.1: The robot's planning process is informed by an approximate human model and the robot's own model, while the human's planning process is informed by an approximate robot model and the human's own model.

$\{a_1^{\phi_1}, a_2^{\phi_2}, ..., a_i^{\phi_i}, ..., a_n^{\phi_n}\}$. *Here* $a_i^{\phi_i}$ *represents the* i^{th} *action in the plan performed by the agent* ϕ_i *(where* ϕ_i *can be either H or R).*

In our current setting, we assume that only one agent is executing its action at any given time (please see discussion section to see how we plan to relax this assumption). To generate an explicable plan, the robot needs to synthesize a composite plan that is as close as possible to the plan that the human expects. This is an especially daunting challenge, given that we have multiple points of uncertainty (e.g., $\widetilde{M_H}$ and $\widetilde{M_R}$). Nevertheless, a similar method to Zhang *et al.* (2017) can be utilized here by updating Eq. (2.1) as follows:

$$\begin{aligned} &argmin_{\pi_C^{M_R,\widetilde{M_H}}} cost(\pi_C^{M_R,\widetilde{M_H}}) \\ &\qquad\qquad + \alpha \cdot dist(\pi_C^{M_R,\widetilde{M_H}}, \pi_C^{\widetilde{M_R},M_H}) \end{aligned} \tag{2.7}$$

where $\pi_C^{M_R,\widetilde{M_H}}$ is the composite plan created by the robot using M_R and $\widetilde{M_H}$, while $\pi_C^{\widetilde{M_R},M_H}$ is the composite plan that assumed to be expected by the human. Similar to our prior work, we assume that the distance function $dist(\pi_C^{M_R,\widetilde{M_H}}, \pi_C^{\widetilde{M_R},M_H})$ can be calculated as a function of action labels for $\pi_C^{M_R,\widetilde{M_H}}$:

$$argmin_{\pi_C^{M_R,\widetilde{M_H}}} cost(\pi_C^{M_R,\widetilde{M_H}}) + \alpha \cdot F \circ L_{CRF}(\pi_C^{M_R,\widetilde{M_H}} \mid \{S_i \mid S_i = L^\star(\pi_C^i)\}) \quad (2.8)$$

Similarly, the labeling process for each action is modeled by a CRF L_{CRF} trained on a set of labeled team execution traces ($\{\pi_C^i\}$). For planning, we can easily adopt any state space planner that uses forward search, while ensuring that the heuristic itself takes into account the explicability score. To search for an explicable plan, we use a heuristic search method as shown in Algorithm 2.3.2; the heuristic is $f = g + h$, where g is the cost of the plan prefix and h is calculated as follows:

$$h = -F_\theta(L_{CRF}(CurrentState.path + rp)) \quad (2.9)$$

where + above represents concatenation and $rp = relaxedPlan(CurrentState, Goal)$. The planner algorithm is provided in Algorithm 1 and the algorithm to calculate the f value is given in Algorithm 2.3.2.

2.3.2 *Monitoring & Replanning for Interactive Teaming*

In an interactive setting, given that the robot does not have access to the complete and accurate human model nor the human's expectation of its own model, the robot will rely on replanning when the human deviates from its plan. This is discussed in more detail next. The main components of our monitoring & replanning system for training the CRF model are as follows:

- Controller: The service controlling robot actions and the planner used by the robot to achieve the goal is presented in Algorithm 3, it starts with an initial plan and performs replanning whenever the actual human action does not align with the explicable plan.

- Planner: This module is responsible for generating the composite plans. It takes the current state, combined robot and human planning model (where the human model is an approximation of the exact model), the trained CRF model, and any plan prefix. Robot calls the planner before starting any execution. If the executed plan deviates, the controller calls the planner again with updated current state and a plan prefix consisting of all actions that have been executed up to that point. The details about the planner are covered in Algorithm 1.

Algorithm 1: Algorithm for a planner to generate explicable plans

```
Input: StartState, CombinedModel, Goal, PlanPrefix
CurrentState:= StartState;
CurrentPlan := Planner(CurrentState, CombinedModel, Goal, PlanPrefix);
statePriorityQueue.add(allNeighboursWithFValue (currentState, Goal,
  PlanPrefix));
while statePriorityQueue is not empty do
    currentState := statePriorityQueue.getBestState();
    if currentState satisfies Goal then
        return currentState.path;
    else
        statePriorityQueue.add(allNeighboursWithFValue (currentState,
          Goal, PlanPrefix));
```

The controller service runs a monitoring component, which ensures that the human performs the expected action. If the state changes do not correspond to the expected action, the monitor calls the planner again to produce a new plan (replanning process). The controller feeds the planner the latest state along with the list of actions that have been executed till that point (referred to as current plan prefix). To allow the

Algorithm 2: Algorithm for allNeighboursWithFValue to calculate the f value for each of the neighboring states.

```
Input: CurrentState, Goal, PlanPrefix
neighborList := [];
for state in CurrentState.neighbors do
    rp := state.relaxedPlan(Goal);
    h := findExplicabilityScore(PlanPrefix + CurrentState.path + rp);
    g := CurrentState.path.cost;
    f := g + h;
    neighborList.add(tuple(state, f));
return neighborList;
```

CRF model to incorporate plan context from previously executed actions, it needs to consider current plan prefix as part of a larger plan containing the previously executed actions, rather than a new planning problem.

2.4 EVALUATION

To evaluate our system, we tested it on a modified synthetic first response domain, where the robots were assigned to a first-response task after a disaster occurred. In this scenario, the human's task is to team up with a remote robot that is working on the disaster scene. The team goal is to search all the marked locations as fast as possible and the human's role is to help the robot by providing high-level guidance about what the next marked location to visit. The human peer has access to the floor plan of the scene before the disaster. However, some paths may be blocked due to the disaster that the human may not know about; the robot, however, can use its sensors to detect these changes. Due to these changes in the environment, the

Algorithm 3: Algorithm for the controller service.

```
Input: CombinedModel, Goal
CurrentPlan := Planner(CurrentState, CombinedModel, Goal);
CurrentPlanPrefix := [];
for action in CurrentPlan do
    if action is robot action then
        execute action;
        add action to CurrentPlanPrefix;
    else
        executedAction := waitForHumanAction();
        add executedAction to CurrentPlanPrefix;
        if executedAction = action then
            Continue;
        else
            CurrentState := Monitor();
            CurrentPlan := Planner(CurrentState, CombinedModel, Goal,
             CurrentPlanPrefix);
```

robot might not take the expected paths of the human. Therefore, the robot delays in between performing its actions while it diverges from the expected path, so the human changes preference and the robot re-plans after that. If the human does not change her preference, however, the robot continues to visit the original room asked by the human. An example of the scenario is discussed in detail in section 2.4.1.

For data collection, we implemented the discussed scenario by developing an interactive web application using MEAN (Mongo-Express-Angular-Node) stack. We collected human trials for two different settings of the domain. The trials collected were from the people having basic knowledge of computers and not the researchers having the background understanding of this experiment. For the data collection they were only given the instructions to perform the experiment and the map where to perform the experiment.

We collected human trials in two different settings of this domain. In the first setting, the human's commands have priorities and the robot would always move toward the commanded location. Therefore, the robot needs the human command to move toward visiting the marked locations. In the second setting, the robot respects the human's commands only if it aligns with its own plan (a plan computed using FF Hoffmann and Nebel (2001) based on the robot model and approximate human model), which contains both human and robot actions. Hence, the robot may start to execute the predicted plan without the human command. However, when the human commands are aligned with the predicted plan, then the robot moves 10% faster, and the time to visit the marked locations reduces. In both settings, the human can command the robot to change the next room to be visited during the task, simply by clicking on any of the marked locations. In both settings, the robot delays 1 second before performing the next action.

After each action, the system asks the human whether the robot's action makes sense or not. This is translated later as the explicability labels. If the action of the robot makes sense, therefore it is inferred as explicable for the human. Otherwise, it is inexplicable. The costs of all of the actions are the same in both versions. For each scenario, there is a test instance with the exact same settings and instructions respectively. All scenarios were limited to four marked locations to be visited and random number of obstacles and changes in the map. All scenarios were limited to four marked locations to be visited and random number of obstacles and changes in the map. We have generated a set of 16 problems for training and 4 problems for

Table 2.1: Plan Action Distance (to the actual interactive human-robot plan) Comparison Between Human vs FF Plan and Human vs Explicable Plan

	Explicable Plan	Optimal Plan
Distance Score from Human Plan	0.9277	0.9267

testing. In both settings, if the human actions deviate from what the robot predicted, the robot starts to re-plan from the current state to the goal. Then, the robot follows the amended plan. The robot continues this behavior until all goals are met. We used FF plan in testing for both settings. We collected an initial set of 44 plan traces, which we then used to train our CRF model. The human subjects are selected from a major university in North America. They had general background knowledge about the interactive online games. Their age ranges from 21 to 33 years old. We combined the team plans alongside with their labels from both scenarios to form one model and then used the model to label the test instances. All training sets were collected from human trials, with random initial states and random goals.

2.4.1 Use Case

Consider a scenario as shown in Figure 2.3. The robot is shown in its initial state with the red box; Grey cells are the unseen obstacles which has happened due to the disaster and only the robot that is working on the scene can sense these changes. All of the tile types for each cell are represented in Figure 2.2.

In the FF plan, the human ordered the robot to visit room #4, room #1, room #2 and room #3 respectively. The highlighted yellow cells show where on the map the human set those orders accordingly. In the explicable plan, however, the human peer ordered the robot to visit room #1 first, but since the room #4 is closer than room #1, it decides to visit room #4. As the robot gets closer to room #4, it waits and hopes that the human changes her preference to room #4. After the human preference changes (on the cell marked as E2), the robot visits room #4 as it is shown in Figure 2.3 and continues to room #1. The same situation happens again on the location marked as E3 and the human changes order to visit room #1. The robot continues until all of the rooms are visited.

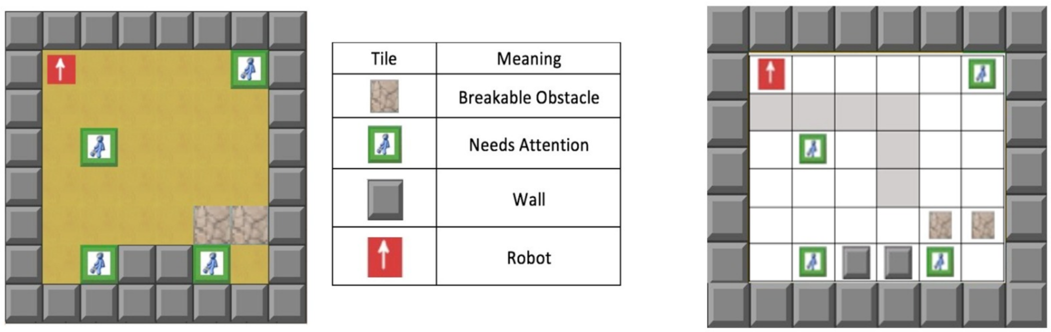


Figure 2.2: *Left:* A sample map that the human subject sees, *Middle:* the description of each tile type; *Right:* The robot's view, the gray cells are unseen obstacles due to the disaster.

Since in the explicable plan the robot senses the blocked path, it estimates the human change and re-plan as it performs actions. The plan generated by Fast Forward (FF) planner does not make this distinction. We used the trained CRF model to evaluate the explicability of the FF plan and explicable plan. The FF plan has the explicability score of 0.16 while the explicable plan has the score of 0.29 (calculated by (6)).

The FF planner produces the inexplicable plans because the planner itself is oblivious to human's preconceived notions about the robot's action model or to the actual team characteristics (in most scenarios the human would prefer the robot to incur additional costs if it can lead to a convenient and intuitive plan for the human). By using the concepts of explicability during the plan generation process, we are able to

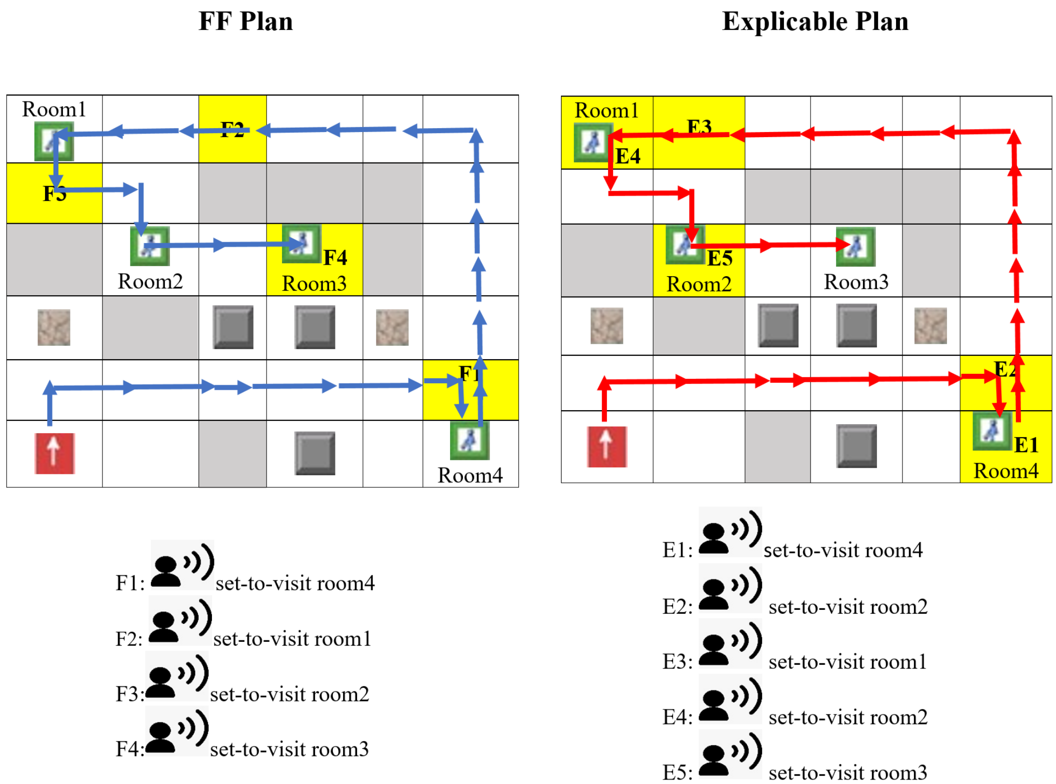


Figure 2.3: Use Case. The initial state is indicated with a white arrow inside the red box. Yellow cells refers to human actions; Set-to-visit is a human action which commands the robot to visit a marked location. Grey cells are the unseen obstacles which has happened due to the disaster and only the robot that is working on the scene can sense these changes. A sample map of the actual environment is shown in Figure 2.2.

capture any such preconceptions and model the team dynamics to match what the human would expect from her teammate.

2.4.2 Results

Table 2.2 and 2.3 show the overall and detailed interactive explicability score for plans generated by FF and our approach. We analyzed the results in terms of the explicability scores computed using Equation (2.6) while ignoring human actions.

The human plan in Table 2.2 is the plan created by the human subjects for a mixed of both settings considered. The interactive explicable plan is created using the heuristic search method mentioned in Equation (2.9).

As illustrated in Table 2.2, the explicability score of interactive explicable plan is comparable to the human plan (0.7% difference), while the explicability score difference between FF plan and human plan is 10.8%.

The results indicate that the explicability score of plans generated by our algorithm is comparable to the actual interactive plan, implying that the plans created by our algorithm can lead to more efficient collaboration in practice. One possible reason for the low explicability score of FF plan is that FF tends to create plans that are less costly while ignoring the fact that the human and robot view the environment differently and thus less costly plans in one view are also more likely to be misaligned with less costly plans in the other. Note, however, that whether the explicable plan would lead to better teaming performance requires further investigation and evaluation with actual human subjects. This will be explored in our future work.

Table 2.4 illustrates the subjective questionnaire results for both games on a 1-10 Likert scale, where 1 is the lowest and 10 is the highest score. The results of the table indicate that human subjects perceive the robot's actions, which is the result of our explicable planner contributing to the team effectiveness and they prefer to collaborate with the robot consequently.

2.5 CONCLUSIONS

In conclusion, this chapter aims to create a general way of generating explicable plans for human-robot teams, where the human is an active player. To generate an explicable plan for a human-robot team, we need not only consider the plan cost,

Table 2.2: Overall Interactive Explicability Scores

Plan Type	Interactive Explicability Score
Interactive Explicable Plan	0.833
FF Planner	0.716
Human Plan	0.803

Table 2.3: Elaborated explicability Score for Test Scenarios

Scenario #	FF Plan	interactive Explicable plan
1	0.16	0.34
2	0.18	0.30
3	0.20	0.33
4	0.28	0.38

but also the preconceptions that the human may have about the robot. Also, the robot should try to perform actions that may benefit the human plans, even if they incur a higher cost for the robot. Although we have mainly focused on two member teams, we believe that these ideas can be easily extended to larger team sizes with a few changes to the current formulation. One of the main challenges in larger team sizes would be to maintain the order in which the agents may choose to perform their actions. Another assumption we made for this work was that all action executions were sequential, it would be interesting to see if this formulation can be extended to support simultaneous action executions. It should be straightforward to extend the current formulation to support simultaneous action executions by considering joint

Table 2.4: Questionnaire Results Based on Likert Scale

Questions	
Robot contribution to team effectiveness	8.37
Human contribution to team effectiveness	6.86
Preference to collaborate with the robot	8.41

actions at any time step. Another way we may be able to achieve this would be by using temporal planners Do and Kambhampati (2003) instead of relying on sequential ones. Also, the current system assumes the provision of an approximate human planning model and relies on replanning to correct its plans whenever the human deviates from the predicted explicable plan. We could possibly explore the idea of incorporating models like capability model Zhang *et al.* (2015a) to learn such human models. A possible way this work can be further extended would be to incorporate predictability as defined by Zhang *et al.* (2015b) into the plan generation process. So instead of just focusing on generating the most explicable plans, we can try to produce plans that are both explicable and predictable. In conclusion, this work aims at introducing a way of creating plans for human-robot teams, that are naturally more explicable and more preferred by the human. Currently, the system provides the modeling of human planning preferences and relies on replanning to correct its plans whenever the human deviates from the explicable action. It would be also interesting to see if we can incorporate models like capability model Zhang *et al.* (2015a).

Chapter 3

ONLINE EXPLANATION GENERATION FOR PLANNING TASKS IN HUMAN ROBOT TEAMING

As intelligent robots become more prevalent in our lives, the interaction of such AI agents with humans becomes more frequent and essential. One of the most important aspects of human-robot interaction is for the robotic agent to provide explanations to support the rationale behind its decision or behavior Lombrozo (2006). An explanation provides justifications for the robot, which helps the human maintain trust of the robotic peer as well as a shared situation awareness Endsley (1988); Cooke (2015). Prior work on explanation generation, however, often ignores the underlying requirements of the human recipient to understand an explanation Göbelbecker *et al.* (2010); Hanheide *et al.* (2017); Sohrabi *et al.* (2011). A good explanation should be generated in a lucid fashion from the recipient's perspective Chakraborti *et al.* (2017b); Miller (2018), so that it is understood.

To address this problem, a key consideration is that the human recipient (*explainee*) may interpret an explanation differently from the robot (*explainer*) due to a different understanding of the domain. In our prior work Chakraborti *et al.* (2017b), we refer to such differences as *model differences*. The robotic agent, as a result, must ensure that the explanation makes sense in the human's model, which generates the human's expectation of the robot, so that the robot's behavior matches with the human's expectation. An explanation can then be considered as a request to change the human's model to reduce the model differences so that the robot's behavior is consistent with the updated human model. The decision-making process (including explanation generation discussed herein) in the presence of such model differences is more generally referred to as *model reconciliation* Chakraborti *et al.* (2017b); Zhang *et al.* (2017).

One remaining challenge, however, is the consideration of the mental demand of the human to understand an explanation. In most prior work on explanation generation, the human is expected to understand an explanation regardless of how much information it contains. Little discussion has been given on the ways of presenting such information. In this work, we argue that explanations, especially complex ones, should be provided in an online fashion, such that each explanation is broken into multiple parts, which are then communicated separately and intertwined with plan execution. Communicating an explanation in such a manner is expected to result in less mental workload for cognitively demanding tasks since the information is spread out so that the interpretation process becomes incremental, which is known to benefit understanding Fischer and Fischer (1979). One of the main challenges here is that the different parts of an explanation could be dependent on each other, which must be taken into account when generating online explanations. Our online explanation generation process spreads out the information while ensuring that the different parts do not introduce cognitive dissonance so that they are always perceived in a cohesive fashion.

3.0.1 Motivating Example

Let us illustrate the motivation of online explanation generation via a familiar situation involving a daily routine. Mark works at a company and has a voice assistant helping him get ready for work everyday. Usually Mark wakes up, drinks a freshly made coffee, enjoys a filling breakfast, dresses for work, and then drives to work. However, today, Mark has a meeting scheduled in the early morning so Mark needs to arrive at work earlier, a presentation at near lunch time, his car is broken, and there is no coffee beans for fresh coffee. The voice assistant knows that Mark must be reminded of these changes. However, explaining all the changes at the same time may result in unnecessary strain on Mark. In contrast, the robot first suggests Mark to prepare an instant coffee,, explaining that there is no coffee beans left. As Mark is enjoying his coffee, it reminds him to cook a light breakfast since there is an important meeting scheduled early today. As Mark is enjoying his breakfast, the voice assistant advises Mark to prepare a lunch box, since there is a presentation at near lunch time so that Mark may not have time to eat outside. After Mark is done with the lunch box, it asks Mark to call the taxi company since his car is broken. After breakfast, the assistant mentions that Mark needs to dress up today for work. When the taxi arrives, the voice assistant asks Mark to take the lunch box. Comparing the voice assistant's strategy to share all of the information at the beginning, we can see that conveying the information in an online fashion is more cognitively friendly (i.e., involving less information at a time) and hence helps with reducing strain and cognitive load Kahneman (2011); Cowan (2008). These effects are highly desirable for tasks that are cognitively demanding for humans.

In this chapter, we develop a general formulation of online explanation generation by breaking an explanation up into multiple parts to be communicated at different times during plan execution. We develop three variations of online explanation generation methods with each satisfying different "online" requirements. In the first method, the focus is for a robot to explain only plan prefixes. This is in contrast to prior offline methods where the entire plan must be explained, which allows us to break an explanation up into multiple parts with each explaining only a part of the plan. We use a model search method to ensure that the earlier parts communicated do not affect the latter parts of an explanation. In the second method, we further relax the online requirement by requiring only the very next action to be explained (if needed). The assumption here is that the actions already occurred do not affect the understanding of the robot's future actions, which holds in situations where each action is viewed independently or the human has a short cognitive span (such as in highly demanding tasks). In the third method, we relax the assumption of the uniqueness of the human's interpretation of a plan and the robot is only required to explain with respect to any such interpretation. A compilation method is developed that converts this problem into one that requires solving two planning problems. Our methods are evaluated both synthetically and with human subjects in standard planning domains. Results strongly suggest that our methods not only generate explanations that are perceived as less cognitively demanding and much preferred over the baselines but also are computationally efficient.

3.1 Related Work

The advancement of AI and its numerous applications have provided astounding benefits in many areas such as transportation, medicine, finance, education, and entertainment. And yet AI agents have thus far been limited in their ability to operate as a teammate. To be considered a teammate, an AI agent must not only achieve a given task, but also provide a level of transparency about itself to other members of the team Cooke (2015). One way to achieve this is to enable AI agents to be self-explanatory in their behaviors. Recently, the explainable AI paradigm Gunning (2017) rises as one essential constituent of AI systems. Explainable AI maintains a shared situation awareness by facilitating the human's understanding of the AI agent, which also improves the human's trust.

The effectiveness of explainable agency Langley *et al.* (2017) depends on the agent's ability to model the human's interpretation of its behavior. While there exists prior work that focuses on aligning the values Brewka (1996) or goals Andersen *et al.* (2016); Dragan *et al.* (2013), the interpretation also depends on the domain model Gong and Zhang (2020). This means that an explainable AI agent must not only model the domain, but also the human's interpretation of the domain Chakraborti *et al.* (2017a), which may be quite different. This interpretation model of the human enables the AI agent to infer about the human's expectation of itself. Using such a model, an agent can generate explicable plans Zhang *et al.* (2017); Zakershahrak *et al.* (2018); Fox *et al.* (2017), assistive actions Reddy *et al.* (2018), etc., to facilitate fluent human-robot interactions. In these methods, the AI agent substitutes a cost metric with a new metric that simultaneously considers the cost and a distance metric between the robot's behavior and the human's expectation of it. Optimizing this new metric often leads to a trade-off between the plan cost and plan interpretability.

Another way to use the human's interpretation of the domain model requires explicit communication, which has the benefit of maintaining cost optimality. The model can be used to infer about which actions in an optimal robot plan are likely to introduce misinterpretations. In some cases, simply providing the future context for those actions is sufficient Gong and Zhang (2018) to make them interpretable. Methods for analyzing the domain to identify the "*causes*" of the robot's plan (or its failures) have been studied before Göbelbecker *et al.* (2010); Hanheide *et al.* (2017); Sohrabi *et al.* (2011). These methods however assume no differences between the robot's and human's models. More recently, research work has been proposed to specifically address this issue by considering their differences and generating explanations to reduce them Chakraborti *et al.* (2017b); Miller (2018). However, all research above has been focused on generating the "*right*" explanation while ignoring the cognitive requirement of the human for understanding the explanation. In our prior work, we have studied how the ordering for presenting the information of an explanation may influence its interpretation Zhang and Zakershahrak (2019). In this work, we further argue that an explanation should be made in an online fashion for cognitively demanding tasks.

3.2 Explanation Generation as Model Reconciliation

We consider the explanation generation problem in a model reconciliation setting first introduced in our recent work Chakraborti *et al.* (2017b). The reason for this choice is that it represents a more general setting for explanation generation than those used in the previous work, which considers both the robot's (*explainer*) and human's (*explainee*) models as discussed in the related work. An illustration of the model reconciliation setting is presented in Fig. 3.1. Next, we provide a brief review of the formulations used in our setting.

Model reconciliation defines a planning setting. A planning problem is defined as a tuple (F, A, I, G) using PDDL Fox and Long (2003), which is similar to STRIPS Fikes and Nilsson (1971). $M = (F, A)$ is also referred to as the *model* in this work, where F is the set of predicates used to specify the state and A the set of actions used to update the state. Actions are associated with a set of preconditions, add and delete effects. I, G are the initial and goal states, respectively.

Definition 3.2.1 (Model Reconciliation Chakraborti *et al.* (2017b))**.** *A model reconciliation setting is a tuple* $(\pi^*_{I,G}, \langle M^R, M^H \rangle)$ $(M^R \neq M^H)$ *under a given* I, G*, where* $\pi^*_{I,G}$ *corresponds to* π_{M^R} *in Fig. 3.1 and represents the robot's behavior (plan) to be explained.*

Assuming rational agents, the $\pi^*_{I,G}$ above must satisfy $cost(\pi^*_{I,G}, M^R) = cost^*_{M^R}(I, G)$, where $cost(\pi, M)$ returns the cost of a plan π under the model M, and $cost^*_M(I, G)$ returns the cost of the optimal plan for the given initial and goal states under M.

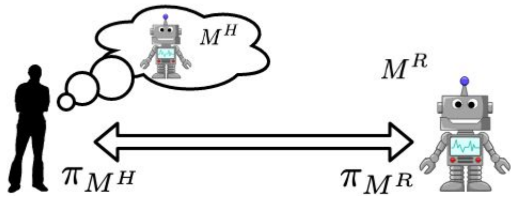


Figure 3.1: The model reconciliation setting first introduced in Chakraborti *et al.* (2017b). M^R represents the robot's model and M^H the human's interpretation model of the robot's behavior. Using M^H, the human obtains his expectation of the robot's behavior π_{M^H}. Whenever that is inconsistent with the robot's actual behavior π_{M^R} (generated by M^R), the robot explains by generating an explanation to reconcile the two models.

In other words, the robot's plan to be explained must be optimal under M^R. It is assumed that the human obtains his expectation of the robot using M^H. Hence, when the robot's behavior does not match with the human's expectation, explanations must be made. The goal of model reconciliation is to make the robot's plan $\pi^*_{I,G}$ also interpretable under the human's model M^H (i.e., generable by M^H) by reducing the differences between M^H and M^R.

To define model differences, a mapping function Γ was defined in Chakraborti *et al.* (2017b) to convert a planning problem into a set of features that fully specify the given problem. For simplicity, we modify the function here to remove the consideration of differences in the initial and goal states. As such, Γ maps any planning problem from its model space $\mathcal{M}$ to the power set of its feature space $\mathcal{F}$ (i.e., $\Gamma : \mathcal{M} \longmapsto 2^{\mathcal{F}}$) as follows:

$$\tau(f) = \begin{cases} a - has - precondition - f, & \text{if } f \in pre(a), a \in A. \\ a - has - add - effect - f, & \text{if } f \in eff^{+}(a), a \in A. \\ a - has - del - effect - f, & \text{if } f \in eff^{-}(a), a \in A. \\ a - has - cost - f, & \text{if } f = c_a, a \in A. \end{cases}$$

$$\Gamma(M = (F, A)) = \{\tau(f) | \bigcup_{a \in A} \{f | f \in \{c_a\} \cup pre(a) \cup eff^{+}(a) \cup eff^{-}(a)\}\}$$

Definition 3.2.2 (Explanation Generation Chakraborti *et al.* (2017b))**.** *An explanation in a model reconciliation setting* $(\pi^*_{I,G}, \langle M^R, M^H \rangle)$*, is a set of unit feature changes* Δ *to* M^H *such that* 1) $\Delta = \Gamma(\widehat{M^H}) \setminus \Gamma(M^H) \subseteq \Gamma(M^R)$*, and* 2) $cost(\pi^*_{I,G}, \widehat{M^H}) - cost^*_{\widehat{M^H}}(I, G) < cost(\pi^*_{I,G}, M^H) - cost^*_{M^H}(I, G)$*, where* $\widehat{M^H}$ *is the model after the changes.*

An explanation hence reconciles M^R and M^H by reducing their differences and making the cost difference between the human's expected plan $cost^*_{\widehat{M^H}}(I, G)$ and the robot's plan $cost(\pi^*_{I,G}, \widehat{M^H})$ smaller after the model updates. When the cost difference becomes 0, the robot's plan becomes optimal (and hence aligned with that) in the human's model.

Definition 3.2.3 (Complete Explanation Chakraborti *et al.* (2017b))**.** *An explanation is complete if it satisfies* $cost(\pi^*_{I,G}, \widehat{M^H}) = cost^*_{\widehat{M^H}}(I, G)$*.*

A minimal complete explanation (MCE) Chakraborti *et al.* (2017b) is defined as a complete explanation that contains the minimum number of unit feature changes.

3.3 Online Explanation Generation (OEG)

Prior work on explanation generation (including Chakraborti *et al.* (2017b)) focuses on providing the rationale behind the robot's decision making. It is often assumed that the explanation is provided in its entirety before the execution. As such, the cognitive requirement of the human for understanding the explanation is largely ignored: when complex explanations are involved (such as in cognitively demanding tasks), communicating all information at the beginning becomes impractical. In such cases, it is desirable to communicate explanations in an incremental fashion. In this chapter, we introduce *online explanation generation* to address the above issue. The key idea here is to break up an explanation into multiple parts while ensuring consistency for interpretation, and communicate them separately during plan execution. Each part of an explanation is referred to as a *sub-explanation*. A key observation that allows us to break up an explanation is that only a part of the robot's plan needs to be explained by each sub-explanation given at a specific time step. Next, we discuss three variations of OEG methods.

3.3.1 OEG for Matching Plan Prefix (OEG-PP)

In this variation, each sub-explanation is required to explain a prefix of the robot's plan, such that it is consistent with the prefix of the human's expectation of the robot's plan. The sub-explanations are made incrementally in the sense that each sub-explanation, when combined with the previous ones, explains a longer prefix of the robot's plan. The implication here is that the human's expected plan after all the sub-explanations will necessarily be the same as the robot's plan, which is the longest prefix of itself.

Definition 3.3.1 (OEG-PP)**.** *An online explanation for matching plan prefix is a* *set of sub-explanations in the form of* $\langle e_k, t_k \rangle$*, where* e_k *represents the set of unit feature changes to be made as the* k*th sub-explanation before executing step* t_k *(the* t_k*th action) of the plan, such that the following holds:*

$$\begin{aligned} \forall k > 0, \mathit{Prefix}(\pi^*_{I,G}, t_k - 1) &= \mathit{Prefix}(\pi^H_{E_{k-1}}, t_k - 1) \\ s.t. \quad \Gamma(M^H_{E_{k-1}}) &= \Gamma(M^H) \cup E_{k-1}, \\ E_{k-1} &= \bigcup_{1}^{k-1} e_i, \text{ and } E_{k-1} \subseteq \Gamma(M^R) \end{aligned} \tag{3.1}$$

where $\mathit{Prefix}(\pi, t)$ returns the prefix of a plan π up to step t (inclusive). E_k represents the union of all sub-explanations up to the kth sub-explanation and $\pi^H_{E_k}$ the optimal plan created under $M^H_{E_k}$, which denotes M^H after incorporating all the changes from e_1 to e_k. More intuitively, at any step $k-1$, the corresponding sub-explanation e_{k-1} is only responsible to explain the actions from t_{k-1} and onward until $t_k - 1$ in the robot's plan.

To generate each $\langle e_k, t_k \rangle$, the search process must consider how the sequence of model changes as a result of each sub-explanation would result in the change of the human's expectation. This allows us to convert the problem of online explanation generation to the problem of model space search as in Chakraborti *et al.* (2017b). The challenge here is that the model changes are not independent, i.e., future sub-explanations may have violated the condition in Eq. (3.1) for the earlier sub-explanations. In such cases, an online explanation may become undesirable since the human may question the robot's earlier actions at a later stage, even though they appeared reasonable. This situation would introduce cognitive dissonance that may affect the human's understanding of the robot's plan.

To address this issue, it must be ensured that the model changes in the sub-explanations e_k and onward, would not change the plan prefix that is already established up to plan step $t_k - 1$. This can be achieved by searching backward from M^R

to M^H. More specifically, given the model reconciliation setting for an explanation generation problem $(\pi^*_{I,G}, \langle M^R, M^H \rangle)$, the following process can be performed recursively to determine each sub-explanation. First, we compute the human's expected plan using M^H, which is denoted as π_H. Denote the index of the first action where $\pi^*_{I,G}$ and π_H differ as t_1, which is the timing for the first sub-explanation. To determine e_1, our search starts from M^R. It finds the largest set of model changes to M^R, denoted as $\overline{e_1}$, such that $\mathit{Prefix}(\pi^*_{I,G}, t_1) = \mathit{Prefix}(\pi^H_M, t_1)$ under any M that lies in between $M^R \setminus \overline{e_1}$ and M^R (i.e., $\Gamma(M^R) \setminus \overline{e_1} \subseteq \Gamma(M) \subseteq M^R$). In this way, we guarantee that no change in $\overline{e_1}$ can violate the condition in Eq. (3.1) once all the feature changes in e_1 are explained, which is exactly what we strive for! e_1 is then computed as the complement of $\overline{e_1}$, or $e_1 = \Gamma(M^R) \setminus (\Gamma(M^H) \cup \overline{e_1})$. Now that we have found $\langle e_1, t_1 \rangle$, we can set M^H to be $\Gamma(M^R) \setminus \overline{e_1}$, or equivalently $\Gamma(M^H) \cup e_1$ (i.e., $M^H_{E_1}$), to determine the next sub-explanation in a recursive manner. The recursion stops when the human's expected plan under $\Gamma(M^H) \cup \bigcup_{i=1}^{k} e_i$ (i.e., $M^H_{E_k}$) matches with $\pi^*_{I,G}$ for the first time, where e_k becomes the last sub-explanation.

The model space search in OEG-PP for determining the kth sub-explanation is illustrated in Fig. 3.2. In practice, this search is computationally expensive. Hence, we implement an approximate method that searches forward from $M^H_{E_{k-1}}$ for the kth sub-explanation. The search is stopped when the smallest e_k that satisfies $\mathit{Prefix}(\pi^*_{I,G}, t_k)$ $= \mathit{Prefix}(\pi^H_{E_k}, t_k)$ is found. This approach is more efficient but comes with the cost that no guarantee can be made regarding the latter sub-explanations–they may introduce violations to the condition in Eq. (3.1) for the earlier steps. If this happens, we backtrack. The implication here is that this method can no longer be used as an *online planning* method (i.e., computing the e_k's online): even though the sub-explanations are communicated online, they must be created offline.

Figure 3.2: The model space search process for the kth sub-explanation in OEG-PP. The search starts from M^R (similar to that used for MCE in Chakraborti *et al.* (2017b)) until finding the largest set of $\overline{e_k}$ (or smallest e_k) that satisfies $\textit{Prefix}(\pi^*_{I,G}, t_k)$ = $\textit{Prefix}(\pi^H_{E_k}, t_k)$, under any M that is in between $M^R \setminus \overline{e_k}$ and M^R. Each node represents a candidate model and each edge a unit feature change. The gray nodes are nodes that are not expanded in the search.

3.3.2 OEG for Matching Next Action (OEG-NA)

In this variation, we relax the requirement in Eq. (3.1) by requiring only the very next action to be interpretable at any step. The assumption here is that the human would not evaluate the robot's behavior retrospectively (or that its influence is minimal), which is reasonable in cognitively demanding tasks where humans must focus more on the current situation due to a very limited cognitive span in such cases Paas *et al.* (2003). It is also worth noting that OEG-PP and OEG-NA represent the two ends of the spectrum for online explanation generation where OEG-PP considers all actions occurred previously while OEG-NA ignores them all. It is expected that some method in between may work the best. Such analysis will be performed in our future work.

Definition 3.3.2 (OEG-NA)**.** *An online explanation for matching next action is a set of sub-explanations in the form of $\langle e_k, t_k \rangle$ such that the following is satisfied:*

$$\begin{aligned} &\forall k > 0, \pi^*_{I,G}[t_{k-1} : t_k - 1] = \pi^H_{E_{k-1}}[t_{k-1} : t_k - 1] \\ &s.t. \quad \Gamma(M^H_{E_{k-1}}) = \Gamma(M^H) \cup E_{k-1}, \\ &E_{k-1} = \bigcup_{1}^{k-1} e_i, \ and \ E_{k-1} \subseteq \Gamma(M^R) \end{aligned} \tag{3.2}$$

The search for OEG-NA naturally starts from $M^H_{E_{k-1}}$ for $\langle e_k, t_k \rangle$ since we no longer worry about matching the prefix.

3.3.3 OEG for Matching Any Prefix (OEG-AP)

One assumption made in both OEG-PP and OEG-NA is that the optimal plan for a given I, G pair is always unique. When we estimate the human's expected plan under a candidate model $M^H_{E_k}$ while searching for the kth sub-explanation, this assumption allows us to use the plan $\pi^H_{E_k}$ returned by any optimal planner, since they will always be the same. $\pi^H_{E_k}$ is then compared against $\pi^*_{I,G}$ to determine whether the e_k (incorporated into $M^H_{E_k}$) satisfies the requirements of OEG. When multiple optimal plans are present, the above check only needs to work for one of those plans. In this variation, we relax the uniqueness assumption of the optimal plans.

Definition 3.3.3 (OEG-AP)**.** *An online explanation for matching any prefix is a set of sub-explanations in the form of $\langle e_k, t_k\rangle$ such that the following is satisfied:*

$$\begin{aligned}
&\exists \pi^H_{E_{k-1}} \in \Pi^H_{E_{k-1}}\\
&\forall k > 0, \mathit{Prefix}(\pi^*_{I,G}, t_k - 1) = \mathit{Prefix}(\pi^H_{E_{k-1}}, t_k - 1)\\
&\qquad s.t. \quad \Gamma(M^H_{E_{k-1}}) = \Gamma(M^H) \cup E_{k-1},\\
&\qquad\qquad E_{k-1} = \bigcup_{1}^{k-1} e_i,\ and\ E_{k-1} \subseteq \Gamma(M^R)
\end{aligned} \tag{3.3}$$

where $\Pi^H_{E_{k-1}}$ represents the set of all optimal plans under $M^H_{E_{k-1}}$. A similar definition can be provided for OEG-NA after removing the uniqueness assumption.

To check for a candidate e_k, according to our previous discussion, we need to search for the largest set of model changes to M^R, denoted as $\overline{e_k}$, such that $\mathit{Prefix}(\pi^*_{I,G}, t_k)$ $= \mathit{Prefix}(\pi^H_{E_k}, t_k)$. An obvious solution to OEG-AP is to obtain $\Pi^H_{E_k}$ by computing all the optimal plans under $M^H_{E_k}$. This approach however is computationally expensive. Instead, we implement a compilation approach. In this approach, to check the above condition, we only need to solve two planning problems. The first planning problem is simple: finding an optimal plan $M^H_{E_k}$ under the given I, G. We denote the returned plan by any optimal planner as $\pi^H_{E_k}$ as usual. The second one is trickier in which we need to obtain a problem under $M^H_{E_k}$ such that *any* optimal plan would have to satisfy the condition $\mathit{Prefix}(\pi^*_{I,G}, t_k) = \mathit{Prefix}(\pi^H_{E_k}, t_k)$. We denote the plan returned as $\hat{\pi}^H_{E_k}$. Now, we know that if the cost of $\hat{\pi}^H_{E_k}$ is equal to that of $\pi^H_{E_k}$, there must exist an optimal plan in the human's model that matches the prefix of the robot's plan. Otherwise, no such plan exists and a sub-explanation must be made. Hence, the key here is to ensure that a given plan prefix is always satisfied in a compiled model.

It turns out that this is not difficult to achieve. For all $a_i, a_{i+1} \in \mathit{Prefix}(\pi^*_{I,G}, t_k)$, where a_i, a_{i+1} are two consecutive actions in $\pi^*_{I,G}$, the compilation can be achieved by adding a predicate p_i to a_i as an effect, which is also added as a precondition for a_{i+1}. a_{i+1}, in its turn, adds p_{i+1} as an effect which is a precondition for a_{i+2}, etc. The search process is the same as that described in OEG-PP. The search stops when any optimal plan in the human's updated model matches the robot's plan. In contrast to OEG-PP, the plan that is returned by an optimal planner under the human's model after an OEG-AP may not be exactly the robot's plan.

3.4 Evaluation

We evaluate our methods for online explanation generation both synthetically and in simulation with human subjects, and compare them with variations of minimally complete explanations (MCE) Chakraborti *et al.* (2017b) as baselines. For the synthetic evaluation, our aim is to show how online explanations differ from MCEs. We evaluate our methods and MCE on 10 different problems across the IPC Rover and Barman domains IPC (2019). For human subject study, our aim is to verify the following:

- *Online explanations reduce mental workload and improve task performance.*

Pr.	OEG-PP		OEG-NA			OEG-AP			MCE	
	$\sum e_k/\lvert e_k\rvert$	Time	$\sum e_k/\lvert e_k\rvert$	Dist.	Time	$\sum e_k/\lvert e_k\rvert$	Dist.	Time	$\lvert E\rvert$	Time
					Rover					
P1	3/1.5	8.9	7/1.2	0.40	17.9	2/1.0	0.40	6.9	3	28.9
P2	5/1.7	22.3	7/1.4	0.11	42.6	3/1.0	0.11	18.3	5	150.5
P3	6/1.5	18.7	8/1.1	0.07	21.3	3/1.0	0.07	1.6	5	176.2
P4	6/1.5	51.0	8/1.3	0.13	94.8	5/1.3	0.13	45.4	6	314.2
P5	5/1.7	54.8	8/1.3	0.14	106.7	3/1.5	0.14	50.4	4	272.8
					Barman					
P1	5/1.3	43.0	5/1.3	0.91	59.9	2/1.0	0.94	24.4	5	180.0
P2	5/1.0	36.2	5/1.0	1.00	33.0	3/1.0	0.90	9.4	5	38.9
P3	5/1.3	36.8	5/1.0	0.90	46.8	3/1.5	0.71	9.7	5	51.8
P4	5/1.3	78.4	5/1.0	0.84	69.0	4/1.0	0.56	20.4	5	61.9
P5	5/1.7	41.9	5/1.0	0.89	54.7	3/1.5	0.56	10.2	5	61.5

Table 3.1: Comparison of explanation size, average sub-explanation size (for online only), plan distance between $\pi^H_{E_k}$ and $\pi^*_{I,G}$ (when applicable) and time (in seconds) using the different methods for the IPC Rover and Barman domains.

A modified rover domain (Sec. 3.4.2) is used. In all evaluations, M^R is the true domain model, and M^H is created by removing model features from M^R. All results are collected on a 2.2 GHz quad core Macbook Pro with 16 GB RAM.

3.4.1 Simulation Results

Table 3.1 presents the simulation results comparing OEG-PP, OEG-NA and OEG-AP with MCE. The benefits of online explanations are clear: the average size of sub-explanations is significantly smaller than the size of MCE, although the sum of their sizes is generally larger than the size of MCE. This shows that most explanations can indeed be broken up and communicated incrementally while subject to the requirements of online explanations! The effect of OEG-AP on the size of explanations is interesting, which suggests that removing the uniqueness assumption of the optimal plan has a positive impact on explanation generation: the sum of sub-explanations has a size that is smaller than MCE. This intuitively makes sense since not all the sub-explanations in MCE may be required as long as the robot's plan is optimal in the updated human's model (but differs from the plan found there by an optimal planner). To see the influence of removing the uniqueness assumption from another angle, for both OEG-NA and OEG-AP, we evaluate how the human's expected plan ($\pi^H_{E_k}$) after the explanation (returned by an optimal planner) may be different from $\pi^*_{I,G}$ using action plan distance, which has a value between 0 (no difference) and 1 (maximum difference). For OEG-NA, this distance is generally non-zero since only the very next action is considered when making a sub-explanation. For OEG-AP, the distance is also non-zero in general but due to the non-uniqueness of the optimal plan. Computationally, OEG methods are generally a bit faster than MCE which may appear to be surprising. Some analysis reveals that this is due to the fact that the incremental search in online explanation generation in fact reduces the search space by removing candidate features to be added to M_H for later searches. For OEG-NA and OEG-AP, this may also be due in part to the fact that they often terminate earlier and before $\pi^H_{E_k}$ becomes exactly $\pi^*_{I,G}$.

3.4.2 Human Study

To test our hypothesis, we compare the explanations created by our methods with variations of MCE methods in a modified rover domain. The task is for the rover to collect and analyze soil and rock samples, take pictures of targets, and send them to the lander. To ensure that the performance difference is not solely due to breaking up the information, we implement another baseline that randomly breaks up an MCE into multiple parts and communicates each part separately so that they are uniformly distributed through the plan execution (referred to as MCE-R).

We conducted our experiment using Amazon Mechanical Turk (MTurk) with a 3D simulation of the rover domain (see Fig. 3.3). The subjects were first given an introduction to the rover domain and the task they were supposed to help with. In the experiment, we deliberately removed certain information from the introduction. In particular, we did not inform them that the storage space and memory of the rover is limited, the camera must be calibrated, and calibrated with respect to the target before taking an image. These introduced the differences between M^H and M^R.

Each subject was given a 30-minute limit to finish the task. Explanations were provided using plain English language and the rover actions were depicted using GIF images in the 3D simulation as the rover executed the plan. The human subject acted as the rover's supervisor, and was asked to determine whether each of the rover's action was questionable or not. Random actions were added into the plan to make sure that the subject must question some actions to perform well. Each subject was only allowed to perform the task for one setting (OEG-PP, OEG-NA,

Figure 3.3: The 3D visualization of the modified rover domain. There are four robots on Mars, each has a different camera resolution and sampling equipment. The mission is to sample soil, rock and take images at different locations and communicate it to the lander shown on the right side of the picture.

OEG-AP, MCE, or MCE-R) to reduce the influence of learning from repeated runs. To simulate highly demanding tasks, we have incorporated three spatial puzzles as secondary tasks. At the end of the study, the subjects were provided the NASA TLX to evaluate the workload NASA (2019) under several categories Tsang and Velazquez (1996).

Results: We created the surveys using Qualtrics and recruited 150 human subjects on MTurk, with 30 subjects for each setting. To improve the quality of the responses, we set the criteria that the worker's HIT acceptance rate must be greater than 98%. After filtering out invalid responses (that failed to identify the 2 purposely inserted random actions out of a total of 30 actions in the plan), we obtained 94 valid responses in total: 19 for each of MCE-R and MCE, 20 for OEG-PP, and 18 for each of OEG-NA and OEG-AP. Their ages ranged from 18 to 70, and 29.8% of them were

	MCE-R	MCE	OEG-PP	OEG-NA	OEG-AP	Ground Truth
Accuracy	0.746	0.804	**0.858**	**0.852**	**0.872**	
# Actions	8.789	7.263	**5.250**	**5.330**	**4.940**	2.0/30

Table 3.2: The accuracy and number of questionable actions based on the subjects' feedback for the five settings.

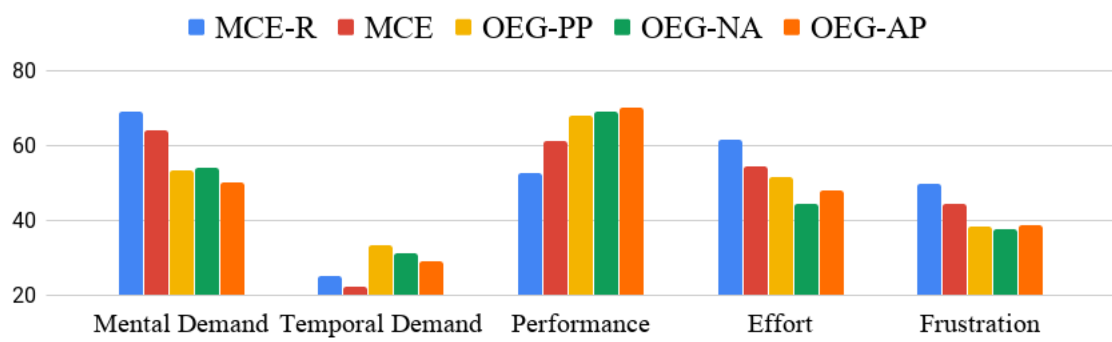


Figure 3.4: Comparison of TLX categories for the five settings.

female.

The results show that OEGs in general performed significantly better than the baselines in both objective (Table 3.2) and subjective measures (Fig. 3.4). Table 3.2 shows that the numbers of questionable actions are significantly lower for OEGs than MCEs (with p-values < 0.001). This indicates that the subjects had more trust towards robots in the OEG settings. The accuracy for identifying the correct actions (questionable vs. non-questionable) is also higher for OEGs (with p-values < 0.001). Among the three OEG methods, OEG-AP performed the best but no significant differences were observed in either the objective or subjective measures. This seems to suggest that the performances were dominated mainly by the average size of sub-explanations, which did not vary much among the OEGs (i.e., $\sum_{e_k} /|e_k|$: OEG-PP 6/1.5, OEG-NA 5/1.25, OEG-AP 3/1.0, MCE 5/NA, MCE-R 5/1.0).

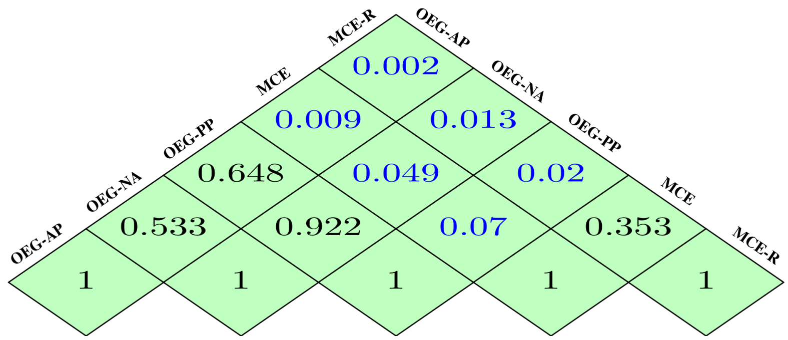


Figure 3.5: p-values for the weighted sum of the subjective measures, with weights 1.0 for all TLX categories.

It is worth noting that MCE-R performed *worse* than MCE objectively with p-values 0.043 and 0.028 respectively for the two measures in Table 3.2, which suggests that the performance difference was unlikely due to simply breaking up the information, thus confirming the usefulness of OEGs. The subjective measures in Fig. 3.4 for the most part reaffirm the conclusions. Due to intertwining explanations with plan execution, OEGs are expected to create more temporal demand. The p-values for the subjective measures are presented in Fig. 3.5. The results indicate statistically significant differences between OEGs and MCEs. The group-wise p-value is 0.0068 between OEGs and MCEs.

3.5 Conclusions

In this chapter, we introduced a novel formulation for explanation generation that was focused on reducing the mental workload for the human to interpret an explanations. We took a step further from prior work, which considered only the correct explanations, by proposing explanations that were also easily understandable. We provided three methods and evaluated them both in simulation and with human subjects. Results confirmed that they improved task performance and reduced mental workload.

ACKNOWLEDGMENT

This research is supported in part by the NSF grant IIS-1844524, the NASA grant NNX17AD06G, and the AFOSR grant FA9550-18-1-0067

Chapter 4

ORDER MATTERS: GENERATING PROGRESSIVE EXPLANATIONS FOR PLANNING TASKS IN HUMAN-ROBOT TEAMING

As robots start to benefit a diverse set of domains, human-robot interaction has evolved to be an increasingly important subject. In human-robot teaming, in particular, it is desired that the interaction occurs in a coherent manner that is observed in human-human teaming Chakraborti *et al.* (2017a); Cooke (2015). Similar to a human teammate, a robotic agent is required to not only understand its human partners, but also explain its own decisions or behaviors when necessary. Explanations in a teaming context provide the rationale behind an individual agent's decision making Lombrozo (2006), and help with building a shared situation awareness and maintaining trust between teammates Endsley (1988); Cooke (2015). Although there exists prior work on generating explanations, the focus has been on generating the right explanations from the explainer's perspective rather than good explanations for the explainee Göbelbecker *et al.* (2010); Hanheide *et al.* (2017); Sohrabi *et al.* (2011).

Unsurprisingly, the right explanation may not necessarily be a good explanation–anyone with teaching or mentoring experience would share the sympathy. Such dissonance between the explainer and explainee may be a result of various inconsistencies, such as information asymmetry or different cognitive capabilities, just to name a few. These inconsistencies may be summarized as *model differences*–the differences between the cognitive models that govern the generation and interpretation of an explanation, respectively, for the explainer and explainee Chakraborti *et al.* (2019). When these two models are the same, as is assumed in most prior works, an explanation from the perspective of the explainer would be not only correct but also perfectly understandable to the explainee, that is, as if the explanation were made to the explainer himself. The more general case when the models differ has also been investigated Chakraborti *et al.* (2017c); Zakershahrak *et al.* (2019) under the model reconciliation setting, where the focus is on explaining domain model differences such that the two models become more compatible. One remaining challenge in explanation generation, however, is to account for the differences in the cognitive capabilities to understand an explanation.

In this work, we take a step further by generating explanations while considering the differences between the cognitive capabilities of the explainer and explainee. This is especially relevant to human-robot teaming since robots are frequently deployed to situations that require high cognitive and computational abilities that we do not have. To accommodate this, the motivation here is to generate explanations that reduce the cognitive effort required to understand them for the explainee. In this work, we focus on the influence of the order of information. In a moderately complex domain, making an explanation is not an instantaneous effort; instead, information must be conveyed in small parts sequentially. Our proposal in studying the order of information is inspired by studies in psychology and education on the limitations of human cognitive systems Ericsson and Smith (1991); Kahneman (2011) and progression in learning Schwarz *et al.* (2009). Consequently, we term our approach *progressive explanation generation*. Consider the following example of a conversation between two friends, which illustrates the importance of providing information in a proper order when making an explanation:

```
Amy: Let's go to the outlet today.
Monica: My car is ready.
Amy: Great!
Monica: The rain will stop soon.
```

```
Amy: Wonderful!
Monica: By the way, today is a holiday
(shops closed).
Amy: You are telling me now!
Monica: Let us go to the central park!
Amy: ...
```

Such cognitive dissonance illustrated above occurs frequently in our lives and it is our aim in this work to avoid similar situations when a robot is making an explanation to you. The challenge lies in modeling the humans' preferences for information order in receiving such explanations to assist understanding. To this end, a general formulation based on goal-based Markov Decision Processes for generating progressive explanation is presented given the sequential information communication in an explanation. We propose to learn a quantification of the cognitive effort for each step as a reward function in an inverse reinforcement framework Ng and Russell (2000); Abbeel and Ng (2004); Ziebart *et al.* (2008). We set out to validate the following hypothesis:

- H1. Our learning method can learn about the humans' preferences in receiving explanations.

Both domain-dependent and domain-independent features are used in learning based on explanations provided via human subject studies. We evaluated first on a scavenger-hunt domain. Upon analyzing the results, however, it revealed something more fundamental: the preferences arise strongly from both domain dependent and independence features. The correlation with domain independent features pushed us to verify this result further in an escape room domain. The strong weights on domain independent features, which capture plan changes during the explanation process, implies that understanding an explanation is a dynamical process:

- H2. Humans replan ***dynamically*** to understand during an explanation instead of after an explanation for moderately and highly complex tasks.

Results confirmed our hypothesis that the process of understanding an explanation was a dynamic process. The human preference that reflected this aspect corresponded exactly to the progression for knowledge assimilation hidden deeper in our cognitive process. Our results will benefit the design of robots that make explanations across various domains since such a preference is domain independent. The last hypothesis is about the effectiveness of progressive explanations:

- H3: Progressive explanations reduces cognitive load and improves task performance.

Comparison with two baseline methods validated H3. We showed that the progressiveness in explanations corresponded well to the "progressiveness" of the curve on domain independent features.

4.1 Related Work

Explainable AI Gunning (2017) is increasingly considered to be an important paradigm for designing future intelligent agents, especially as such systems begin to constitute an important part of our lives. The key requirement of explainable agency Langley *et al.* (2017) is to be "*explainable*" to the human partners. To be explainable, an agent must not only provide a solution to achieve a goal, but also make sure that the solution is perceived as such by its human peers. A determinant here is the human's interpretation of the agent's behavior. It is critical to take careful steps to avoid situations where the agent's assistance would be interpreted as no more than an interruption, which resulted in the pitfall of earlier effort in designing intelligent assistants, such as the loss of situation awareness and trust Endsley (2016); Langfred (2004).

The key challenge to explainable agency hence is the ability to model the human cognitive model that is responsible for interpreting the behaviors of other agents Chakraborti *et al.* (2017a). With such a model, there are different ways to make the robot's behavior explainable. One way is to bias the robot's behavior towards the human's expectation of it based on the human's cognitive model. Under this framework, a robot can generate legible motions Dragan and Srinivasa (2013) or explicable plans Zhang *et al.* (2017); Zakershahrak *et al.* (2018). Essentially, the robot sacrifices the plan quality to respect the human's expectation–the resulting plan is often a more costly plan. Another way is to provide a forewarning of the robot's intention before execution, such as for persuasion Petty and Cacioppo (1979). In Gong and Zhang (2018), the approach there is to provide additional context to help explain the robot's decision. The third way, which is the most relevant to ours, is for the robot to explain its decision via explanations Göbelbecker *et al.* (2010); Hanheide *et al.* (2017); Sohrabi *et al.* (2011). The benefit of generating explanations, compared to generating explainable plans, is that the robot can keep its original (and optimal) plan. However, as mentioned earlier, the focus there is often on providing the rationale behind the explainer's decision making, while largely ignoring the explainee. In Chakraborti *et al.* (2017c), this gap is addressed by considering explanation generation as a model reconciliation problem, which takes into account the explainee's model. Although the cognitive requirement is implicitly considered, the aim there is to reconcile (i.e., reduce) the differences in domain models, so that the robot's plan would be interpretable also in the model of the explainee.

The idea of progressive explanation generation is connected to heuristics in the planning literature for reducing replanning effort Fox *et al.* (2006). However, the focus there is on system effort Likhachev *et al.* (2005). Our results, however, suggest that these heuristics may have a deeper connection to the human cognitive process in decision-making domains. The idea in generating progressive explanations to be less intrusive bears some similarities to the idea of nudging the human towards a new path Lien *et al.* (2004) or providing constant and non-intrusive reminders for performing various tasks Maxwell *et al.* (1999). The general idea is to facilitate "*smooth*" or socially acceptable Miller (2018) interactions, whether physical or cognitive. We implement a similar idea here for explanation generation by reducing the cumulative cognitive effort required. We learn a quantification of the cognitive effort required at each step of an explanation in an inverse reinforcement learning framework. based on both domain-independent and domain-dependent features, whose weights are learned in an inverse reinforcement learning framework Ng and Russell (2000); Abbeel and Ng (2004); Ziebart *et al.* (2008).

4.2 Model Reconciliation

We base our work on a general model reconciliation setting for explanation generation that considers both the models of the explainer and explainee, which is introduced in Chakraborti *et al.* (2017c). As shown in Fig. 4.1, the human uses M^H to generate her expectation of the robot's behavior, while the robot's actual behavior is being created using the robot's model M^R, which is different from M^H. Therefore, π_{M^R}, which is the plan created from M^R, could also be different with π_{M^H}, which is the plan created from M^H. Whenever these two plans differ, the robot's plan must be explained.

Definition 1 (Model Reconciliation Chakraborti *et al.* (2017c))**.** *A model reconciliation setting is a tuple* $(\pi^*_{I,G}, \langle M^R, M^H \rangle)$*, where* $cost(\pi^*_{I,G}, M^R) = cost^*_{M^R}(I, G)$.

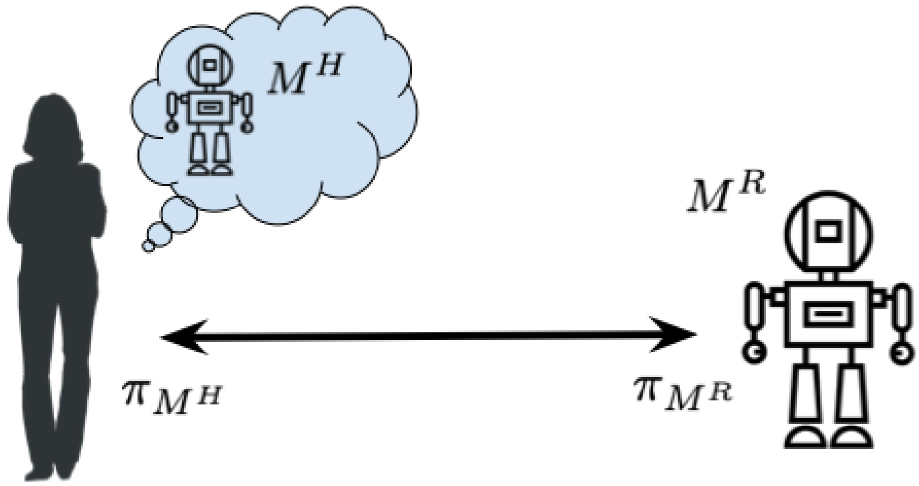


Figure 4.1: Explanation generation as model reconciliation Chakraborti *et al.* (2017c). M^R denotes the robot model and M^H denotes the human model that is used to generate her expectation of the robot's behavior (π_{M^H}). When the expectation does not match the robot's behavior, π_{M^R}, explanations must be generated.

where $\pi^*_{I,G}$ is the robot's optimal plan to be explained. $cost(\pi^*_{I,G}, M^R)$ is the cost of the robot's plan under the model M^R. $cost^*_{M^R}(I, G)$ returns the optimal plan given the initial state and goal state pair using M^R. Therefore, the constraint in the Definition 1 ensures that the robot's plan is optimal in its own model.

In this setting, the robot must generate an explanation to modify the human's model M^H such that $\pi^*_{I,G}$ becomes explainable in the human's modified model (denoted as $\widehat{M^H}$) after the reconciliation. As a result, an explanation for a model reconciliation setting can be considered as requesting changes to the model of the human. Note that making an explanation may also lead to an error report if it is identified that the robot's model was incorrect.

To capture the model changes, a model function $\Gamma : \mathcal{M} \rightarrow 2^F$ is defined to convert a model to a set of model features Chakraborti *et al.* (2017c), where $\mathcal{M}$ is the model space and F the feature space. In this way, one model can be updated to another model with editing functions that change one feature at a time. The set of feature changes is denoted as $\Delta(M_1, M_2)$ and the distance between two models as the number of such feature changes is denoted as $\delta(M_1, M_2)$. In this work, we assume that the model is defined in PDDL Fox and Long (2003), an extension of STRIPS Fikes and Nilsson (1971), where a model is specified as a tuple $M = (D, I, G)$. The domain $D = (F, A)$ is comprised of a set of predicates, F, and a set of actions, A. F is used to specify the state of the world. Each action $a \in A$ changes the state of the world by adding or deleting predicates. Therefore, an action can be represented as $a = (pre(a), eff^+(a), eff^-(a), c_a)$; where $pre(a)$ denotes the preconditions of the action a, and $eff^+(a), eff^-(a)$ indicate add and delete effects, respectively, and c_a is the cost of the action. For example, a very simple model for Amy in our motivating example would be:

```
Initial state: not-holiday
Goal state: happy
Actions:
OUTLET-SHOPPING
  pre: not-holiday (car-ready is-sunny)
  eff+: happy
VISIT-PARK
  pre: (car-ready is-sunny)
  eff+: happy
```

For simplicity, we use only boolean variables above. The variables in parenthesis are optional predicates that are preferred but not required. The goal is to achieve the effect of `happy`. In this example, the model, denoted as $M_{\texttt{Amy}}$, will be converted by the model function Γ to:

$\Gamma(M$_Amy) = {

```
init-has-not-holiday,
goal-has-happy,
OS-has-precondition-not-holiday,
OS-has-add-effect-happy, ...}
```

where OS above is short for OUTLET-SHOPPING. The function essentially turns a model into a set of features that fully specifies the model. Hence, changing the set of features will also change the model.

Definition 2 (Explanation Generation)**.** *The explanation generation Chakraborti* et al. *(2017c) problem is a tuple* $(\pi^*_{I,G}, \langle M^R,\, M^H\rangle)$ *where an explanation is a subset of* $\Delta(M_R, M_H)$ *such that:*
1) $\Gamma(\widehat{M^H}) \setminus \Gamma(M^H) \subseteq \Gamma(M^R)$*, and*
2) $cost(\pi^*_{I,G}, \widehat{M^H}) - cost^*_{\widehat{M^H}}(I, G) < cost(\pi^*_{I,G}, M^H) - cost^*_{M^H}(I, G)$.

where $\widehat{M^H}$ denotes the model after the changes. The first condition requires the changes to the human's model to be consistent with the robot's model. The second condition states that the robot's plan must be closer (in terms of cost) to the optimal plan after the model changes than the situation before–an explanation should have the effect of moving the expected plan closer to the robot's optimal plan.

Definition 3 (Complete Explanation)**.** *A complete explanation Chakraborti* et al. *(2017c) is an explanation that additionally satisfies* $cost(\pi^*_{I,G}, \widehat{M^H}) = cost^*_{\widehat{M^H}}(I, G)$.

A complete explanation requires the model changes to make the robot's plan also optimal in the changed human model, so that the robot's plan becomes interpretable in the human's model as well. A minimally complete explanation (MCE) is also defined in Chakraborti *et al.* (2017c), which is a complete explanation with the minimum number of unit feature changes. An example of $\widehat{M_{\texttt{Amy}}}$ (corresponds to $\widehat{M^H}$) after a minimally complete explanation is:

Initial state:

`not-holiday car-ready (+) is-sunny (+)`

Goal state: `happy`

Actions:

```
OUTLET-SHOPPING
pre: not-holiday (car-ready is-sunny)
eff+: happy
VISIT-PARK
pre: (car-ready is-sunny)
eff+: happy
```

where the strikeout denotes the feature removed and +'s denote additions. These changes correspond to the explanation made in our motivating example. In this case, the robot model, M^R, corresponds to $M_{\texttt{Monica}}$, is the same as $\widehat{M_{\texttt{Amy}}}$ after the explanation (with the model changes incorporated).

4.3 Progressive Explanation Generation

In progressive explanation generation, our focus is on how the ordering of presenting information in an explanation may affect its understanding. An explanation in our setting is naturally specified as a sequence of feature changes. Since we process information as it is received, the cumulative cognitive effort can then be considered as the sum of effort associated with understanding each feature change in a sequential order. We couple the cognitive effort for each change with a *model distance metric*, denoted as $\rho(M_i, M_{i+1})$ for the ith feature change, where M_i is the model before the i-th feature change and M_{i+1} is the model after that change. Thereby, progressive explanation generation can be defined as the following optimization problem:

Definition 4 (Progressive Explanation Generation (PEG))**.** *A progressive explana-*

tion is a complete explanation with an ordered sequence of unit feature changes that minimize the sum of the model distance metric: $\arg\min_{\Delta(\widehat{M^H}, M^H)} \sum_{f_i \in \langle \Delta(\widehat{M^H}, M^H) \rangle} \rho_i$*, where* ρ_i *is short for* $\rho(M_i, M_{i+1})$*,* i *is the index of unit feature changes, and* f_i *denotes the* i*-th unit feature change.*

The angle brackets above convert a set to an ordered set and the summation is over every unit feature change in an explanation, which is computed for before and after each unit feature change is made in a progressive fashion. The goal of PEG is to minimize the cumulative model distance metric, and thereby minimize the cognitive effort required from the explainee to understand the explanation.

4.3.1 PEG with Different Distance Heuristics

Depending on how the model-plan distance metric is defined, different explanation may be resulted. Next, we look at a few options for defining this distance, which intuitively have an impact on cognition. Search methods based on these options are provided afterwards.

Problem 1: Progressive explanation generation with

$$\rho_i = |cost^*_{M_{i-1}}(I, G) - cost^*_{M_i}(I, G)| \tag{4.1}$$

In this case, the distance at each step is characterized by the cost difference of the plans in the two models adjacent to a unit feature change, respectively. The search problem is in the model space and is expensive to solve. Here, we can take advantage of the following equation, which follows from basic arithmetic:

$$\sum_i \rho_i \geq |cost^*_{M^H}(I, G) - cost^*_{\widehat{M^H}}(I, G)| \tag{4.2}$$

The equality above holds if and only if the changes in plan cost are monotonic with respect to the index i. This also reflects the progressive nature of such explanations.

This observation leads to an efficient heuristic, where

$$h(M_i) = |cost^*_{M_i}(I,G) - cost(\pi^*_{I,G}, \widehat{M_H})| \tag{4.3}$$

Additionally, without the loss of generality, assuming that $cost^*_{M^H}(I,G) \leq cost^*_{\widehat{M^H}}(I,G) = cost(\pi^*_{I,G}, \widehat{M_H})$ is satisfied, the search process could first check adding preconditions, removing add effects, adding delete effects, or increasing action costs. Since these changes will increase the cost of the plan, they will more likely lead to faster search process.

Theorem 1: The heuristic described above is admissible and consistent for problem 4.1.

Proof: It can be easily verified that $h(M_i) = |cost^*_{M_i}(I,G) - cost(\pi^*_{I,G}, M_H)| \leq \sum_{k>i} \rho_k$. Hence, the heuristic above is admissible. For consistency, notice that $|cost^*_{M_{i-1}}(I,G) - cost(\pi^*_{I,G}, \widehat{M_H})| = h(M_{i-1}) \leq \rho_i + h(M_i) = |cost^*_{M_{i-1}}(I,G) - cost^*_{M_i}(I,G)| + |cost^*_{M_i}(I,G) - cost(\pi^*_{I,G}, \widehat{M_H})|$. *Problem 2*: Progressive explanation generation with

$$\rho_i = d(\pi^*_{M_{i-1}}(I,G), \pi^*_{M_i}(I,G)) \tag{4.4}$$

where $d(\pi^*_{M_{i-1}}(I,G), \pi^*_{M_i}(I,G))$ denotes the minimum editing distance between the two optimal plans that are created in M_{i-1} and M_i, respectively. Note that $\pi = \pi^*(I,G)$, which is the robot plan to be explained. Similarly, we can apply the following heuristic:

$$h(M_i) = d(\pi^*_{M_i}(I,G), \pi^*_{I,G}) \tag{4.5}$$

Theorem 2: The heuristic described above is admissible and consistent for problem 4.4. *Proof*: The plan editing distance clearly satisfies $\sum_i d_i \geq d(\pi^*_{M_i}(I,G), \pi^*_{I,G})$, since the distance metric is positive and symmetric. A similar proof follows from

Theorem 4.1.

There are many other ways such model-plan distances maybe defined. For example, one may prefer more significant changes to the model at the beginning than later in the explanation. Also, instead of plan editing distance, you may consider other common plan distance metrics, such as action,state, and causal link distances [9]. Another interesting consideration is the influence of plan hierarchies [7], [25]. For example, one may consider aggregating similar feature changes into the same explanation step, which introduce similar changes to the plan. The focus could also be on the changes to plan hierarchies.

Given the heuristics, the planning methods can be implemented as standard A^* searches. At each step, the search algorithm can choose a unit feature change from all possible changes that satisfy condition 1 in Definition 2. As discussed, we may choose to first consider the changes that are more promising. The search can easily incorporate other considerations such as conciseness. This is especially useful for cases when some feature changes do not affect the plans generated or their costs. In such cases, progressive explanations may include those unnecessary changes. This can be addressed by adding to the g value a small cost per every change made. This approach is tested on five IPC rover domain problems for each problem and results of this section is presented in Table 4.1 using an intel processor 2.7 GHz, Quad-Core i7 and 16 GB of RAM.

4.3.2 Learning the Model Distance Metric

In the previous section, we introduced different distant-metrics for the model-plan to provide the explanations. In this section, we want to extend our approach further, by learning a distance metric for PEG based on the human preferences in a human-robot teaming scheme.

Pr.	Total Features (# Missing Features)	PEG Size	$\sum_i \rho_i$	Time(s)
P1	75(8)	4	71	5.37
P2	75(6)	1	64	4.13
P3	75(6)	3	27	3.56
P4	75(5)	3	64	12.25
P5	75(10)	3	484	103.92
P1	75(8)	2	16	3.24
P2	75(8)	3	8	4.09
P3	75(6)	5	21	3.35
P4	75(6)	3	10	2.72
P5	75(5)	5	22	12.45

Table 4.1: Simulation results using heuristics of Problem 1 and 2 over five scenarios of IPC rover domain.

To learn the model distance metric for PEG, we formulate the problem as an inverse reinforcement learning (IRL) Ng and Russell (2000); Abbeel and Ng (2004); Ziebart *et al.* (2008) framework, where we assume the task of generating explanations can be expressed as a goal-based Markov Decision Processes (MDP). A goal-based MDP is defined by a 6-element tuple (S, A, T, R, γ, G), where S is the state space and A is the action space. The domain dynamics is represented as the transition function T that determines the probability of transitioning into state s' when taking an action a in state s (i.e., $P(s'|s, a)$). R is the reward function and the goal of the agent is to maximize the expected cumulative reward. γ is the discount factor that encodes the agent's preference of current rewards over future rewards. G is a set of goal states

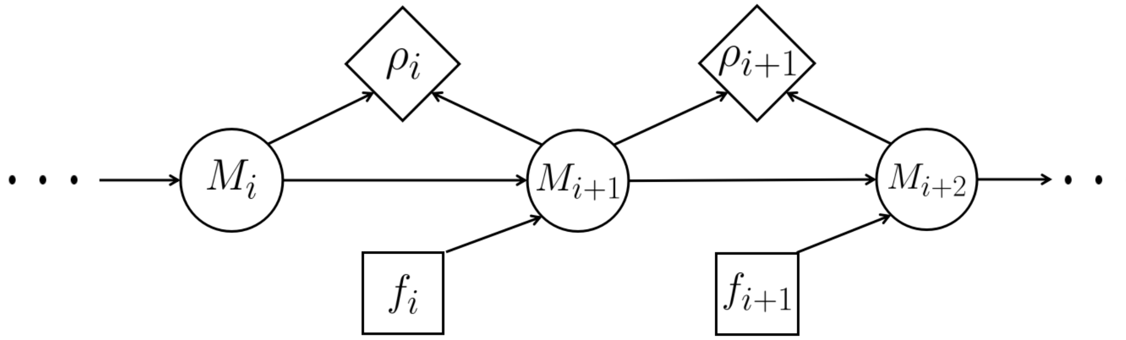


Figure 4.2: Illustration of the MDP that underlies PEG. At each time step, the human's model M_i serves as the state. When the robot provides a unit feature change f_i (as part of the explanation) to the human, the model changes according to f_i to be the next state, M_{i+1}. The model distance metric ρ_i, which is short for $\rho(M_i, M_{i+1})$, captures the cognitive effort required to understand f_i.

where for each $g \in G$, $T(g, a, g) = 1, \forall a \in A$. We chose goal-based MDP since in each scenario, although the start state could be the same, the goal could be different and therefore the policy would be different.

Fig. 4.2 demonstrates the MDP that underlies PEG. In our work, the state space S is the set of all possible human models and the action space A is the set of all possible unit feature changes. The transition function T captures the probability that the human model would be updated to M' when the human model is M and the robot explains f to her (i.e., $P(M'|M, f)$). The model distance metric ρ serves as the reward function, which depends on both the current and updated human models.

4.3.3 Applying IRL

Following prior work on IRL Ng and Russell (2000); Abbeel and Ng (2004); Ziebart *et al.* (2008), we define the distance metric as a linear combination of a set of weighted features:

$$\rho(M, M') = \sum_i \theta_i \cdot \psi_i(M, M') = \Theta^T \Psi(M, M')$$

where $\Psi = \{\psi_1, \psi_2, \ldots, \psi_k\}$ is the set of features with respect to state pair (M, M'). $\Theta = \{\theta_1, \theta_2, \ldots, \theta_k\}$ is the set of weights corresponding to the features.

Given a set of traces in a domain as a set of explanations (each is a sequence of unit features changes), which are obtained from human subjects, our goal is to learn the model distance metric ρ, which in turn requires us to learn the weights Θ given a set of features. Since noise is expected in the traces, we learn the weights by maximizing the likelihood of the traces using MaxEnt-IRL Ziebart *et al.* (2008) as follows:

$$\begin{aligned} \Theta^* = \arg\max_{\Theta} \mathcal{L}(D) &= \arg\max_{\Theta} \frac{1}{|D|} \log P(D|\Theta) \\ &= \arg\max_{\Theta} \frac{1}{|D|} \sum_{G \in \mathcal{G}} \sum_{\widehat{\zeta_G} \in D_G} \log P(\widehat{\zeta_G}|\Theta) \end{aligned} \tag{4.6}$$

where D is the training data set, $\mathcal{G}$ the collection of goal sets G for different scenarios. $\widehat{\zeta_G} = (M_0, f_1, M_1, \ \ldots, \ f_n, M_n)$ is an explanation for achieving G with ordered feature changes provided by human subjects in a subset D_G. It consists of the initial human model (i.e., $M_0 = M^H$), unit feature change and the updated model at each time step. To mitigate the ambiguity that the distribution of the traces may introduce preference for some traces over others, the principle of maximum entropy Ziebart *et al.* (2008) is employed to define the distribution over all the possible traces for a specific goal (i.e., G):

$$P(\zeta_G|\Theta) = \frac{e^{\rho(\zeta_G)}}{\sum_{\zeta_G} e^{\rho(\zeta_G)}} \tag{4.7}$$

where

$$\rho(\zeta_G) = \Theta^T \Psi(\zeta_G) = \sum_{(M,M') \in \zeta_G} \Theta^T \Psi(M, M')$$

Take Equation 4.7 into Equation 4.6, the optimization becomes:

$$\Theta^* = \arg\max_{\Theta} \frac{1}{|D|} \sum_{G \in \mathcal{G}} \sum_{\widehat{\zeta_G} \in D_G} \Big(\Theta^T \Psi(\widehat{\zeta_G}) - \log \sum_{\zeta_G} e^{\Theta^T \Psi(\zeta_G)} \Big) \tag{4.8}$$

Note that $\widehat{\zeta_G} \in D_G$ in the first term above represents a trace in the training data set while ζ_G in the second term above refers to *any* possible trace of the domain. Since Equation 4.8 is convex, we use a gradient-based method to learn Θ and divide the traces into pairs of human models as in Ziebart *et al.* (2008):

$$\nabla_\Theta \mathcal{L} = \frac{1}{|D|} \sum_{G \in \mathcal{G}} \Big(\sum_{(M,M') \in D_G} \Psi(M, M') - \sum_{(M,M') \in D_G} P(M, M' | \boldsymbol{\theta}) \Psi(M, M') \Big)$$

Different from traditional applications of MaxEnt-IRL Ziebart *et al.* (2008), the model distance metric in our work depends on both the current and next human model. As a result, $P(M, M'|\Theta)$ above represents the model pair occurrence frequency (MPOF) for a pair (M, M'), which can be computed using dynamic programming. If we denote the probability of occurrence of (M, M') at time t as $\mu_t(M, M')$, we then have $P(M, M'|\Theta) = \sum_t \mu_t(M, M')$. The updating rules for μ_t is as follows:

$$\mu_1(M, M') = P\Big((M_1, M_2) = (M, M')\Big)$$

$$\mu_{t+1}(M, M') = \sum_f \sum_{M''} \mu_t(M'', M) P(f|M) P(M'|M, f)$$

The values for μ_1 are initialized to the probability of the state pair (M, M') being the first pair of a trace. The probability of the occurrence of (M, M') at a certain time step then is calculated based on the occurrence frequency of the previous state pair, which has M as the second entry in the pair, any unit feature change f that the robot would explain to the human while in state M (i.e., according to a stochastic policy), and the probability that the human model would end up in M' when explaining f in state M (i.e., the transition function).

The stochastic policy $P(f|M)$ specifies the probability of explaining f when the human model is M, which is computed as $P(f|M) = \frac{P(M,f)}{P(M)}$. Similarly, they can be calculated using dynamic programming as in Ziebart *et al.* (2008). $\mu_1(M, M')$ can then be approximated using sampled traces generated by the stochastic policy and transition function in each iteration. After learning the parameters for the model distance metric, we utilize uniform cost search for a specific goal to retrieve the best sequence of f_i from a common initial state by maximizing the reward of each state:

$$\zeta_G^* = \arg\max_{\zeta_G} \sum_{(M,M')\in\zeta_G} \Theta^T \Psi(M, M') \tag{4.9}$$

4.3.4 Features Selection

The features used in our learning algorithm for the model distance metric belong in general to two categories: domain dependent and domain independent features.

The domain dependent features in our study are chosen to be the ones that we consider to have an impact on the cognitive load. These features should be fully specifiable by M and M' only given our IRL formulation. Although this imposes a restriction on the set features we can select, it still allows for a rich set of possibilities for any given domain. In our future work, we will further investigate the impact of this restriction on the learned model distance metric.

Domain independent features are chosen to reflect replanning cost. We consider two types of domain independent features: (1) action distance Fox *et al.* (2006), and (2) cost distance. Each of them represents a type of plan distances. The motivation to use plan distances is that, as the information is communicated progressively for an explanation (as unit model changes), humans process it as it is received (i.e., replan based on the current information). Intuitively, the effort involved in the replanning process is correlated to how many changes must be made to a plan, which is often captured by a distance metric. For any model M_i, we denote the plan as π_i. The following distance metrics are considered:

Action Distance: The action distance feature represents the distance between two plans π_i and π_j obtained from states M_i and M_j respectively, as distance$(\pi_i, \pi_j) = \dfrac{\sum_{k=1}^{n} |C(a_{ki}) - C(a_{kj})|}{max(cost(\pi_i), cost(\pi_j))}$. Where $n = |\pi_i \cup \pi_j|$ and $C(a_{ki})$ is the number of occurrences of action a_k in plan π_i, and $cost(\pi_i)$ is the cost of plan π_i.

Cost Distance: Similarly, the cost distance is the difference between the cost of plans π_i and π_j obtained from M_i and M_j respectively: $C(\pi_i, \pi_j) = |cost^*_{M_i}(I, G) - cost^*_{M_j}(I, G)|$.

Levenshtein Distance: The Levenshtein distance Levenshtein (1966) is the minimum editing distance between plans π_i and π_j obtained from M_i and M_j. The larger the minimum editing distance, the more different the plans are. The equation below provides the mathematical definition used to calculate the Levenshtein distance:

$$lav_{\pi_i,\pi_j}(m,n) = \begin{cases} \max(m,n) & \text{if } \min(m,n) = 0, \\ \min \begin{cases} lav_{\pi_i,\pi_j}(m-1,n) + 1 \\ lav_{\pi_i,\pi_j}(m,n-1) + 1 \\ lav_{\pi_i,\pi_j}(m-1,n-1) + \mathbf{1}_{\pi_i(m) \neq \pi_j(n)} \end{cases} & \text{otherwise.} \end{cases}$$

where $lav_{\pi_i,\pi_j}(m,n)$ is the distance between the first m actions in π_i and first n actions in π_j. $\mathbf{1}_{\pi_i(m) \neq \pi_j(n)}$ returns 0 when $\pi_i(m) = \pi_j(n)$ and returns 1 otherwise.

4.4 Evaluation

We evaluated our approach by conducting human-subject studies using Amazon Mechanical Turk (MTurk) in two different domains: scavenger-hunt and escape-room. These domains were designed to expose the subjects to moderately complex situations that required a non-trivial amount of cognitive effort in a short amount of time.

4.4.1 *Scavenger-Hunt*

The task is situated in a damaged office building after an earthquake, portrayed by a floor-plan in Fig. 4.3. Normally, the human uses the doors connecting the rooms to exit the building via the elevator (bottom right corner) from his office (bottom left corner). However, an earthquake may interrupt the human's original path in different ways. The goal of the participant, as a member of the first response team, is to help the trapped human navigate through the building. At any step, the participant can explain one of the changes illustrated in a red box, such as a fire blocking the door or a power outage. The participants are given a map that contains all the possible changes. These are also shown on the right side of Fig. 4.3. There are a total of 10 possible changes that may have changed the plan of the trapped human. However, for any given scenario, only a few selected changes will be present.

Here, the communication bandwidth is limited, and the external agent can only convey one obstacle requirement at a time (e.g., you need a password to deactivate the lock on the second room). The unit feature changes in this domain modeled as 10 different contingencies. Consequently, since the domain dependent features, i.e. the presence of the damage-related issues are binary, we used one hot vector encoding to represent these categorical features as shown in Fig. 4.4. Furthermore, as we encoded these boolean features, it allows our IRL model to learn the actual importance of each categorical feature rather than assuming a natural ordering among them which can be resulted based on the subject's responses. Here, we run the value iteration algorithm instead of the uniform cost search since the initial state and the goal state differ based on the issues present in the scenario.

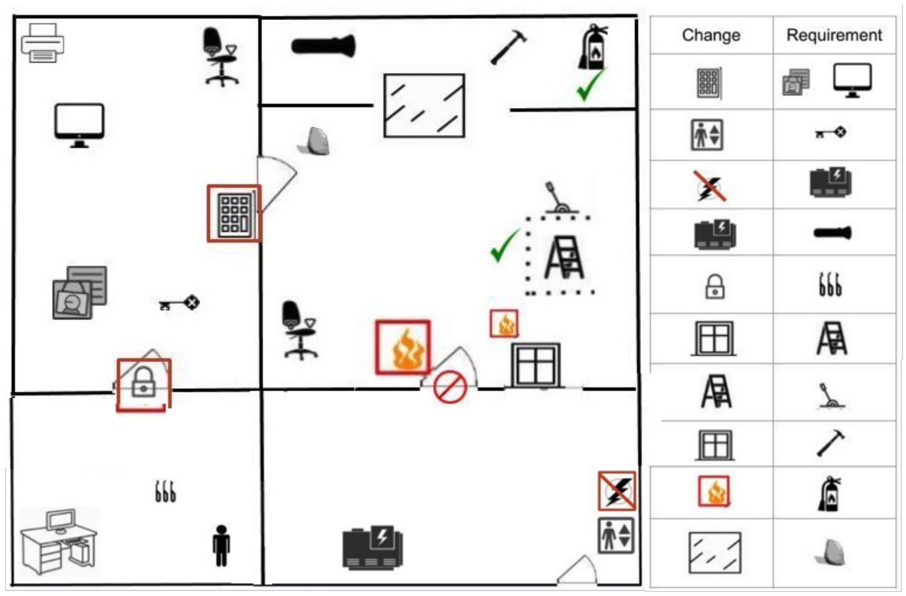


Figure 4.3: Illustration of the scavenger-hunt domain.

Experiment Design

We first explained domain to the participants and emphasized that whether the trapped person understood their explanations determined the life or death of that person. This was meant to encourage the participants to clearly explain the situation in a way to be understood. We then asked the participants to explain the situation to help the trapped human escape the building while playing the role of a member on the first response team. To ensure the quality of the data, We implemented a sanity check question to make sure the participants understood the task. We removed the responses with wrong answers to the sanity questions or if it took them over 3 minutes to finish the task.

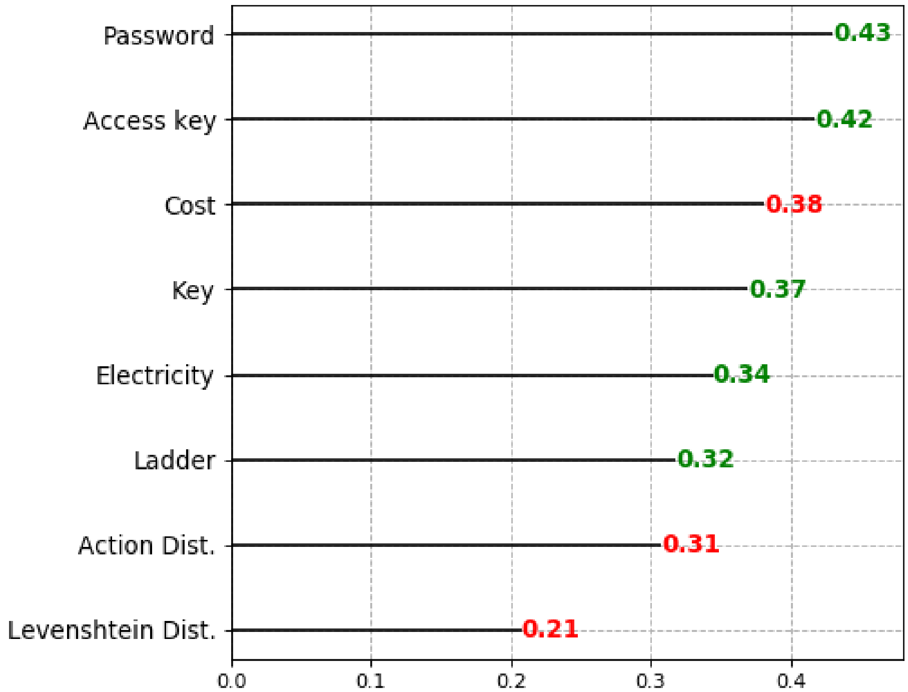


Figure 4.4: Normalized feature weights for the scavenger-hunt domain. The domain dependent features are one hot vector encoding for the state-pairs.

Results & Analysis

We conducted a survey using Qualtrics and recruited 122 human subjects using Mturk, with HIT acceptance rate of 99%. After sifting through the responses as described in the previous section, we used 93 responses, out of which 66 responses were over 5 training scenarios and 27 responses were over 3 testing scenarios. The aim of this evaluation was to analyze if we could learn the human preferences from training scenarios and apply them to testing scenarios (H1). We compared the outputs of our explanation generation algorithm based on the reward function learned by IRL with the subjects' responses for the testing scenarios. The accuracy of our method was 85.2%: our approach successfully matched 23 out of the 27 human responses across 3 testing scenarios. This result showed that humans indeed had certain preferences for information order in such situations and that our method could capture these preferences.

Fig. 4.4 shows the weights learned by IRL for both domain dependent and domain independent features. As Fig. 4.4 suggests, the domain independent features seem to have played an important role. This result inspired us further to verify since the significant weights on domain independent features, which captured plan changes during the explanation process, suggested that the understanding an explanation was a dynamics process. Consequently, we created another domain to verify this result and investigate further. This domain was introduced to impose similar preferences on different domain changes to minimize the influence of semantics (e.g., explaining a fire event could be naturally associated with a higher preference than obtaining a computer passcode).

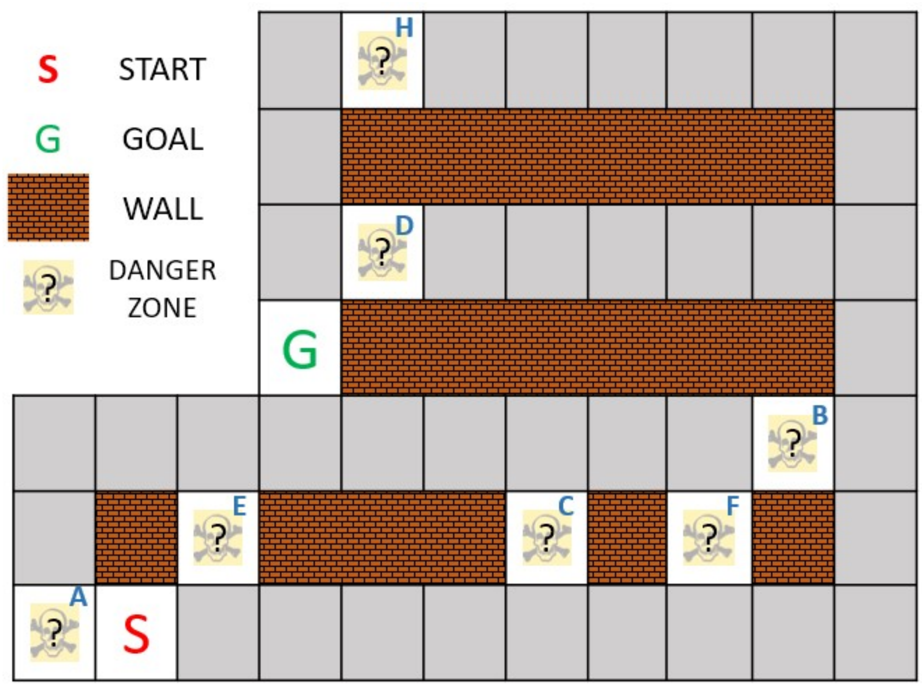


Figure 4.5: Illustration of the escape-room domain.

4.4.2 Escape Room

The task is situated in a damaged nuclear plant represented as a maze-like environment in Fig. 4.5. The goal of the human inside is to navigate from the starting location S to the goal location G as fast as as possible without going through dangerous locations. The set of actions in this domain are going to each of gateway cells from S, and then to G from. For instance, *go to cell* E *from* S, *go to* G *from* E, assuming cell E is not a dangerous passage. Some of the marked locations (see Fig. 4.5) may be affected by the disaster, and become dangerous. Similarly, the participant played an external agent here to inform the internal person about which locations were dangerous. The external agent could only convey one piece of information at a time (e.g., D is a danger zone). The states of the 7 marked locations correspond to 7 contingencies (modeled as unit feature changes in the domain) that may have affected the human's plan.

Experimental Design

We designed 8 different scenarios for the escape-room domain. We used 5 scenarios for training and 3 for testing. Each scenario involves a different set of contingencies and we ensure that there are contingencies in the testing scenarios that did not appear in the training scenarios. During training, the participants are at first introduced to the domain and informed that they are supposed to act as the external agent to communicate the contingencies to the internal person in the scenario. They are explained that the internal person is desperate to escape soon to give them a sense of urgency as well as an incentive to elucidate the situation. We also asked the participants at the beginning about what path the internal person would take assuming no marked locations are dangerous. We use the answer to this question later to sift the data.

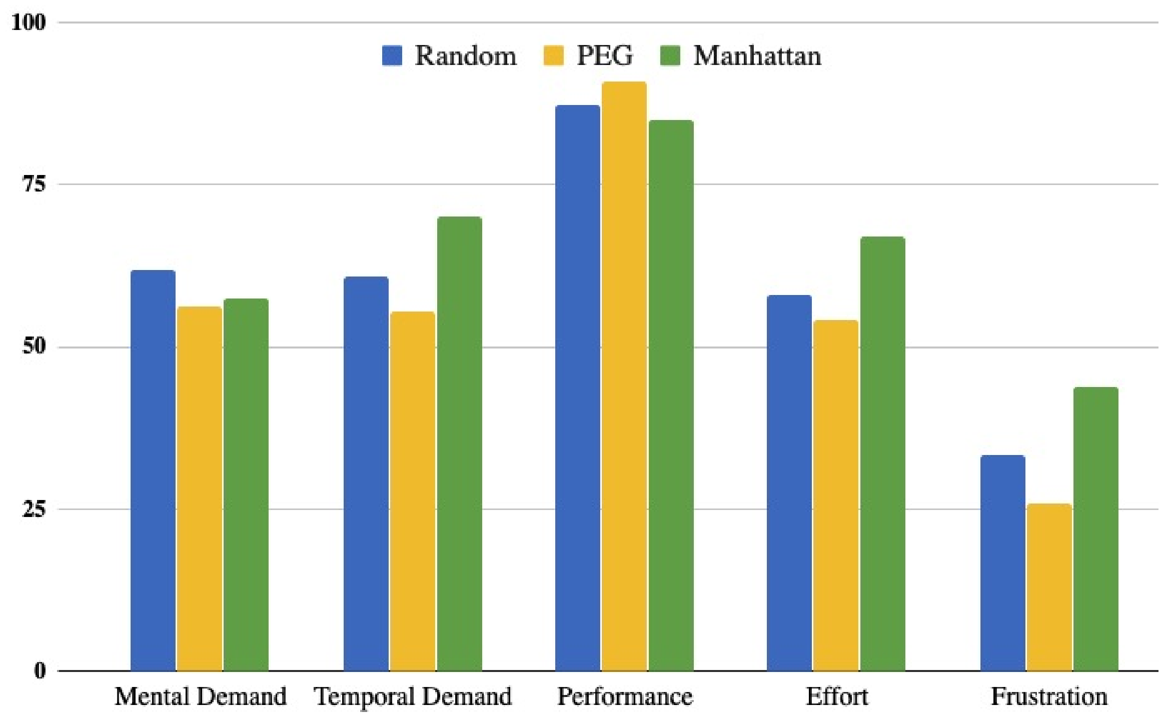


Figure 4.6: NASA TLX results for testing.

In this domain, we further introduced a testing phase for evaluating the effectiveness of PEG. In this phase, new participants played the role of the internal agent. We tested the subject performance with our progressive explanation generation method and two baselines. In particular, we provided the subjects the contingencies that were ordered by 1) our method, 2) a random order, and 3) the Manhattan distance (of the contingency) relative to the starting location S. To create a highly cognitive demanding situation, the subjects were pushed to complete the task within 4 minutes. Responses that failed the sanity check question or ran over 4 minutes were not used. After the task, the subjects were provided the NASA Task Load standard questionnaire (TLX) NASA (2019) to evaluate the efficiency of the different methods.

Results & Analysis

To improve the quality of the responses, we set the criteria that the worker's HIT acceptance rate must be greater than 99% and has been granted MTurk Masters. In the training phase, we created the surveys using Qualtrics and recruited 35 human subjects on MTurk, out of which 21 responses were used. For testing, we have recruited 163 human subjects out of which 87 responses were used. 58 of our subjects were male and 29 were female. The average of age of our subjects was 38.17 with a standard deviation of 11.13. For domain dependent features, we chose 4 features related to relative position of the contingency being explained with respect to the contingencies that have already been explained. We refer to these features as x_{min}, x_{max}, y_{max}, and y_{min}. Table 4.2 shows the normalized weights Θ for each feature after learning via IRL as explained in Sec. 4.3.3. Interestingly, the action distance and Levenshtein distance maintained high weights, which was aligned with our prior results and further validated H2. Simultaneously, however, the weight for the plan cost distance dropped significantly. We attributed this "anomaly" to the simple cost structure in the escape-room domain since the increasing of plan cost does not necessarily increase the cognitive effort there.

Feature Category	Feature Name	Weights
Domain independent	Action Distance	0.44
	Cost Distance	0.04
	Levenshtein Distance	0.46
Domain dependent	x_{min}, y_{min}	0.38, 0.41
	x_{max}, y_{max}	0.35, 0.39

Table 4.2: Normalized feature weights for escape-room domain

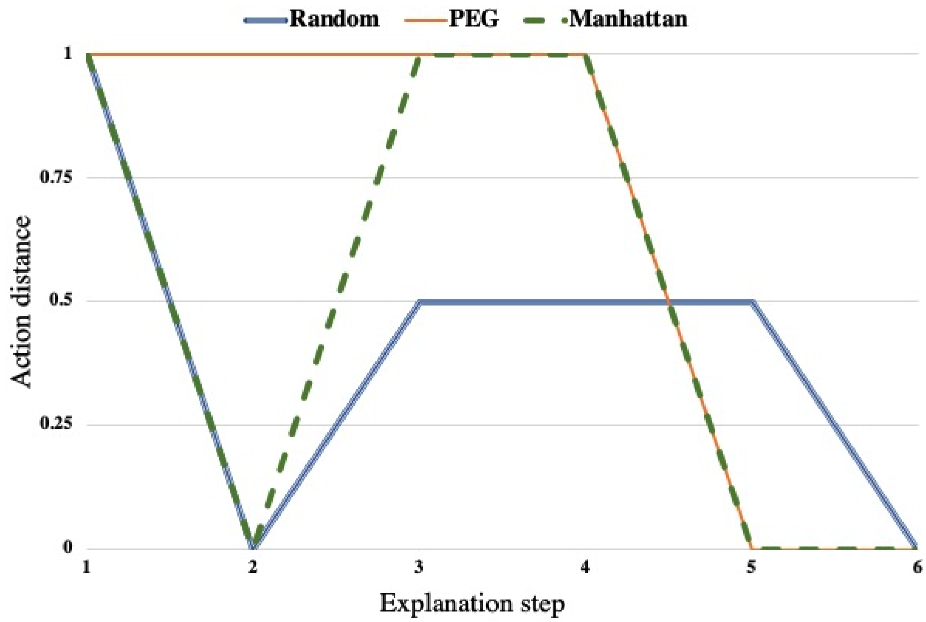


Figure 4.7: Changes of action distance per explanation step for escape-room domain

The subjective results for testing are presented in Fig. 4.6. We can see that our method (PEG) performs better than the baselines for all NASA TLX metrics, a statistically significant difference was observed between PEG and other methods for a weighted sum of TLX metrics, as shown in Table 4.3. Objective metrics further confirmed that our method improved task performance as presented in Table 4.4 which represents the percentage in which the participants came up with the correct plan after the respective explanations. This result verifies H3.

Fig. 4.7 shows the action distance per explanation step for one of the testing scenarios, which is very similar to the Levenshtein distance in this domain. An interesting observation is that the curve of PEG is smoother, i.e., it is missing the oscillation seen in other methods. This intuitively illustrates the progressiveness of explanation enabled by our method, which suggests that the progressiveness of explanations is correlated with the progressiveness of these features.

	Mental Demand	Temporal Demand	Performance	Effort	Frustration	WT TLX (exc Perf)
Random	63.10	61.96	89.06	59.04	33.96	52.47
PEG	**56.19**	**55.37**	**90.74**	**54.11**	**25.89**	**41.93**
Manhattan	57.43	69.93	85.00	66.86	43.93	58.35

Table 4.3: Subjective results for each NASA TLX category

	Random	PEG	Manhattan
Accuracy	85.4 (41/48)	**96.3 (26/27)**	66.7 (8/12)

Table 4.4: Objective performance in terms of task accuracy

4.5 Conclusions

In this chapter, we studied the problem of PEG. We took a step further from the prior work by considering not only the right explanation for the explainee, but also the underlying cognitive effort required to understand the explanation from the explainee's perspective, resulting in a general framework for PEG. To address the challenge with modeling human preferences of the information order, we adopted the formulation of a goal-based MDP and applied IRL to learn the reward function based on traces. Our first contribution is that we show that humans indeed demonstrate preferences for the information order in explanations and we can indeed learn about such preferences using our framework. This verified H1. Upon analyzing the data, we noted strong weights for domain independent features, which suggested that the cognitive process for understanding an explanation is dynamic. This was validated in another domain. Together, results from these two domains validated H2. Finally, we showed that PEG did improve task performance and reduce cognitive load.

Many future direction are possible. One interesting direction is to investigate sub-explanations as more than one unit feature change. This means that the robot may be allowed to explain multiple aspects at the same time. One may anticipate that this would be useful for aspects that are highly correlated. Another possible direction is to generalize the MDP model to remove the Markov assumption, which is quite restrictive for modeling human cognition.

Chapter 5

TOWARDS HUMAN MACHINE SYMBIOSIS: DESIGN FOR EFFECTIVE AI FACILITATION

As Artificial Intelligent (AI) applications increase their influence in technology, intelligent agents advance their presence in our lives. Human-AI teaming facilitates applications where the task is cognitively demanding for human-level cognition or safety is a concern to humans such as military operations, search and rescue scenarios, and space robotics Pérez-D'Arpino *et al.* (2020). As a result, Human-AI planning is becoming a critical capability. Thus, a robotic teammate expects to act and explain the rationale behind its decision making, if necessary, compatible with a human peer Cooke (2015); Lombrozo (2006).

Explainable AI (XAI) Gunning (2017) is increasingly considered an essential paradigm for a new warfighting concept, where it requires new approaches to military planning and fast convergence of capabilities across various domains. Although XAI approaches continuously contribute to the scope and efficiency of planning Human-AI planning problems, many problems remain unaddressed. The critical requirement of explainable agency Langley *et al.* (2017) is to be "*explainable*" to the human partners while developing a deep understanding of machine-computed plans. This ability leads to trust in machine planners and effectively synthesizes insights from multiple human planners and automated planners. However, although AI planners can leverage data at an astounding speed and provide reasoning for their decision-making, they also make sure their human peers perceive the solution. Therefore, AI agents cannot perceive the situation that human experts may recognize using their background knowledge. Moreover, AI planners are deterministic, given the planning domain, limiting their utility in adversarial settings. Further, AI planners are too data demanding while focusing on specific aspects of the planning. Consequently, the human remains an essential element for planning, especially when the problem's constraints are difficult to express for an agent in a dynamic setting.

As AI breakthroughs gain momentum, intelligent machines are increasingly taking on more crucial roles in human-robot teams. The human teammate must understand the decision-making process of the agent. On the other hand, the validity of intelligent machines is constrained by their ability to explain their decision-making processes. As a result, explainable Human-AI planning is an essential paradigm that requires new approaches for fast convergence of capabilities across various domains such as the military.

In this chapter, I explore how explainable agency develops a deep understanding of machine-computed plans, concentrating on the following question: *how to effectively facilitate the decision-making process of a human-AI team?* The answer to this question requires exploring new methods of human-AI collaboration on complex planning tasks. The main factors to be considered for facilitation in this study are:

- The novelty of information shares from one teammate to another
- The required level of engagement
- The activity context, or task complexity
- Known characteristics of the teammate (agent persona)

Humans are known to have limited computational and working memory abilities compared to a machine. Therefore, to maintain collaborative engagement or cognitive readiness for collaborative planning tasks, I introduce a new role in the team: the facilitator.

In previous chapters, different model-based approaches are developed to predict where the human teammate would misunderstand the plan and therefore generate an explanation accordingly. The robot's generated explanation or interactive explicable behavior maintains the human teammate's cognitive workload and increases the overall team situation awareness throughout human-robot interaction. In this chapter, I focus on a rule-based approach to preserve the collaborative engagement of the team by exploring essential aspects of the facilitator agent design. In addition to recognizing wherein the plan might be discrepancies, focusing on the decision-making process provides insight into the reason behind the conflict between the human and the robot. Also, employing a rule-based framework, I shift the focus from assisting an individual (human) teammate to helping the team interactively by focusing on the team decision processes.

The results suggest that the facilitation behavior has an impact on increasing the decision quality (Hypothesis 1). Further, results indicate that the facilitation interventions will increase the planning outcome score, and the decision quality (Hypothesis 2). Finally, according the results, the facilitation interventions will result in higher team collaboration and teamwork (Hypothesis 3).

The facilitation agent will be a resource to the team lead and propose/deliver interventions that will mitigate the team's tendency to rely on previous, successful operation plans. It will ultimately ensure that any good ideas get fair consideration with a more accurate risk analysis. It will support a planning team that has decided to include an AI agent on its planning team, which is being used to propose alternative missions and ensure that those alternatives will also get fair consideration.

Throughout a collaborative planning task, because it is a cognitively demanding task, the facilitator's role is helpful towards goal alignment, planning strategies, and bias mitigation. The facilitator helps the planning team effectively communicate to share [relevant] information to produce an efficient plan.

This chapter takes a step further from previous explanation generation research by developing a planning framework where the Human and Robot cooperate on a collaborative task. I create a facilitation framework to find equilibrium for the human-robot interaction in a performance/practice scenario. The main objectives of this framework are:

- Developing an AI facilitator to provide collaborative, agile decision making in complex planning tasks
- AI facilitated collaborative decision making by diverse, virtual, elastic teams (DVET) of humans Decisions: e.g., plan option generation, analysis, assessment; prioritization
- Human-AI symbiosis in planning tasks using rule-based facilitation intervention

To study facilitation behavior as explanation generation on team cognition, I created a collaborative setting where teammates share their unique knowledge of the task and take responsibility on different planning tasks to fulfill the main objective. The team may consist of a planning agent. The facilitator shares insight based on observed collaborative behavior of the team and influences the decision-making process. The framework created for this study is depicted in Figure 5.1. At different study steps, the facilitator provides priorities, new information to each planner or the whole team. The scenario for this study is elaborated in Section 5.2.

In this framework, the facilitator implicitly affects decision-making by providing insights (facilitation behavior) to increase the plan quality. The facilitation behavior consists of onboarding, mitigating information pooling bias, and debriefing. These notions are defined in Section 5.1.1.

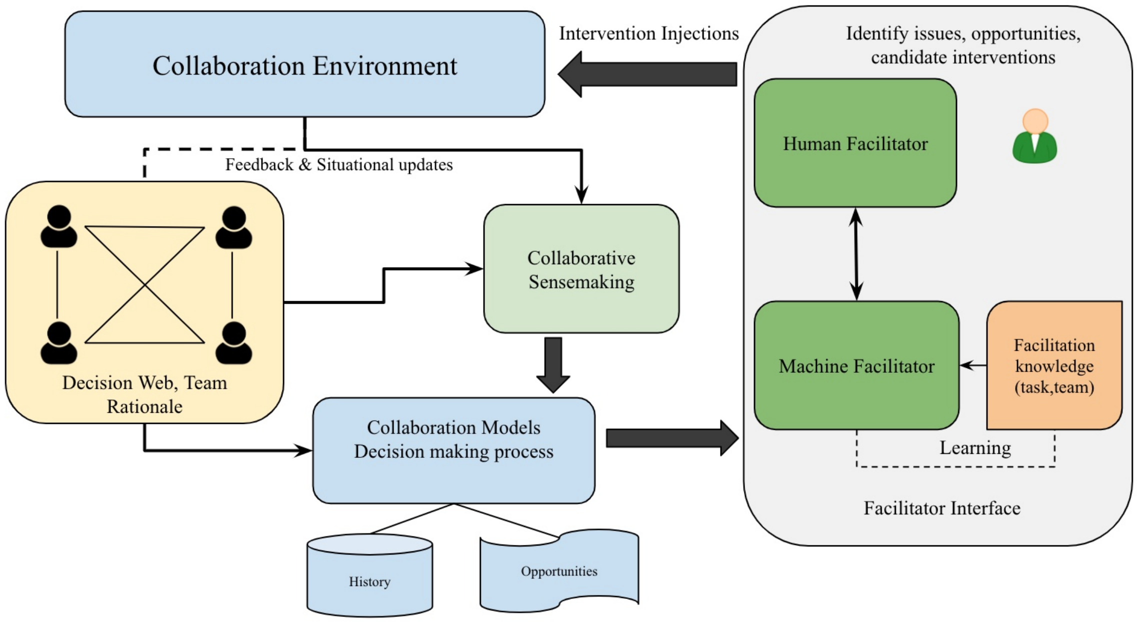


Figure 5.1: The framework for Human-AI teaming utilizing facilitator

5.0.1 Challenges

The following are challenges of intelligent agent architecture designed to account for humans in the loop, categorized based on different aspects:

1. Human-aware planning/decision making challenges

    - Tracking mental models of the humans in the loop using the catalyze/synthesize/harmonize concepts
    - Intention and plan recognition; designing environments for easy goal recognition
    - Joint planning for human-machine collaboration
    - Learning the mental models

2. Evaluation-related challenges

    - Measuring trust between human and AI systems

- Ethical aspects of human-aware AI systems—in particular, the possibility of manipulating humans by leveraging the knowledge of their mental models

3. Facilitation intervention challenges

    - Generating legible/explicable facilitation behavior for every team member
    - Provide interventions in the absence of any shared vocabulary
    - Communication between human and AI systems: natural language; augmented

I address the challenges above by providing facilitation interventions while the human planners are in charge of planning. This approach aims to provide a unique insight into facilitation and team collaboration, which leads to Human-AI symbiosis.

5.1 Towards Human Machine Symbiosis

Our Human-AI symbiosis system leads to robustness, scalability, and efficiency in dynamic missions involving multi-modal resources, constraints, and executives. Without the team performance to suffer in this framework, new human and machine planners can be added or removed from the planning team to address the recently added dimension of objectives. Therefore, incorporating an AI agent as a collaboration planning facilitator is one approach to enable effective Human-AI planning.

The facilitation agent can ask questions, or probes related to the planning task, interface, given information, or teamwork flow throughout the mission. These facilitation actions help planning team members adapt their decision-making to their dynamic planning team's capabilities and constraints.

5.1.1 *Definitions*

I utilize these definitions for our hypotheses and further develop metrics:

1. **Onboarding new team members (Harmonize decision team)**: keep the evolving/elastic team on the same page.

2. **Catalyze decision making**: keep the team moving forward (e.g., collaborative exploration of option space)

3. **Synthesize decision rationale**: continually capture decisions, alternatives, justification, and collective activity to structure collaboration, guide its interpretation and inform downstream decision making (e.g., detailed planning)

Because I am focused on how biases affect teaming cognition, I elaborate on harmonizing the decision team and catalyze the decision making progress:

I elaborate these metrics above by determining the facilitation actions throughout the collaboration task:

1. **Harmonize Decision Team**: Probe and align shared understanding across the team of problem/decision and emerging decision processes.

2. **Catalyze Decision-Making Process**:

    - Stimulate team collaboration on the decision-making process.
    - Capture the team information pooling bias and mitigate it by providing alternative rationale and collaboration activities
    - Abstraction: Integrates feedback from downstream decision making

3. **Synthesize**: Debrief the mission the team had finished and discuss identified teaming issues and opportunities.

5.1.2 Research Questions and Hypotheses

My research, in this chapter, focuses on answering the following question: Will effective use of a rule-based facilitation agent result in (near) optimal plans with a deeper understanding of the decisions by the hybrid planning team? This analysis effort intends to create curves that relate some measure of technical performance to mission effectiveness utilizing the decision quality defined in Section 5.4.1. Although somewhat speculative, this effort provides a new framework and a set of assumptions that could be useful for shaping a more extensive research effort. Accordingly, I aim to test the following hypotheses:

- **Hypothesis 1.** The facilitation interventions will result, in robust complex plans that increase the overall decision quality.
- **Hypothesis 2.** The facilitation interventions will increase the planning outcome score due to the higher-level collaboration of the hybrid planning team. These interventions will be provided online and delivered based on which part of the planning task they decide.
- **Hypothesis 3.** The facilitation interventions will result in superior team collaboration/teamwork.

To test these hypotheses, each team will randomly attend one of these three collaboration conditions described as below:

1. No Facilitator: Participants collaborate to plan for a logistic relief task.
2. Agent Facilitator: Participants collaborate to plan + an agent facilitates their decision-making process. The agent is Wizard of Oz and has rule-based scripts for interventions.

3. Human Facilitator: Participants collaborate to plan + a human facilitates their decision-making process. The human facilitator has rule-based scripts + uses informal language to facilitate the process.

5.1.3 Bias

To better understand the effect of facilitation behavior on collaborative planning and team cognition I induced three biases in different conditions:

- Information pooling bias: The tendency to discuss/share common information and omit unique information Stasser and Titus (1985); Rajivan *et al.* (2013); Rajivan and Cooke (2018). Each team member has some information shared by all team members and some that are unique to them. The bias is one in which people tend to discuss standard details and fail to share unique.

- Anchoring bias: "The tendency to rely too heavily or overly restrict one's attention to one trait or piece of information when making judgments. The information in question can be relevant or irrelevant to the target decision and numerical or non-numerical. Includes focalism or the focusing illusion" Tversky and Kahneman (1974). The group sees salient information and anchors or gets stuck on it. The first piece of information presented at the very beginning while the rest of it will be provided with delay.

- Groupthink (Bandwagon) Bias: Groupthink is the team version of the above bias. This bias plays on our need to "fit in" and conform to social norms. An individual's perspective can be steered toward a group's plan, even if that individual disagrees with the group's beliefs. The fear of being the odd-one-out and ridiculed coupled with the idea that others are smarter than us leads us to doubt ourselves and favor the collective response. This is the bias that prevents an individual from speaking up and expressing an alternative view Kahneman *et al.* (1982); Tsintsadze-Maass and Maass (2014).

5.2 Experimental Setup

I evaluate the introduced approach by conducting a human-subject study using a web based framework in a disaster response domain. This domain is designed to expose the subjects to complex teaming collaboration situations that required a non-trivial amount of cognitive effort in a short amount of time.

To adjust the study design, I have benefited from pilot studies analysis. I analyzed part of the data as it is being collected to fine-tune the experimental design and ensure that sufficient data is collected. Additionally, I explored the pilot data further to inform the experimental design and my operational analysis. The pilot study results are only used for design purposes and are excluded from the study results.

5.2.1 The Disaster Response Domain

To test the significance of the facilitator and its effect on the team, I have designed a collaborative disaster relief planning task. I designed the domain as a collaboration task on an island, where a disaster happened, and the participants collaborated to solve the problem.

In this task, four warehouse managers have to collaboratively devise a plan to respond to a disaster that happened on a fictional island. There is an urge to communicate and converge capabilities across various domains fastly.

To address the situation, the participants have to work within a team and move supply materials from one safe location to an affected area to assist the population recover from a natural disaster. Their task is to provide five decisions based on the planning interface and the brief mission document provided to them:

- Deciding on the destination priorities
- Assigning the destinations (disaster relief focuses) to warehouse managers
- Assigning truck
- Choosing the best route to get to the goals
- Distributing supplies between warehouses

To simulate an actual situation, there is different information provided to each warehouse manager, categorized as common/uncommon to other team members and relevant/irrelevant to the task. Moreover, to better understand human-AI teaming, one warehouse manager is operated by a (wizard of Oz) planning agent. The planning agent script is written specifically to provide it with a persona, induce biases, and represent a less cooperative agent. This automated warehouse communicates with others using the Text to Speech software and has a robotic voice. The agent is:

- Not transparent
- Reactive
- Does not share information until asked
- Does not initiate collaboration

The agent responds to the other teammates' questions or the facilitator's interventions but does not volunteer to share the information. As a result, throughout the collaboration, the team could easily exclude the agent, and therefore, the agent would withhold the critical information needed to plan the tasks accurately.

5.3 Evaluation

As mentioned in Section 5.1.2, three different conditions for the designated experiment have been conducted. For all three conditions, I have the same initial state for all the warehouse managers and the requirement of the damaged areas. In each condition, I conducted two missions: in the first mission, the participants have enough resources to fulfill their task. However, in the second mission, there are not sufficient resources to accomplish the collaborative planning task.

The facilitator has access to the exact requirements of each disaster focus and to each participant's current supply and the state of the main decisions at all times. Figure 5.4 shows the facilitator view.

The participants are recruited from a North American university. The participants are put randomly in a team for the study. Each unit consists of three participants, the automated warehouse manager, called the planning agent, and on conditions two & three, a facilitator. Each team went through a training mission to learn the interface basics and get familiar with the study. Then, the team will start to collaborate on two tasks. First, the experimenter introduce the participants to the domain by online training (`http://charttask.com/training1.html`). During the training, they learn about the details of the interface they are about to use, the map, and how to utilize the information that the facilitator might provide. Figure 5.2 shows the map.

Each human subject assigned a role on the task. As Figure 5.3 illustrates, the participant's main task is shown using the color green. Therefore, each player must collaborate with other teammates to understand the best assignment for the designated task. As explained briefly in Section 5.2, each subject has limited information about the resources available to herself and a range of requirements needed for each

(a) Route 1

(b) Route 2

(c) Route 3

(d) Route 4

Figure 5.2: The map of the island where the natural disaster happened. The players have access to the map and its relevant information at all times during the planning task. The route information is shown as presented in the interface for the first mission in figures 5.2a, 5.2b, 5.2c, 5.2d.

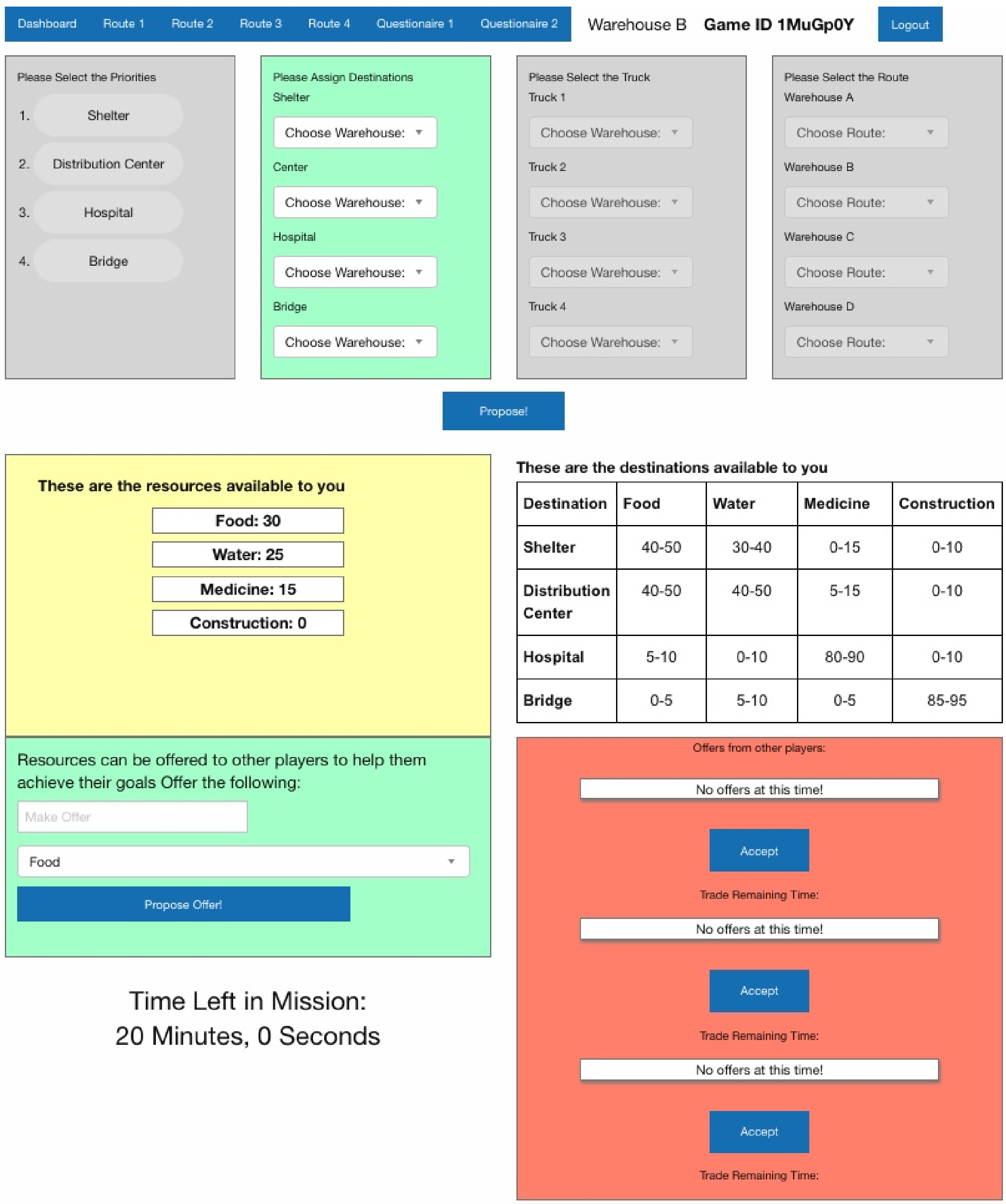


Figure 5.3: Player's dashboard view for the disaster response domain

disaster focuses.

The facilitator starts each mission, and the participants have 25 minutes to collaborate and fulfill each task. During each mission, the facilitator provides three categories of intervention based on the definitions in Section 5.1.1.

- Harmonize: Onboarding new agent planner; introducing new member and its capabilities; The facilitator, at the beginning of the first mission, mentions that the warehouse manager Alpha is being called on an emergency and being replaced by her assistant. The assistant is an AI planning agent familiar with the planning task and can communicate with the participants.
- Catalyze: The AI agent is withholding the unique information about the task, creating an information pooling bias situation because this information is required to make a good plan. The facilitator then prompts the team or an individual to share what they perceive as relevant information to other team members. The planning agent starts to share information at this point.
- Synthesize: At the end of the mission, the facilitator debriefs the planning team and provides the score and summary of their collaboration. The facilitator asks the team to answer on: (1) What went well during the first mission. (2) What didn't go well? (3) Discuss what needs to be improved.

5.4 Measures

To measure the effectiveness of the facilitation behavior over the collaborative planning task, I have defined three sets of measures: decision quality and planning outcome score, which is categorized as objective, and collaboration score, which is classified as subjective. The measurement definition is provided in the following subsections.

Figure 5.4: Facilitator's dashboard view for the disaster response domain. Compare the table of the resources in the middle to the Player's view of Figure 5.3.

5.4.1 *Decision Quality*

The score reflects how I judge the relevance of a plan by considering a different aspect of the plan. Here, I consider two approaches for plan evaluation: utility based, and planning outcome based. For utility-based, I assign a score to the plan based on a set of features and predict how likely this plan will succeed in execution on a scale (range). The feature is comprised of three categories: (1) Plan efficacy: The competence of the plan based on the selection of priorities, destinations, trucks, and routes (2) Plan cost: Number of resource moves (3) Plan bonuses: The distance of the created plan with the optimal plan. This distance is calculated using the constraint set from the problem. This is a measurement of how much they would be penalized if they ignored the constraints. A plan is considered better if and only if it has a higher score. The designed experiment provides insight into how teams of humans and machines can be more effective as a planning team with a Collaboration Facilitation Agent (CFA).

The score is color-coded and presented in Figure 5.5. I use three different categories for the score for the debrief session: below average (< 50) shown in red, moderate ($< 50 < 70$) shown in yellow, and high (< 70) shown in green. The detailed score is only presented to the facilitator. The facilitator decides how to use the score information in the debrief session as explained in Session 5.2.

5.4.2 *Planning Outcome Score*

For planning outcome, I focus on how well the plan would carry out practically based on the objectives and constraints set for each scenario. Here, I answer whether the plan devised by the team would fulfill the defined problem if it would be put to action. More specifically, I make sure that the created plan will accomplish the

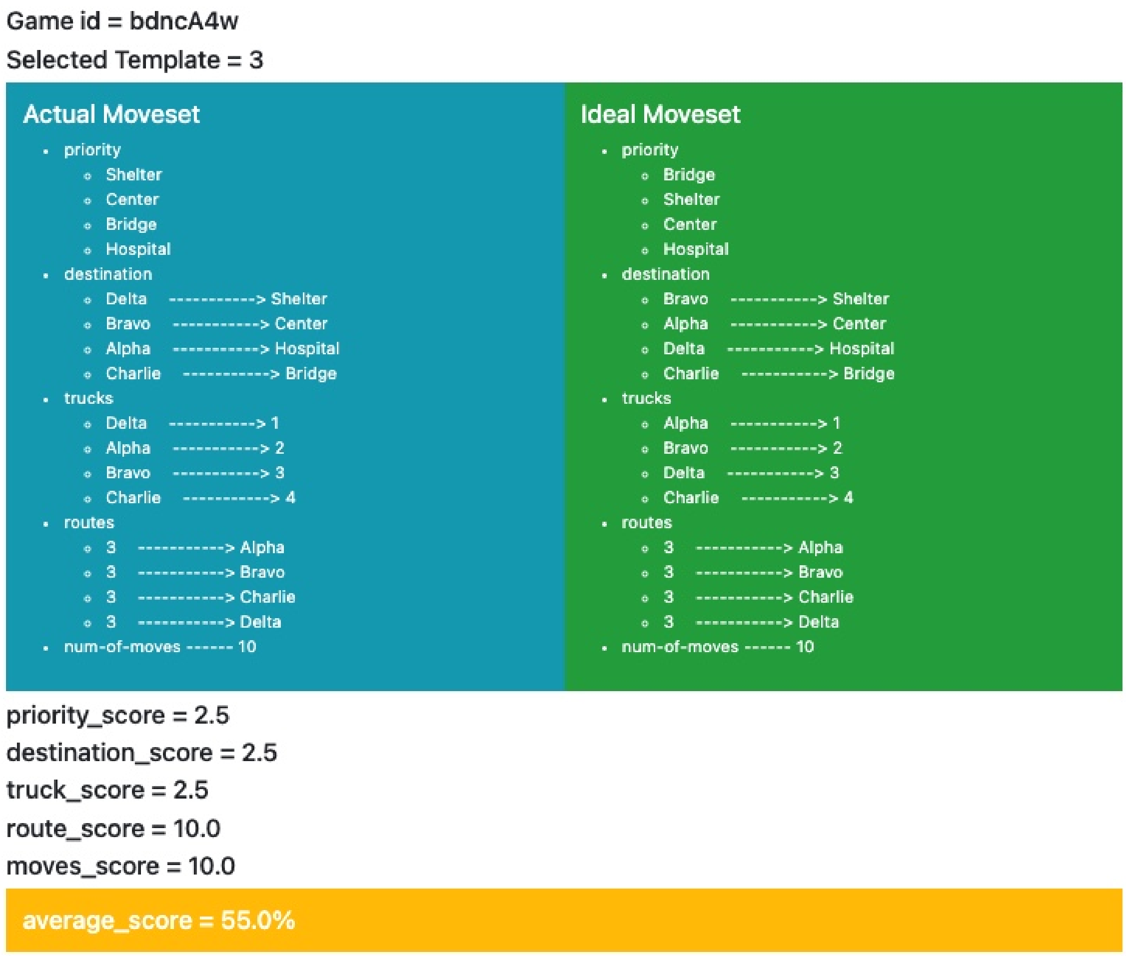


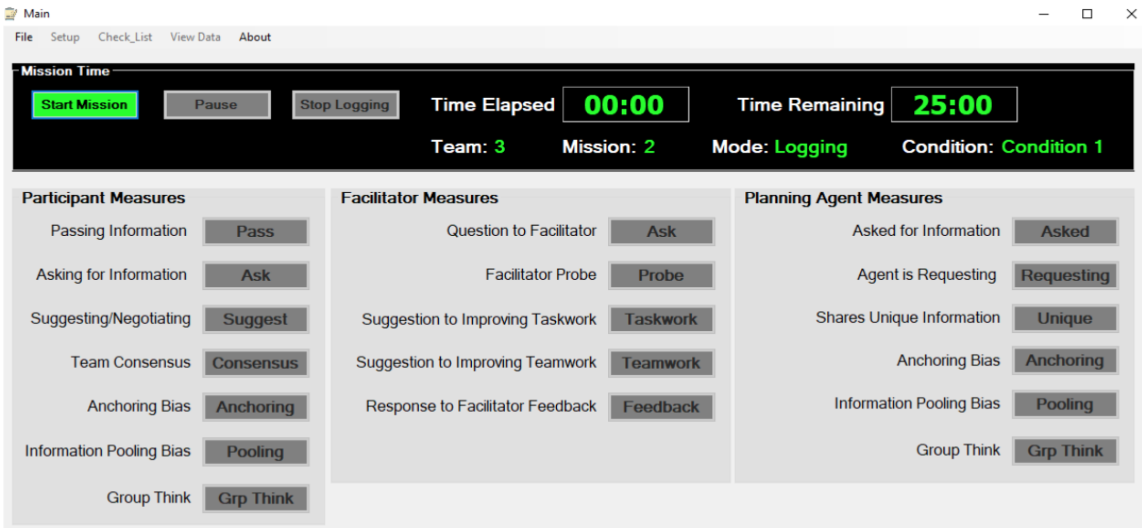
Figure 5.5: The color-coded decision quality was calculated based on the sub-tasks defined in the experiment out of 100. For each category, the score is calculated out of 10 using a similarity comparison function between the ideal decisions and actual decisions.

goal set at the beginning for the team if unfolded and executed. Finally, an objective follow-up metric is to check if the created plan will address the priorities in the same order that the team set as the first sub-task.

Figure 5.7a and Figure 5.7b illustrate the calculation of the planning outcome score. The top table (Figure 5.7a) illustrates the type of a decision, what is decided, the number of supply moves, as well as the truck, destination, and route selection with their score. Figure 5.7a first shows the initial situation for Alpha, Bravo, Charlie, and Delta and the requirement for each of the destinations to be fulfilled. Each row shows the source and destination warehouse manager in a supply distribution move and the offered amount. The red value shows the moved resource being *deducted* from the source warehouse, and the green value shows the *added* to the destination warehouse. Finally, the destination location main column shows the assigned destination that the warehouse that received the offer is responsible for. A negative number means the number of resources needed to fulfill that task under the illustrated category. Similarly, the positive number represents the surplus of the displayed type for that warehouse. For instance, the 11th row in this table shows Alpha; Bravo; food; 5; 20 (in red); 25 (in green); Center; -15; 0; -15; 15. This means Alpha offered five food to Bravo. The remaining food for Alpha is 20, and the updated food for Bravo is 25. Bravo is responsible for the distribution center and needs 15 more units of food, and 15 more medicine units, while it does not need water and has 15 surpluses of construction supplies.

5.4.3 Team Process Score (TPS)

To calculate the subjective score of team process score, I have created software shown in Figure 5.6 to label the team interactions. The experimenter labels the interaction between teammates and between the facilitator and the teammates, as



Participant Measures Facilitator Measures Planning Agent Measures

Figure 5.6: Team Process Score. For simplicity, the software has three sections, categorized for participant, Facilitator, and planning agent (automated warehouse). See Table 5.1, Table 5.2, and Table 5.3 for the definition of each of the measures.

well as the team decision making, whether they planned based on a consensus or not, how effectively they shared information, etc.

The definition for each of the measures in the TPS software is provided in Table 5.1, Table 5.2, and Table 5.3.

5.5 Results

To measure the facilitator's impact, I used the metrics tied to the plan's quality and practicality of the generated plan.

Table 5.4 shows overall decision quality for different conditions, which illustrates the facilitator's effectiveness for decision quality. I have utilized the score calculations discussed in Section 5.4 to evaluate the effect of the facilitator across three different conditions mentioned in Section 5.3. Furthermore, Table 5.5 provides the decision

Participants	Definition
Passing info.	The participant shares information regarding the mission to the team
Asking Info.	Participant requests info.
Suggesting/negotiating	Proposes a course of action for a priority, also the inverse, a counteroffer.
Team Consensus	Group agrees on a course of action (order of priority task, prioritizing who trades 1st) when making a small decision (hospital should be first) or large one (order of all priorities). Only select once per decision, not for every person. After 1 agreement silence is compliance.
Anchoring Bias	If participants base their decision for their destination on highest supplies (or otherwise focus on supplies).
Information Pooling Bias	When participant neglects to share unique information.
Groupthink	When a participant gives a suggestion (that is incorrect) and participants agree with it with no further discussion or deliberation.

Table 5.1: Participant Labeling definitions for TPS

Decision	Action	# moves (M)	Priority (P)	Destination (D) SCHB	Trucks (T)	Route (R)	P Score	D Score	T Score	R Score	M Score	Average Score %
-1	initialize	-1	0000	0000	0000	[0, 0, 0, 0]	0	0	0	0	0	0
Priority	H-C-S-B	0	HCSB	0000	0000	[0, 0, 0, 0]	0	0	0	0	0.0	0.0
Priority	S-C-H-B	0	SCHB	0000	0000	[0, 0, 0, 0]	0	0	0	0	0.0	0.0
Priority	S-C-B-H	0	SCBH	0000	0000	[0, 0, 0, 0]	2.5	0	0	0	0.0	5.0
Priority	S-C-B-H	0	SCBH	0000	0000	[0, 0, 0, 0]	2.5	0	0	0	0.0	5.0
Dest (Sh-Ce-Ho-Br)	D-B-A-C	0	SCBH	DBAC	0000	[0, 0, 0, 0]	2.5	2.5	0	0	0.0	10.0
Dest (Sh-Ce-Ho-Br)	D-B-A-C	0	SCBH	DBAC	0000	[0, 0, 0, 0]	2.5	2.5	0	0	0.0	10.0
Truck	D-A-B-C	0	SCBH	DBAC	DABC	[0, 0, 0, 0]	2.5	2.5	2.5	0	0.0	15.0
Route	3-3-3-3	0	SCBH	DBAC	DABC	[3, 3, 3, 3]	2.5	2.5	2.5	10.0	0.0	35.0
Route	3-3-3-3	0	SCBH	DBAC	DABC	[3, 3, 3, 3]	2.5	2.5	2.5	10.0	0.0	35.0
Route	3-3-3-3	0	SCBH	DBAC	DABC	[3, 3, 3, 3]	2.5	2.5	2.5	10.0	0.0	35.0
Route	3-3-3-3	0	SCBH	DBAC	DABC	[3, 3, 3, 3]	2.5	2.5	2.5	10.0	0.0	35.0
Route	3-3-3-3	0	SCBH	DBAC	DABC	[3, 3, 3, 3]	2.5	2.5	2.5	10.0	0.0	35.0
Final trades	Final trades	10	SCBH	DBAC	DABC	[3, 3, 3, 3]	2.5	2.5	2.5	10.0	10.0	55.0

(a)

Trade				Source Player				Destination Player				Destination Location				
Src-Player	Dst-Player	Off. Resource	Off. Amount	Food	Water	Medicine	Supply	Food	Water	Medicine	Supply	Name	Food	Water	Medicine	Supply
Alpha				25	25	30	10									
Bravo				20	35	15	20									
Charlie				15	10	20	40									
Delta				25	40	35	25									
	Shelter							20	40	5	0					
	Center							40	35	30	5					
	Hospital							20	30	70	0					
	Bridge							10	10	0	95					

Alpha	Bravo	food	5	20				25				Center	-15	0	-15	15
Charlie	Bravo	food	15	0				40				Center	0	0	-15	15
Charlie	Bravo	water	10		0				45			Center	0	10	-15	15
Delta	Alpha	water	5		35				30			Hospital	0	0	-40	10
Delta	Bravo	medicine	30			5				45		Center	0	10	15	15
Bravo	Alpha	medicine	15			30				45		Hospital	0	0	-25	10
Charlie	Alpha	medicine	20			0				65		Hospital	0	0	-5	10
Bravo	Charlie	supply	20				0				60	Bridge	-10	-10	0	-35
Delta	Charlie	supply	25				0				85	Bridge	-10	-10	0	-10
Alpha	Charlie	supply	10				0				95	Bridge	-10	-10	0	0

Delta--->	Shelter			25	35	5	0	20	40	5	0					
Bravo--->	Center			40	45	30	0	40	35	30	5					
Alpha--->	Hospital			20	30	65	0	20	30	70	0					
Charlie--->	Bridge			0	0	0	95	10	10	0	95					

(b) The planning outcome of the supply distributions with respect to the decided priorities.

Facilitator	Definition
Question Facilitator	The facilitator is asked a question directly by one of the participants.
Facilitator Probe (Question)	An intervention were the facilitator request the participants to think deeper about there decision making to see if there is a better solution available
Suggestion to improving task work	Intervention that focuses on the task the is currently being completed (route selection, trading resources).
Suggestion to improving teamwork	Intervention that focuses on how the team collaborates
Response to facilitator feedback	Anytime there is a direct response to facilitator. Mostly be used after the mission.

Table 5.2: Facilitator Labeling definitions for TPS

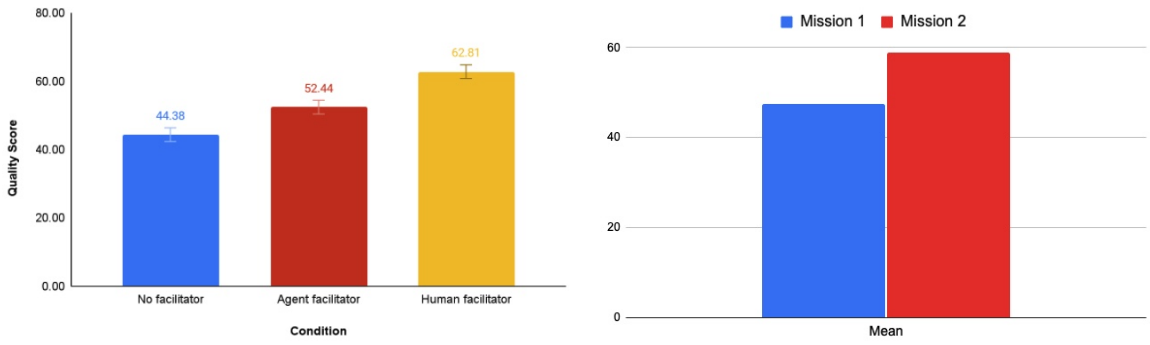
quality results for the mission 1 and mission 2 across different conditions.

I used a 3 (condition) × 2 (mission) split-plot analysis of variance (ANOVA) to analyze the teams' decision quality. There were significant condition, $F(2, 30) = 21.22$, $p < 0.001$, and mission main effects, $F(1, 30) = 19.95$, $p < 0.001$. However, the condition by mission interaction effect was not significant, $F(2, 30) = 1.17$, $p = 0.328$. Based on the significant condition main effect, pairwise comparisons (Least Significant Difference-LSD) indicate that teams in the human facilitator condition had a higher decision-quality than those in both the agent facilitator and no facilitator conditions ($p < 0.001$). The agent facilitator condition was also higher than the no facilitator condition ($p < 0.001$; see Figure 5.8). This finding supports the hypothesis (H1) that the facilitator affects the quality of communication, leading to the teams better

Agent	Definition
Anchoring Bias	If Alpha is able to induce bias. Alpha bases destination decision on highest supplies available.
Information Pooling Bias	When Alpha does not share unique information. This will be selected if Alpha is not asked to share information.
Groupthink	When Alpha introduces a suggestion and participants agree without exploring other alternatives.
Agent Asked	When participants ask Alpha for information or to confirm a decision.
Agent Requesting	When Alpha states its needs for supply redistribution.
Agent Supplies	When Alpha shares unique information (according to the script).

Table 5.3: Agent Labeling definitions for TPS

identifying the relevant and unique information provided to them. Significant mission main effects show that the decision quality significantly increased from Mission 1 to Mission 2 ($p < 0.001$). This shows that the participants are required to communicate even more effectively in mission 2 compared to mission one to recognize and fulfill the top priorities because there are not enough resources to fulfill the second mission. Also, the increased score is partially due to the learning effect that results from gaining more experience by spending more time with the environment. Furthermore, while the learning effect exists similarly for all of the conditions when going from Mission 1 to Mission 2, the lack of a mission $\times$ condition interaction suggests that learning occurs equally for all conditions, but that decision quality is further impacted by the presence of a facilitator.



(a) (b)

Figure 5.8: Decision quality score (a) across the conditions and (b) across the missions (Error Bars indicate that Standard Error (SE)of the mean).

I used another 3 (condition) × 2 (mission) split-plot analysis of variance (ANOVA) to analyze the teams' outcome score. There were significant condition, $F(2, 30) = 31.51$, $p < 0.001$, and mission main effects, $F(1, 30) = 20.24$, $p < 0.001$. However, the condition by mission interaction effect was not significant, $F(2, 30) = 0.247$, $p = 0.782$. Based on the significant condition main effect, pairwise comparisons (Least Significant Difference-LSD) indicate that teams in the human facilitator condition had a significantly higher outcome score than those in both the agent facilitator and no facilitator conditions ($p < 0.001$). The agent facilitator outcome score was also significantly higher than the no facilitator condition ($p < 0.001$). This finding supports the hypothesis (H2) that the facilitator interventions affect the outcome score. This represents that the facilitator positively affects the interaction quality of the team which leads to the teams better identifying the relevant and unique information provided to them. Significant mission main effects show that the outcome score significantly increased from Mission 1 to Mission 2 ($p < 0.001$). This shows that the participants are better perceiving the agent's persona as well as mitigating the induced biases better. However, while the learning effect exists similarly for all of the conditions when going from Mission 1 to Mission 2, the significant increase in the outcome score in the facilitator conditions, compared to no facilitator, suggests that the learning does play a critical role for all, but facilitator condition matters on top of that.

The experimental design allows us to speculate on how the CFA technical capability will impact joint forces' ability to implement the concept of Multi-Domain Operations effectively. As part of this effort, I include an analysis effort focused on extrapolating the experimental results and insights into a framework for discussing Measures of Effectiveness for multi-domain operations. Figure 5.10 and Figure 5.11 show the sub-tasks decision quality across the three conditions. The results illustrate that the human facilitator performs significantly better from the first mission to the second mission that may be attributed to the human facilitator. Most importantly, the AI facilitator does not seem as effective as the human facilitator. This observation is also verified with the Table 5.5. Particularly, the human facilitator performs better in helping the participants route selection sub-task. Another interesting aspect is that the priority score across all conditions declined from the first mission to the second. This is due to the fact that in the second mission there are not enough resources to fulfill the task which puts a lot of calculation overload on the team. The destination assignment improves from mission one to mission two, as the team learns to communicate better. The truck selection shows a significant improvement from the first mission to the second mission for facilitator conditions compared to no facilitator condition. These results with the Table 5.4 results suggest that the facilitator helps mitigating the biases better.

Conditions	Overall (Average) Decision Quality
No Facilitator	44.38
Agent Facilitator	52.44
Human Facilitator	62.81

Table 5.4: Overall decision quality across conditions

	Mission	Average Decision Quality
No Facilitator	1	41.67
No Facilitator	2	47.22
Agent Facilitator	1	46.25
Agent Facilitator	2	58.63
Human Facilitator	1	55
Human Facilitator	2	70.63

Table 5.5: Overall decision quality across missions and conditions

5.5.1 *When Does Facilitation Matter?*

Following the discussion provided, one crucial question that needs an answer is to explore what situation needs facilitation and which type of facilitation (*agent or human*) would be appropriate.

The first part of the answer is to analyze further the planning outcome score shown in the Table 5.6. The results suggest that cognitive demand may be the same across conditions, but that the presence of a facilitator mitigates the demand. These results verify H2. Table 5.7 provides average and standard deviation for outcome

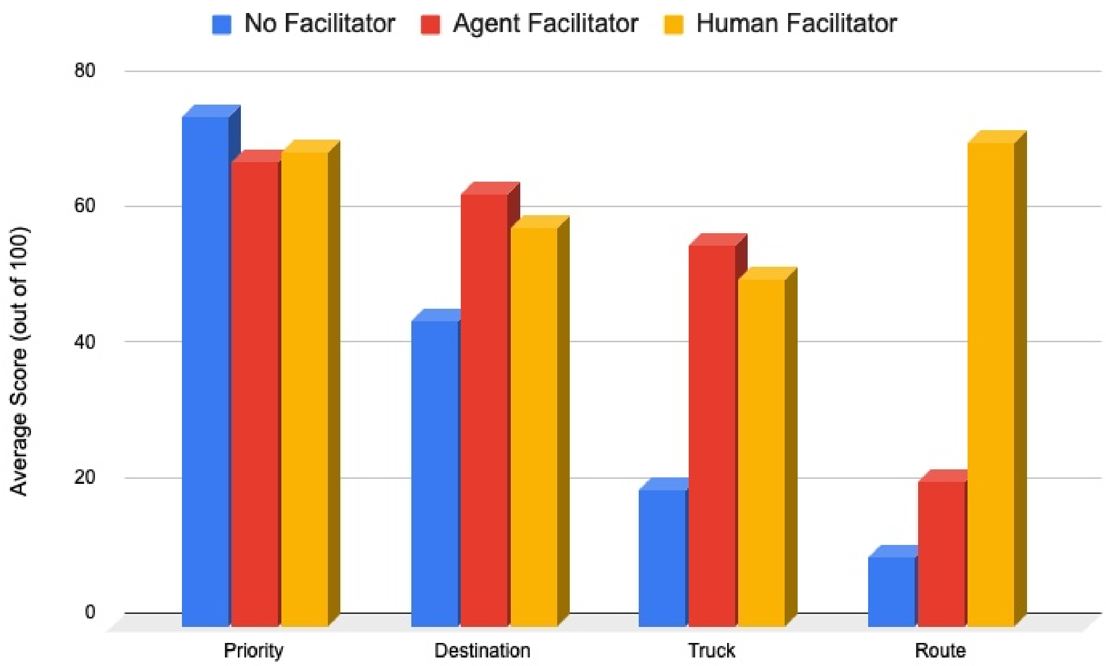


Figure 5.9: The decision quality for different sub-tasks across different conditions

	No Facilitator	Agent Facilitator	Human Facilitator
Planning outcome Score (out of 1000)	789.25	867.95	912.78

Table 5.6: The planning outcome score across the three conditions

score across different conditions and missions.

To analyze the TPS, I classified it into three different categories: (1) Team process communication score: pass information, ask for information, suggest negotiate, agent asked, agent requests, agent supplies unique information; (2) Facilitator measures; (3) Team process bias score: anchoring bias, information pooling bias, groupthink bias (see Table 5.1, Table 5.2, and Table 5.3). For each of these categories, I summed the average number of occurrences of each factor across each mission and condition.

I used another 3 (condition) × 2 (mission) split-plot analysis of variance (ANOVA) to analyze the team process communication score. There was a significant condition

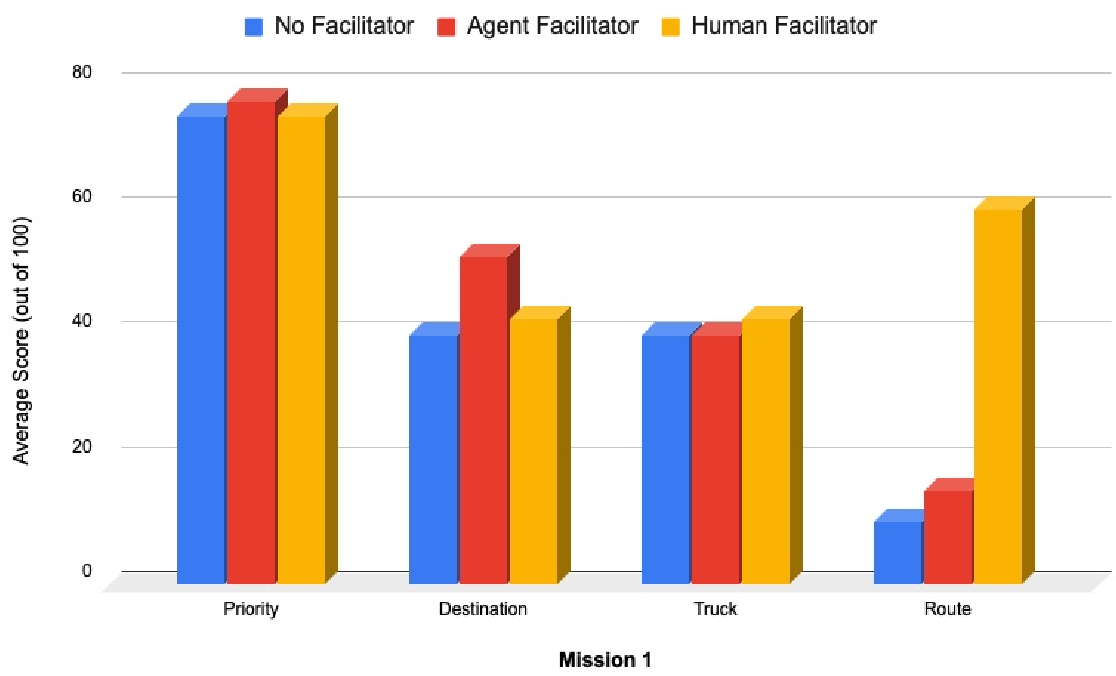


Figure 5.10: The decision quality for different sub-tasks in mission 1 across different conditions

	Mission 1	Mission 2
No Facilitator	245.5 (12.12)	176 (50.71)
Agent Facilitator	155.6 (68.89)	108.5 (54.83)
Human Facilitator	108.89 (39.07)	65.56 (68.63)

Table 5.7: The planning outcome average and standard deviation in parenthesis across the three conditions and two missions

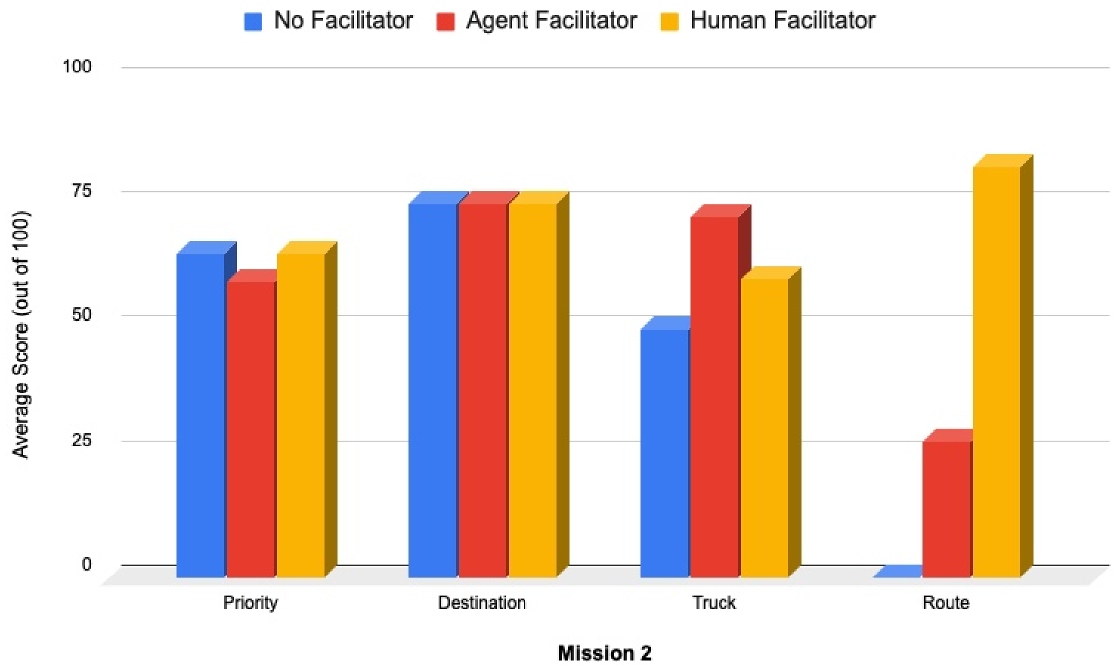


Figure 5.11: The decision quality for different sub-tasks in mission 2 across different conditions

effect, $F(2,30) = 11.56$, $p < 0.001$, and the condition by mission interaction effect $F(2,30) = 3.64$, $p < 0.05$. However, there was no significant mission main effect, $F(1,30) = 0.91$, $p = 0.347$. Based on the significant condition main effect, pairwise comparisons (Least Significant Difference-LSD) indicate that team process communication score was significantly higher in the human facilitator condition than the agent facilitator condition, and it was significantly higher in the agent facilitator condition than no facilitator condition ($p < 0.001$; See Figure 5.12). This finding shows that the facilitator affects the team communication positively. As a result, the team asked the agent for more unique information, asked each other more for information, and negotiated more on the best strategies to approach the planning task in the facilitator conditions.

I used another 3 (condition) $\times$ 2 (mission) split-plot analysis of variance (ANOVA)

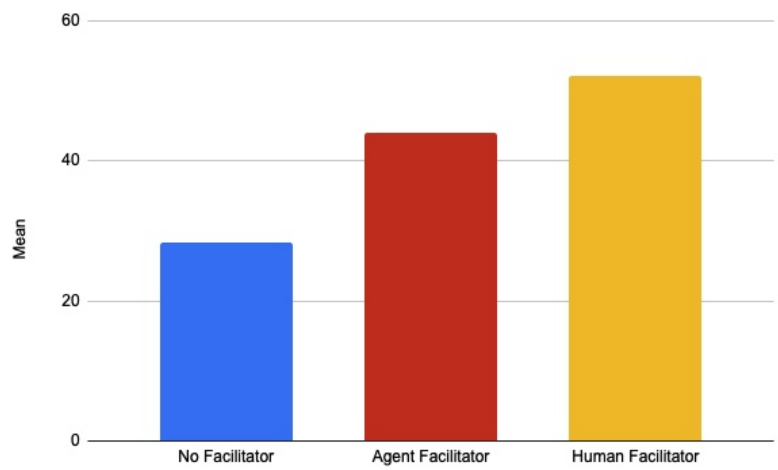


Figure 5.12: Team process communication score across the three conditions

to analyze the TPS facilitator measure. There were significant mission, $F(1, 30) = 26.20$, $p < 0.001$ and condition main effects $F(2, 30) = 37, 80$, $p < 0.001$, and condition by mission interaction effect $F(2, 30) = 7.03$, $p < 0.05$. Based on the significant mission main effect, pairwise comparisons (Least Significant Difference-LSD) indicate that facilitator interaction score was significantly decreased from Mission 1 to Mission 2 ($p < 0.001$). Based on the significant condition main effect, only no facilitator condition significantly differ than the other two conditions, but the other two conditions were not significantly differ one from the other ($p = 0.108$; See Figure 5.13).

I used another 3 (condition) $\times$ 2 (mission) split-plot analysis of variance (ANOVA) to analyze the team process bias score. There was a significant condition effect, $F(2, 30) = 6.87$, $p < 0.05$. However, there were no significant mission main effect, $F(1, 30) = 0.51$, $p = 0.482$, and no significant condition by mission interaction effect $F(2, 30) = 2.07$, $p = 0.144$. Based on the significant condition main effect, pairwise comparisons (Least Significant Difference-LSD) indicate that team process bias score was significantly lower in human facilitator condition than the agent facilitator condition, and it was significantly lower in agent facilitator condition than no facilitator

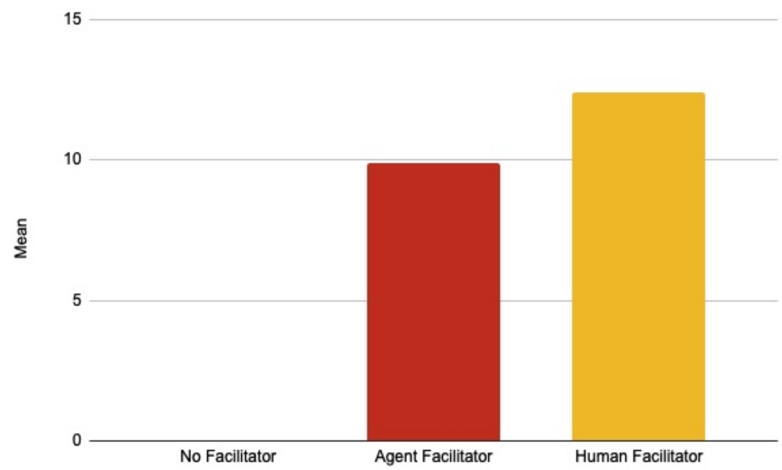


Figure 5.13: Team process facilitator measure across the three conditions

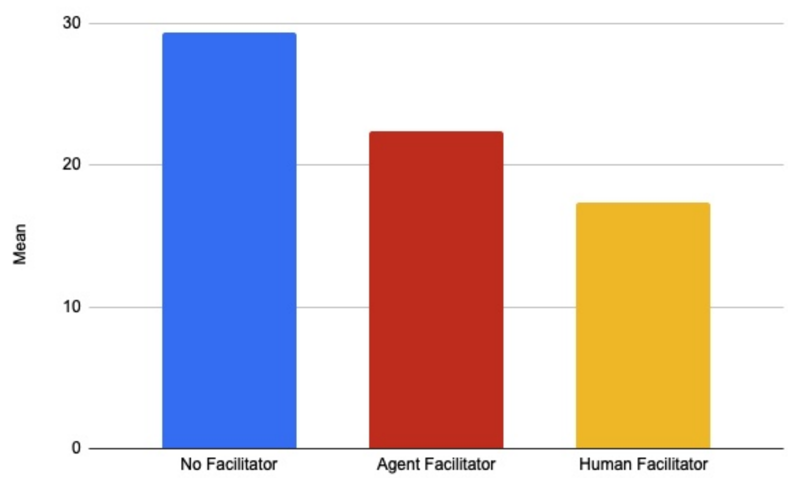


Figure 5.14: Team process bias score across the three conditions

condition ($p < 0.05$; See Figure 5.14). This finding supports the hypothesis (H3) that the facilitator interventions affect mitigating the three biases that is calculated in the TPS participant score (the lower the score the lower the biases).

These results indicate that facilitation behavior affects the negotiation positively and mitigates information pooling bias. Further, facilitation effect helps passing information and communication. These results, verify H3. Figures 5.15 and 5.16 show

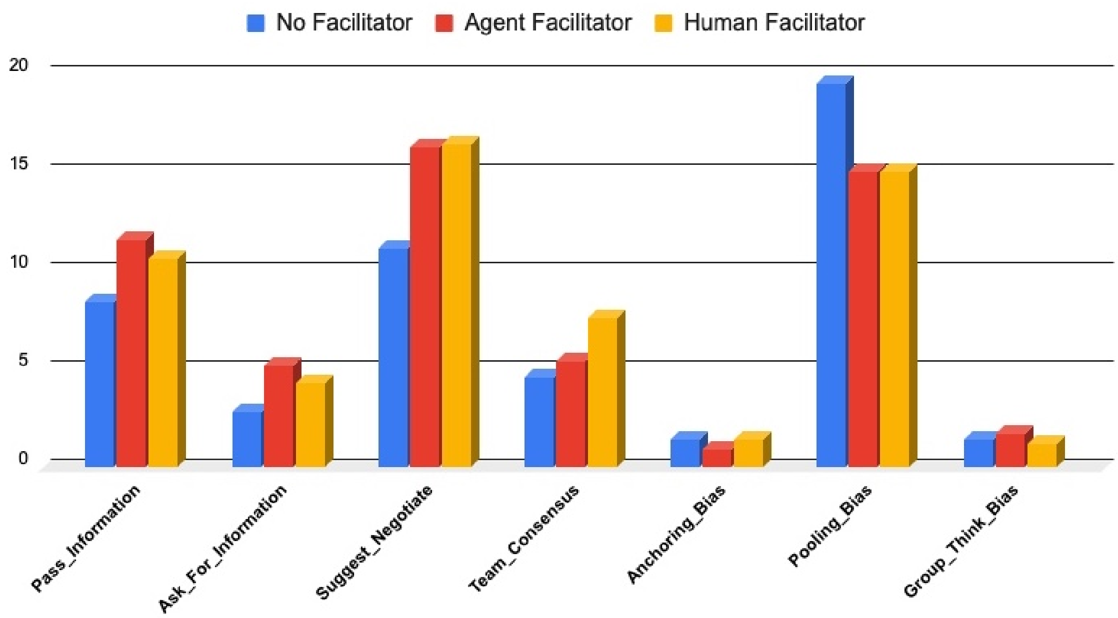


Figure 5.15: The Team Process Score (TPS) for the human participants in mission one based on average number of occurrences for each factor

the TPS for the human participants across conditions for each mission.

Figures 5.17 and 5.18 illustrate the team interaction with the automated planning agent across conditions for each mission. According to these results, the team asks more questions from the agent in the facilitation conditions. Moreover, the human facilitator effect is evident when the team asked the agent for unique information (See "Agent supplies unique" bar in both Figure 5.17 and Figure 5.18 and compare it with no facilitator condition). Finally, the facilitator, as seen before in outcome score and decision quality, has a positive effect in mitigating the agent information pooling bias due to a better communication of team with agent. These results, in part, verify H3.

To answer the second part of the question, I focus on human success in mitigating the information pooling bias mostly at task 4, which is route selection. According to the results, human facilitation is more suitable for mitigating information pooling and anchoring bias. On the other hand, agent facilitation is more ideal for objective

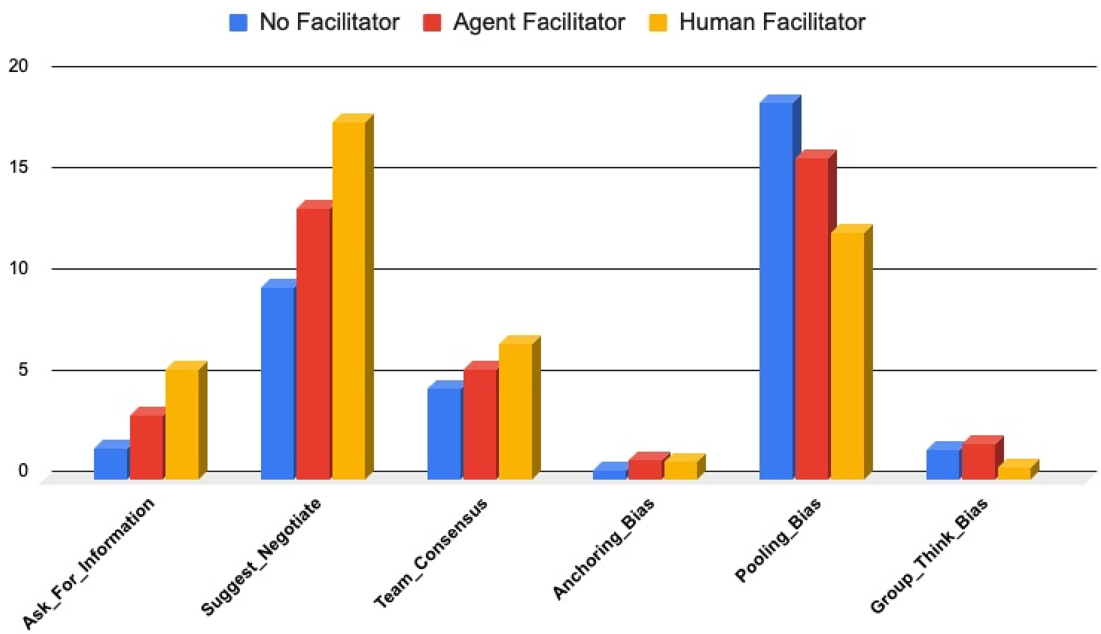


Figure 5.16: The Team Process Score (TPS) for the human participants in mission two based on average number of occurrences for each factor

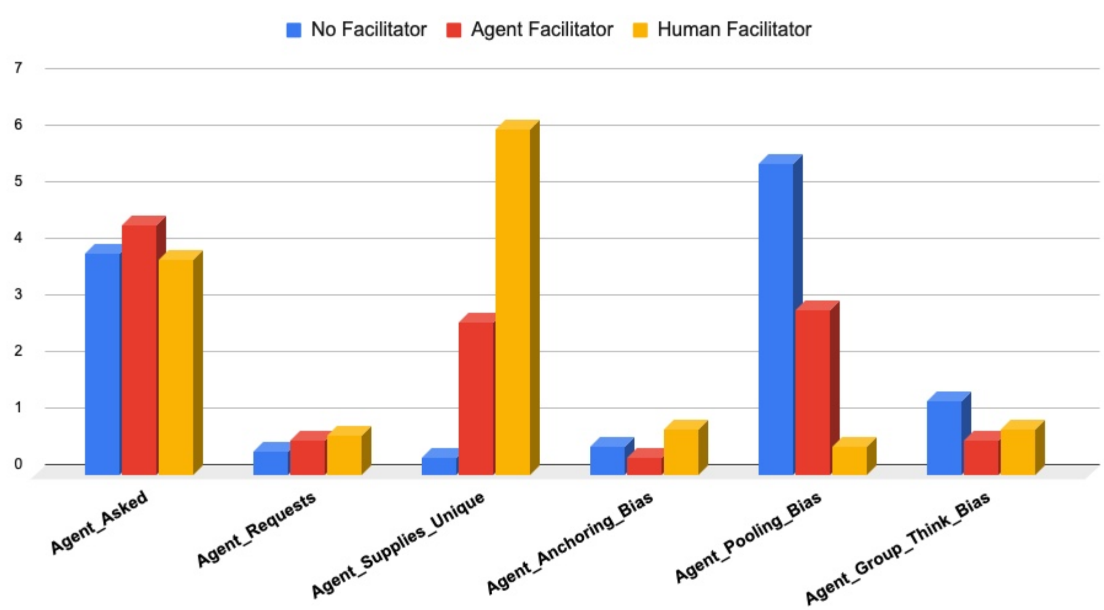


Figure 5.17: TPS for the automated planning agent in mission one based on average number of occurrences for each factor

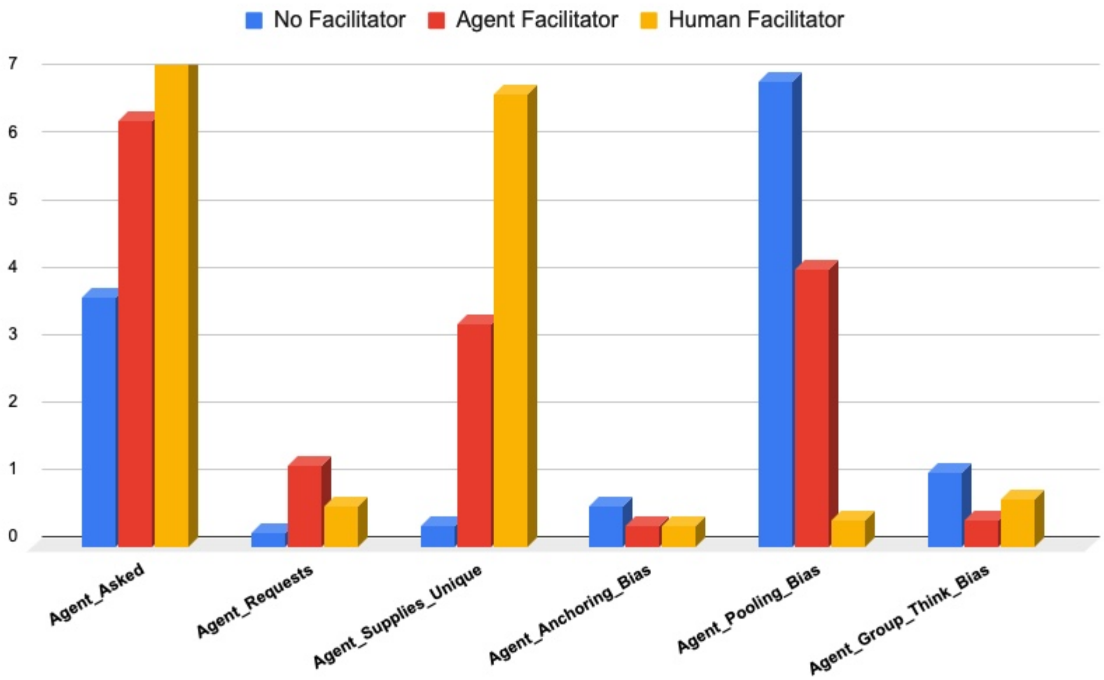


Figure 5.18: TPS for the automated planning agent in mission two based on average number of occurrences for each factor

metric calculation, such as time, distance, or the cost of the decisions.

The planning outcome and decision quality results suggest that Human teammates set a lower error margin for AI facilitator than the human counterpart. One explanation is that the human collaborators get annoyed faster by the AI facilitator and pay less attention to its interventions and probes (See Figure 5.9 and compare the route score in the human facilitator with the AI facilitator condition). Another explanation is that Human teammates expect the AI facilitator to lower their cognitive load by simulating their decisions.

On the other hand, focusing on TPSs analysis, the human facilitator focuses on the tone and personalization of the interventions more than the process. The human teammates perceive the human facilitator more as decision critique rather than process critique. Consequently, this observation suggests that human teammates perceive the human facilitator more as someone like them, while they perceive the AI facilitator as more functional. These results illustrate the importance of decision critique vs. process critique and suggest that there should be an equilibrium between the two to empower human decisions on such planning tasks. Together, the human and agent facilitation working together would create more successful facilitation, resulting in a deeper understanding of the task.

5.6 Discussion

In this chapter I stepped further from the previous research in human-robot teaming and introduced a new role, a facilitator. The facilitator can be viewed as a teacher, or coach that helps the team to communicate better to create a plan while facing a complex task in presence of different team biases. To better understand the facilitator impact, I designed a multi-agent planning testbed with two different logistical planning scenarios in an environment, where three humans join an automated planning agent (operated as the wizard of OZ) to provide a plan for the task. The result of this chapter illustrates the facilitator impact on situation awareness, mitigating biases, decision quality, and outcome scores. Consequently, the teams that empowered with a facilitator create a higher quality plan in terms of effectiveness (H1), as well as the decision quality (H2) compared to no facilitator condition. The results also show that facilitation influences the team communication quality, that result in superior teamwork (H3).

The results of this study envision new methods of human-AI collaboration to create effective human-AI teams. Consequently, to get to human-machine symbiosis AI should take on the roles that humans do not want to take on, instead of replicating humans.

Chapter 6

CONCLUSION

This dissertation creates motivating approaches for human-agent team designs. Throughout this dissertation, I have laid out different experiments to study the effect of communication in human-robot teams, both in terms of explanation and facilitation interventions. I have explored how to maintain collaboration engagement or cognitive readiness for collaborative planning tasks.

In Chapter 2, I created a general method of generating explicable actions to make explainable plans for human-robot teams, where the human is an active teammate. To this end, I provided an optimization formula to consider the plan cost and the preconceptions that the human may have about the robot. Therefore the robot actions may incur a higher cost for the robot, while mostly understandable for humans. This approach is generalizable to large teams learning about the human capability model. As a result, the produced plans will be both explicable and predictable.

In Chapter 3, I introduced a novel formulation for explanation generation by creating sub-explanations. This approach focused on reducing the mental workload for the human to interpret a complex explanation. I have provided three different methods and formulation. This method is evaluated both in simulations and with human subjects while improved task performance and reduced mental workload.

In Chapter 4, I have created an Inverse Reinforcement Learning (IRL) method to learn human preferences of the information order. I have adopted the IRL formulation to learn the reward function of a goal-based MDP. I show that humans indeed demonstrate preferences for the information order in explanations and employ the theory of mind. We can learn the human desires and beliefs using our framework.

Further, the results verified that the explanations generated by this approach improve task performance and reduce cognitive load.

Chapter 5 provides a novel rule-based path to explore the design of an AI facilitator. A facilitator helps a team collaborate on creating a plan in the presence of team biases and ineffective communication, on naming a few. I study the importance of the facilitator on the team outcome. Further, I provide a machine-learning approach to learn the weights to evaluate plans based on their practicality and effectiveness. This evaluation bridges between a rule-based model and the previous model-based approaches.

Chapter 7

FUTURE WORK

To better understand the facilitation impact on the planning and the human perception of agent persona and abilities, further analysis and studies are required. One direction is to find (if) there is a sweet spot to achieve the most desirable planning outcome.

Sweet spot: A point, range, or particular set of constraints will achieve the most desirable [or valuable] interaction for planning outcome.

Assuming there is a sweet spot, the goal is to find the optimal trade-off of cognitive workload versus team performance based on various utility measurements. Hence, one of the future directions of the proposed framework is to learn to interact with autonomy (throughout practice) and increasing team performance.

Accordingly, for future studies, we aim to validate the following hypotheses:

- **Hypothesis 1.** There is a sweet spot to align the cognitive workload and team performance.

- **Hypothesis 2.** The facilitation process needs to adapt to the extent of the team's cognitive workload (or the key individual in the group).

- **Hypothesis 3.** Onboarding a new member affects team collaboration. Therefore, the facilitation threshold varies for different teams.

Moreover, creating a cloud-based framework for the AI facilitator that helps the human facilitator is one of the most important future directions. Exploring ways to engage the human facilitator to the machine suggestions will open up new intuitions for facilitation. Such framework will provide a closer insight on how different ideas find their way from decision-making systems to impact our life when take effect in a large scale, such as in societies. Lastly, by performing a proper learning on facilitation behavior, the embedded features in the task that affect ideal facilitation can be extracted. These features can be used to transfer facilitation experience from one domain to another, or to fine-tune the facilitation behavior on a specific task based on a generalized model similar to Vaswani *et al.* (2017).

REFERENCES

Abbeel, P. and A. Y. Ng, “Apprenticeship learning via inverse reinforcement learning”, in “Proceedings of the twenty-first ICML”, p. 1 (ACM, 2004).

Andersen, R. S., O. Madsen, T. B. Moeslund and H. B. Amor, “Projecting robot intentions into human environments”, in “2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)”, pp. 294–301 (IEEE, 2016).

Brewka, G., “Artificial intelligence—a modern approach by stuart russell and peter norvig, prentice hall. series in artificial intelligence, englewood cliffs, nj.”, The Knowledge Engineering Review **11**, 1, 78–79 (1996).

Chakraborti, T., G. Briggs, K. Talamadupula, Y. Zhang, M. Scheutz, D. Smith and S. Kambhampati, “Planning for serendipity”, in “Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on”, pp. 5300–5306 (IEEE, 2015).

Chakraborti, T., S. Kambhampati, M. Scheutz and Y. Zhang, “Ai challenges in human-robot cognitive teaming”, arXiv preprint arXiv:1707.04775 (2017a).

Chakraborti, T., A. Kulkarni, S. Sreedharan, D. E. Smith and S. Kambhampati, “Explicability? legibility? predictability? transparency? privacy? security? the emerging landscape of interpretable agent behavior”, in “Proceedings of the Twenty-Ninth International Conference on Automated Planning and Scheduling, ICAPS 2018, Berkeley, CA, USA, July 11-15, 2019”, edited by J. Benton, N. Lipovetzky, E. Onaindia, D. E. Smith and S. Srivastava, pp. 86–96 (AAAI Press, 2019), URL https://aaai.org/ojs/index.php/ICAPS/article/view/3463.

Chakraborti, T., S. Sreedharan, Y. Zhang and S. Kambhampati, “Plan explanations as model reconciliation: Moving beyond explanation as soliloquy”, in “Proceedings of IJCAI”, (2017b).

Chakraborti, T., S. Sreedharan, Y. Zhang and S. Kambhampati, “Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. ijcai (2017), 156–163”, (2017c).

Chakraborti, T., Y. Zhang, D. E. Smith and S. Kambhampati, “Planning with resource conflicts in human-robot cohabitation”, in “Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems”, pp. 1069–1077 (International Foundation for Autonomous Agents and Multiagent Systems, 2016).

Cirillo, M., L. Karlsson and A. Saffiotti, “Human-aware task planning for mobile robots”, in “Advanced Robotics, 2009. ICAR 2009. International Conference on”, pp. 1–7 (IEEE, 2009).

Cooke, N. J., “Team cognition as interaction”, Current directions in psychological science **24**, 6, 415–419 (2015).

Cooke, N. J., M. L. Hilton *et al.*, *Enhancing the effectiveness of team science* (National Academies Press Washington, DC, 2015).

Cowan, N., "What are the differences between long-term, short-term, and working memory?", Progress in brain research **169**, 323–338 (2008).

Do, M. B. and S. Kambhampati, "Sapa: A multi-objective metric temporal planner", J. Artif. Intell. Res.(JAIR) **20**, 155–194 (2003).

Dragan, A. and S. Srinivasa, "Generating legible motion", in "Robotics: Science and Systems", (2013).

Dragan, A. D., K. C. Lee and S. S. Srinivasa, "Legibility and predictability of robot motion", in "Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction", pp. 301–308 (IEEE Press, 2013).

Endsley, M. R., "Design and evaluation for situation awareness enhancement", in "Proceedings of the Human Factors Society annual meeting", vol. 32, pp. 97–101 (SAGE Publications Sage CA: Los Angeles, CA, 1988).

Endsley, M. R., *Designing for situation awareness: An approach to user-centered design* (CRC press, 2016).

Ericsson, K. A. and J. Smith, *Toward a general theory of expertise: Prospects and limits* (Cambridge University Press, 1991).

Fern, A., S. Natarajan, K. Judah and P. Tadepalli, "A decision-theoretic model of assistance.", in "IJCAI", pp. 1879–1884 (2007).

Fikes, R. E. and N. J. Nilsson, "Strips: A new approach to the application of theorem proving to problem solving", Artificial intelligence **2**, 3-4, 189–208 (1971).

Fischer, B. B. and L. Fischer, "Styles in teaching and learning", Educational leadership **36**, 4, 245–254 (1979).

Fox, M., A. Gerevini, D. Long and I. Serina, "Plan stability: Replanning versus plan repair.", in "ICAPS", vol. 6, pp. 212–221 (2006).

Fox, M. and D. Long, "Pddl2. 1: An extension to pddl for expressing temporal planning domains", Journal of artificial intelligence research **20**, 61–124 (2003).

Fox, M., D. Long and D. Magazzeni, "Explainable planning", arXiv preprint arXiv:1709.10256 (2017).

Göbelbecker, M., T. Keller, P. Eyerich, M. Brenner and B. Nebel, "Coming up with good excuses: What to do when no plan can be found", in "ICAPS", (2010).

Gong, Z. and Y. Zhang, "Behavior explanation as intention signaling in human-robot teaming", in "2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)", pp. 1005–1011 (IEEE, 2018).

Gong, Z. and Y. Zhang, "What is it you really want of me? generalized reward learning with biased beliefs about domain dynamics.", in "AAAI", pp. 2485–2492 (2020).

Gunning, D., "Explainable artificial intelligence (xai)", Defense Advanced Research Projects Agency (DARPA), nd Web (2017).

Hanheide, M., M. Göbelbecker, G. S. Horn, A. Pronobis, K. Sjöö, A. Aydemir, P. Jensfelt, C. Gretton, R. Dearden, M. Janicek *et al.*, "Robot task planning and explanation in open and uncertain worlds", Artificial Intelligence **247**, 119–150 (2017).

Hoffmann, J. and B. Nebel, "The FF planning system: Fast plan generation through heuristic search", Journal of Artificial Intelligence Research **14**, 253–302 (2001).

IPC, "Competition domains for international planning committee", `http://www.plg.inf.uc3m.es/ipc2011-learning/Domains.html` (2019).

Kahneman, D., *Thinking, fast and slow* (Macmillan, 2011).

Kahneman, D., S. P. Slovic, P. Slovic and A. Tversky, *Judgment under uncertainty: Heuristics and biases* (Cambridge university press, 1982).

Kautz, H. A. and J. F. Allen, "Generalized plan recognition.", in "AAAI", vol. 86, p. 5 (1986).

Knepper, R. A., C. I. Mavrogiannis, J. Proft and C. Liang, "Implicit communication in a joint action", in "Proceedings of the 2017 acm/ieee international conference on human-robot interaction", pp. 283–292 (ACM, 2017).

Langfred, C. W., "Too much of a good thing? negative effects of high trust and individual autonomy in self-managing teams", Academy of management journal **47**, 3, 385–399 (2004).

Langley, P., B. Meadows, M. Sridharan and D. Choi, "Explainable agency for intelligent autonomous systems", in "Twenty-Ninth IAAI Conference", (2017).

Levenshtein, V., "Binary codes capable of correcting deletions, insertions, and reversals", in "Soviet physics doklady", vol. 10, pp. 707–710 (1966).

Lien, J., O. B. Bayazit, R. T. Sowell, S. Rodriguez and N. M. Amato, "Shepherding behaviors", in "ICRA", (IEEE, 2004).

Likhachev, M., D. I. Ferguson, G. J. Gordon, A. Stentz and S. Thrun, "Anytime dynamic a*: An anytime, replanning algorithm.", in "ICAPS", vol. 5, pp. 262–271 (2005).

Lombrozo, T., "The structure and function of explanations", Trends in cognitive sciences **10**, 10, 464–470 (2006).

Maxwell, B. A., L. A. Meeden, N. Addo, L. Brown, P. Dickson, J. Ng, S. Olshfski, E. Silk and J. Wales, "Alfred: The robot waiter who remembers you", in "Proceedings of AAAI workshop on robotics", (AAAI Press, 1999).

Miller, T., “Explanation in artificial intelligence: Insights from the social sciences”, Artificial Intelligence (2018).

NASA, “Nasa task load index”, `https://humansystems.arc.nasa.gov/groups/TLX/` (2019).

Ng, A. Y. and S. J. Russell, “Algorithms for inverse reinforcement learning”, in “ICML”, pp. 663–670 (Morgan Kaufmann Publishers Inc., 2000).

Nikolaidis, S., K. Gu, R. Ramakrishnan and J. Shah, “Efficient model learning for human-robot collaborative tasks”, arXiv preprint arXiv:1405.6341 (2014).

Paas, F., A. Renkl and J. Sweller, “Cognitive load theory and instructional design: Recent developments”, Educational psychologist **38**, 1, 1–4 (2003).

Pérez-D’Arpino, C., R. P. Khurshid and J. A. Shah, “Experimental assessment of human-robot teaming for multi-step remote manipulation with expert operators”, arXiv preprint arXiv:2011.10898 (2020).

Petty, R. E. and J. T. Cacioppo, “Effects of forwarning of persuasive intent and involvement on cognitive responses and persuasion”, Personality and Social Psychology Bulletin **5**, 2, 173–176 (1979).

Rajivan, P. and N. J. Cooke, “Information-pooling bias in collaborative security incident correlation analysis”, Human factors **60**, 5, 626–639 (2018).

Rajivan, P., M. A. Janssen and N. J. Cooke, “Agent-based model of a cyber security defense analyst team”, in “Proceedings of the human factors and ergonomics society annual meeting”, vol. 57, pp. 314–318 (SAGE Publications Sage CA: Los Angeles, CA, 2013).

Ramırez, M. and H. Geffner, “Probabilistic plan recognition using off-the-shelf classical planners”, in “Proceedings of the Conference of the Association for the Advancement of Artificial Intelligence (AAAI 2010)”, pp. 1121–1126 (2010).

Reddy, S., A. Dragan and S. Levine, “Where do you think you’re going?: Inferring beliefs about dynamics from behavior”, in “NeurIPS”, pp. 1461–1472 (2018).

Schwarz, C. V., B. J. Reiser, E. A. Davis, L. Kenyon, A. Achér, D. Fortus, Y. Shwartz, B. Hug and J. Krajcik, “Developing a learning progression for scientific modeling: Making scientific modeling accessible and meaningful for learners”, Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching **46**, 6, 632–654 (2009).

Sohrabi, S., J. A. Baier and S. A. McIlraith, “Preferred explanations: Theory and generation via planning”, in “Twenty-Fifth AAAI Conference on Artificial Intelligence”, (2011).

Stasser, G. and W. Titus, “Pooling of unshared information in group decision making: Biased information sampling during discussion.”, Journal of personality and social psychology **48**, 6, 1467 (1985).

Tsang, P. S. and V. L. Velazquez, “Diagnosticity and multidimensional subjective workload ratings”, Ergonomics **39**, 3, 358–381 (1996).

Tsintsadze-Maass, E. and R. W. Maass, “Groupthink and terrorist radicalization”, Terrorism and Political Violence **26**, 5, 735–758 (2014).

Tversky, A. and D. Kahneman, “Judgment under uncertainty: Heuristics and biases”, science **185**, 4157, 1124–1131 (1974).

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, “Attention is all you need”, in “Advances in neural information processing systems”, pp. 5998–6008 (2017).

Zakershahrak, M., Z. Gong, N. Sadassivam and Y. Zhang, “Online explanation generation for human-robot teaming”, arXiv preprint arXiv:1903.06418 (2019).

Zakershahrak, M., S. R. Marpally, A. Sharma, Z. Gong and Y. Zhang, “Order matters: Generating progressive explanations for planning tasks in human-robot teaming”, arXiv preprint arXiv:2004.07822 (2020).

Zakershahrak, M., A. Sonawane, Z. Gong and Y. Zhang, “Interactive plan explicability in human-robot teaming”, in “RO-MAN”, pp. 1012–1017 (IEEE, 2018).

Zhang, Y., S. Sreedharan and S. Kambhampati, “Capability models and their applications in planning”, in “Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems”, pp. 1151–1159 (International Foundation for Autonomous Agents and Multiagent Systems, 2015a).

Zhang, Y., S. Sreedharan, A. Kulkarni, T. Chakraborti, H. H. Zhuo and S. Kambhampati, “Plan explicability for robot task planning”, in “Proceedings of the RSS Workshop on Planning for Human-Robot Interaction: Shared Autonomy and Collaborative Robotics”, (2016).

Zhang, Y., S. Sreedharan, A. Kulkarni, T. Chakraborti, H. H. Zhuo and S. Kambhampati, “Plan explicability and predictability for robot task planning”, in “Robotics and Automation (ICRA), 2017 IEEE International Conference on”, pp. 1313–1320 (IEEE, 2017).

Zhang, Y. and M. Zakershahrak, “Progressive explanation generation for human-robot teaming”, arXiv preprint arXiv:1902.00604 (2019).

Zhang, Y., H. H. Zhuo and S. Kambhampati, “Plan explainability and predictability for cobots”, CoRR abs/1511.08158 (2015b).

Ziebart, B. D., A. Maas, J. A. Bagnell and A. K. Dey, “Maximum entropy inverse reinforcement learning”, in “Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3”, AAAI’08, p. 1433–1438 (AAAI Press, 2008).