Mediating Trust and Influence in Human-Robot Teams via Multimodal Communication and Explanation for Mental Model Alignment

by

Aaquib Tabrez

M.S., University of Colorado Boulder, 2019

B.Tech., National Institute of Technology Karnataka, 2014

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2024

Committee Members:

Bradley Hayes, Chair

Alessandro Roncone

Nisar Ahmed

Daniel Szafir

Sonia Chernova

Tabrez, Aaquib (Ph.D., Computer Science)

Mediating Trust and Influence in Human-Robot Teams via Multimodal Communication and Explanation for Mental Model Alignment

Thesis directed by Prof. Bradley Hayes

Effective communication is a foundational aspect of collaboration and teaming. Good communication enables and sustains the shared situational awareness necessary for adaptation and coordination during uncertain situations in human-robot teaming, and helps identify and remedy potential misunderstandings caused by mismatched expectations or behavior. Given the opaque nature of decision-making in autonomous systems and robots, it is crucial that these agents can explain their decision-making rationale to both experts and novices for safe and trustworthy deployment in real-world applications. Furthermore, for autonomous agents to be effective and capable in human-robot teams, they should not only explain their decisions but also have the capacity to coach and convince their human collaborators.

I argue that by leveraging multiple modalities of communication (such as visual and natural language), we can improve the safety and capability of human-robot teams, enabling appropriate trust, compliance, and reliance, especially in safety-critical, partially observable situations. Therefore, this doctoral thesis focuses on improving human-machine multimodal communication by employing explainable AI techniques to empower autonomous agents to: 1) communicate insights into their capabilities and limitations to human collaborators, 2) coach and influence human teammates' behavior during joint task execution, and 3) successfully convince and mediate trust in human-robot interactions.

Dedication

"Hope" is the thing with feathers
That perches in the soul
And sings the tune without the words
And never stops - at all
– Emily Dickinson

This thesis is dedicated to my parents, friends, family, and loved ones for believing in and supporting me throughout my PhD journey. I would especially like to dedicate this to my dad, who left us too soon.

Acknowledgements

This is my sincere attempt to thank those who have helped me find myself in my research, supported me during my lows, and laughed with me during my highs. First, I express my deepest gratitude to my advisor, Dr. Bradley Hayes, who took me in as a budding mechie with little knowledge of computer science. That decision changed my life, and he not only helped me become a great researcher but also showed me how to be a good human being. Working with him for six years was a genuine honor. I also thank my thesis committee members: Alessandro Roncone, Nisar Ahmed, Daniel Szafir, and Sonia Chernova, for teaching me how to contextualize my research efficiently, see the heart of any research problem, and guide my research in a better direction.

I would like to give special thanks to all my lab mates and collaborators in the CAIRO lab who showed me how fulfilling it is to work with kind and supportive people. I am especially grateful to Matt, Yi-Shiuan, Shivendra, Kyler, and Himanshu—whom I approached to share problems, brainstorm whenever I was stuck, or introduce me to new and fun stuff. I also apologize to them for subjecting them to endless skepticism, whether it was about research, philosophy, or politics.

I am extremely grateful to be blessed with amazing friends who have been bundles of energy, showing me how to live the best moments for myself, and always being there for me when I needed them. To my friends Shoaib, Tanveer, Sohail, Akash, Pratik, Mustafa, Hasil, Laksh, Sandesh, Vimal, and Khoi. Finally, I would like to thank my family for supporting me and sending me from India to pursue my dreams and passion. I would like to thank my Mom and Dad, my brothers Parveez and Shoaib Bhaiyya, and my sisters-in-law Farheen and Abeer Bhabhi. I am extremely grateful for your love and support.

I also thank the U.S. Army Research Lab STRONG program (#W911NF-20-2-0083) and the National Science Foundation for funding this work.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

"The mind is its own place, and in itself can make a heaven of hell, a hell of heaven."
— John Milton, *Paradise Lost*

1.1 Background and Motivation

As autonomous systems and robots become increasingly capable decision-makers, explainable AI (xAI) has emerged as a necessary component for deploying safe autonomous systems. xAI is a subfield of artificial intelligence and machine learning that focuses on developing methods that enable human users to comprehend, trust, and effectively interact with autonomous systems by providing clear, understandable, and transparent explanations of their decision-making processes [1, 2]. The goal of xAI is to make machine learning algorithms and their outputs interpretable, ensuring that users can understand and trust how and why specific decisions or recommendations are made [3]. Most modern machine learning algorithms (e.g., reinforcement learning and deep learning) are considered "black box" models. Their decision-making processes are highly complex (e.g., GPT-3.5, one of the earlier versions of ChatGPT, has 6.7 billion parameters [4]) and are understood only based on their inputs and outputs, not their internal workings [5].

Explainable AI can help bridge the gap between human and autonomous agents by making complex models more understandable. This transparency facilitates faster debugging and failure recovery, builds trust, and enhances collaboration, ultimately improving overall team performance [6, 7, 8]. xAI is also crucial for meeting legal requirements. Regulations such as the General Data Protection Regulation (GDPR) in the European Union, implemented in 2018, mandate trans-

parency and accountability in automated decision-making processes [9]. Similarly, the proposed Algorithmic Accountability Act in the United States, introduced in 2022, seeks to ensure that AI systems are fair and unbiased, requiring impact assessments and explanations for decisions made by AI [10, 11].

Explainability and Aligning in Human-Robot Teaming. In the context of robotics, explainability and transparency are crucial for the safe and trustworthy deployment of autonomous systems in the real world, especially when they are working with or around people [12]. Traditionally, robots have operated separately from humans. Even in potentially collaborative environments like manufacturing, industrial robots most often function in physically separated sections of the assembly floor. The primary reason for this separation is the need for safety assurances; it is essential to be certain that these machines will operate as intended, without causing harm to humans, their environment, or even themselves [13, 14]. Additionally, robots must be able to plan for uncertainty and communicate their future behavior and decision-making rationale to people so they can appropriately trust and rely on them for safe deployment into the real world.

One critical aspect of safe and effective collaboration between teammates is maintaining awareness of the collaborator's mental model, enabling agents to reason about what their teammate is likely to do or need [12, 15]. Humans tend to be adept at this task, able to communicate plans and preferences in ways that are easily understandable by their teammates [16]. Robots, however, do not have the benefit of human intuition. They must instead rely on explicit mathematical formalisms to approximate the mental states of human teammates and plan accordingly [17]. Recent research in explainability and human-agent teaming has leveraged xAI for knowledge sharing and expectation matching to achieve fluent collaboration and improve shared awareness [18, 19, 20, 21]. Explanations enhance transparency and help synchronize expectations between human and robot teams [12, 22]. For robots and autonomous agents to effectively collaborate with humans in high-stakes applications (e.g., autonomous driving), insights into these autonomous systems' capabilities and their limitations are required [23, 24, 25]. Therefore, one research thrust of this thesis focuses on developing novel explainable AI techniques to provide those insights, enabling more fluent teaming

and agent-to-human communication.

Explanations and Communication for Robotic Coaching. Explanations and communication can be leveraged to empower autonomous agents to manipulate, coach, and adapt their teammates' behavior, particularly during joint task execution. The capacity to coach is crucial in applications such as robotic tutors, learning assistants, healthcare for the elderly, and rehabilitation therapy [23, 26, 27]. It is also powerful in human teaming scenarios such as search and rescue, enabling agents to perform functions like robotic coaching and intervening when individuals are about to undertake suboptimal actions or actions that could lead to failure due to uncertainty or sudden environmental changes. To enable robotic coaching of humans, agents should not only be able to communicate or explain but also be proficient in mediating any potential misunderstandings and justifying their rationale for recommendations. Consequently, this facilitates the explicit reconciliation of mental model divergences, informs collaborators as requirements change, and enhances overall collaboration effectiveness. Consider the problem of air traffic control, where a human air traffic controller oversees the movement of multiple aircraft in a busy airspace. If the controller gives a sub-optimal routing instruction that could lead to potential conflicts or inefficient flight paths, a system capable of generating human-interpretable feedback indicating the potential conflict and providing a justifying explanation would be far more useful than one that could not. Such a capability has the potential to improve both the controller's situational awareness and the overall safety and efficiency of air traffic management.

Donald Michie, one of the forefathers of Artificial Intelligence at Bletchley Park, outlined criteria for machine learning in his seminal 1988 work: weak, strong, and ultra-strong* . Michie aimed to establish operational criteria that assess not only the predictive accuracy of machine learning systems but also the comprehensibility of the knowledge they acquire. Many of today's systems with learned control policies continue to satisfy only Michie's weak criterion, meaning

* In 1980, Donald Michie proposed three criteria to evaluate machine learning systems [28]:
1. Weak Criterion: A system increases performance on unseen data by learning from sample data.
2. Strong Criterion: The system contains the weak criterion plus the ability to communicate its learned hypotheses function in symbolic form.
3. Ultra-strong Criterion: The system contains the strong criterion plus the ability to teach a user the learned hypothesis function.

they still cannot communicate their rationale, despite the recognition of explainability as crucial for the safe and transparent deployment of autonomous systems. This underscores the need for developing xAI techniques that not only explain their rationale but are also adept at teaching their understanding to users, thereby satisfying the ultra-strong criterion. Therefore, the second research thrust of this thesis focuses on enabling human coaching by leveraging explanations and communication.

Human-Centered Explainable AI. Research in xAI has primarily targeted algorithm transparency for developers, aiding in model debugging and behavior prediction [7, 29]. These approaches are fundamentally limiting to non-expert stakeholders and end-users who interact with the models or products regularly, and thus directly experience the consequences of failures [3, 30].

For example, consider a healthcare application where a medical diagnosis system uses deep learning to identify diseases from medical images. While developers can use xAI methods to understand how the model makes its predictions and improve it, end users—doctors and patients—need simpler, more social explanations, such as those in natural language, to trust the system and make informed decisions in critical medical scenarios [31, 32]. Similarly, in robotics, imagine a collaborative manufacturing environment where robots work alongside human workers. If a robot's behavior deviates unexpectedly, xAI techniques can provide explanations for its actions, helping workers understand whether the deviation was due to a safety protocol, a sensor error, or a change in task parameters. This understanding not only builds trust but also enables workers to respond appropriately and maintain efficient and safe operations.

Therefore, one of the focuses of this thesis dissertation is generating multimodal explanations that are understandable for both experts and non-experts, drawing insights from everyday human-centric explanations extensively studied in social science and psychology literature[31, 32, 33].

Another challenge in human-machine collaboration scenarios, such as decision support systems, is that people are governed by psychological biases, leading to over-trust (over-reliance) or under-trust, resulting in suboptimal decisions [34, 35, 36, 37]. Over-trust can cause performance failures due to inadequate monitoring and workload delegation, such as neglecting to monitor

traffic when using driver assistance systems in autonomous driving scenarios. Conversely, under-trust occurs when users are overly skeptical of autonomous systems, leading to underutilization of their capabilities. This skepticism can result in users frequently overriding or ignoring the robots' recommendations, causing missed opportunities for improved efficiency and decision-making. We argue that the goal of these systems should be to appropriately calibrate trust, ensuring that trust matches system capabilities and promotes appropriate use. Therefore, this thesis examines how we can leverage behavioral insights from cognitive psychology and the human factors community, combined with multimodal explanations, to foster appropriate trust levels and enhance human-machine collaboration.

Research Themes. With this in mind, I pursue three interconnected research themes at the intersection of xAI and human-robot interaction:

RT1: Characterizing and generating multimodal explanations for autonomous agents to effectively communicate their decision-making rationales.

RT2: Operationalizing a framework for explainable robotic coaching within human-robot teaming scenarios, aiming to enable mental model reconciliation through effective communication.

RT3: Characterizing and evaluating the role of robot justification in mediating trust and influence within human-machine teams to achieve mental model alignment.

1.2 Thesis Statement

This dissertation argues that to be effective teammates, robots should be capable of explaining, coaching, and convincing via multimodal communication to build appropriate trust and reliance. It investigates different characterizations of generating multimodal explanations (visual and natural language) for autonomous agents to communicate their decision-making rationale. Furthermore, robots should not only communicate their rationale but also coach these aspects to their human teammates to improve task understanding and performance, satisfying ultra-strong criteria

[28]. Therefore, this dissertation also explores operationalizing explainable robot coaching within human-robot teaming scenarios by effectively modeling and reconciling mental model divergences using explanations, thereby building trust and transparency. Finally, the dissertation examines the role of robot justification in mediating trust and influence, particularly the psychological effects such as compliance and reliance, based on different justification characteristics like modality, frequency, and content.

1.3 Contributions

The key technical contributions of this thesis dissertation are as follows:

(1) **Reward Augmentation and Repair through Explanation (RARE):** A novel framework for understanding and correcting an agent's decision-making process, which estimates a human's understanding of a domain's reward function through their behavior and provides corrective explanations to repair detected issues. This framework was validated through a human subjects study, showcasing the effectiveness of justification in convincing people.

(2) **Single-shot Policy Elicitation for Augmenting Rewards (SPEAR):** A novel sequential optimization algorithm uses semantic explanations derived from combinations of planning predicates to augment a human agent's reward function. This process, defined as policy elicitation, drives the agent's actions (policy) to exhibit more optimal behavior and reconciles disparities in their reward function. We validated this policy elicitation framework through a series of human subjects studies, demonstrating that reward-based explanations improve task performance and promote active thinking patterns.

(3) **Plan Augmentation and Repair through SEmantic Constraints (PARSEC):** A human-in-the-loop algorithm that facilitates constraint annotation by novice users using natural language for motion planning problems through a novel hierarchical semantic process for robot skill learning and repair.

(4) **AR-based Visual Guidance for Multi-agent Reinforcement Learning:** MARS (Min-entropy Algorithm for Robot-supplied Suggestions), a framework for generating augmented reality-based visual guidance to communicate environmental uncertainty and provide actionable recommendations in joint human-robot tasks. This framework was empirically validated through a human subjects study, showcasing the effectiveness of these visual explanations in improving trust and enabling active engagement in the task.

(5) **Mathematical Framework for Justification Timing:** A novel mathematical framework, informed by the value of information theory, to decide when a robot collaborator should justify its recommendation to a human teammate. This framework was validated by an expert study, determining the utility of justification timing strategies.

(6) **Characterization and Validation of Justification Types:** A methodological characterization of four different types of justification, derived from established features in xAI literature, along with a validation and analysis of these justification types via human subjects study.

1.4 Outline

This document is divided into six chapters. The chapters are described below:

(1) Chapter 2 begins with a review of the literature on aligning mental models in human-robot teaming, focusing primarily on technical methods for mental modeling and aligning mental models within the context of explainable AI and human-robot interaction.

(2) Chapter 3 focuses on an explanation-based human reward coaching framework. Specifically, it explores two coaching frameworks: correcting one action at a time and policy coaching (i.e., improving overall behavior). This chapter also discusses how to generate natural language explanations to enhance human teammates' task understanding and assesses the benefits in human-robot interaction scenarios, particularly in the context of intervention and building appropriate trust.

(3) Chapter 4 examines how natural language communication can be used by novice users to quickly and effectively select constraints to correct faulty robot behavior or adapt robotic skills to human preferences and personalization.

(4) Chapters 5 and 6 investigate multimodal explanations for decision support systems and their role in mental model alignment and justification. In Chapter 5, we present a multi-agent collaborative planning and decision support system for human teammates that leverages visual explanations to influence human thought patterns for compliance and reliance. Chapter 6 looks into the utility of justifications in human-machine teaming scenarios, exploring when and what should be included in them. It presents a framework that determines the timing and evaluates what should go within justifications based on context, utilizing both visual and natural language explanations.

(5) Lastly, Chapter 7 concludes with a summary of the contributions of these works and a discussion of future research avenues enabled by these contributions.

Chapter 2

Mental Modeling in Human-Robot Teaming

"Madness is to think of too many things in succession too fast, or of one thing too exclusively."

— Voltaire, *Candide*

2.1 Introduction

This chapter focuses on characterizing recent work in developing formalisms for **mental models** in human-robot teaming scenarios. As robots become increasingly prevalent and capable, the complexity of roles and responsibilities assigned to them as well as our expectations for them will increase in kind. For these autonomous systems to operate safely and efficiently in human-populated environments, they will need to cooperate and coordinate with human teammates. Mental models provide a formal mechanism for achieving fluent and effective teamwork during human-robot interaction by enabling awareness between teammates and allowing for coordinated action. Much recent research in human-robot interaction has made use of standardized and formalized mental modeling techniques to great effect, allowing for a wider breadth of scenarios in which a robotic agent can act as an effective and trustworthy teammate. This chapter provides a structured overview of mental model theory and methodology as applied to human-robot teaming. It also examines mental model alignment in human-machine teaming within the context of explainable AI techniques and communication. The chapter also discusses evaluation methods and metrics for various aspects of mental modeling during human-robot interaction, along with recent emerging applications and open challenges in the field.

2.2 Mental Models

Mental models, also referred to as **mental representations** in psychology, are organized knowledge structures that allow individuals to interact with their environment [38]. Although the mental model has been used as an explanatory mechanism in a variety of disciplines over the years, its root can be traced back to twentieth-century psychology and epistemology. In 1943, Kenneth Craik posited in his seminal work that the mind provides a "small-scale model" of reality, enabling us to predict events [39]. In essence, mental models serve the crucial purpose of helping people to describe, explain, and predict events in their environment [40]. Since then, mental models have gained popularity in the human factors community for their effectiveness in eliciting and strengthening teamwork fluency for complex task execution, such as in tactical military operations [15, 41]. Inspired by this success, several architectures for HRI have since replicated this fluency and teamwork by developing mental modeling techniques for robotic agents that operate in human-populated environments.

In HRI literature, the concept of mental modeling is often conflated or used interchangeably with another important concept in developmental psychology: **Theory of Mind** (ToM). To be capable of ToM simply denotes an ability to attribute thought, desires, and intentions to others [42]. Theory of Mind is crucial for everyday human social interactions (e.g., for analyzing, judging, and inferring others' behaviors), with evidence that typically developing humans exhibit this capability by the age of 5 [43]. Accordingly, several architectures for human-robot teaming in HRI incorporate aspects of a ToM for other agents [26, 44, 45, 46, 47, 48].

In general, mental models and ToM go hand in hand during human-robot interaction, as a robot modeling other agents is analogous to having an agent with a ToM capacity. Furthermore, it leads to an interesting phenomenon during human-robot teaming as humans also form a ToM directed at their robot teammate. Therefore, mental modeling enables a phenomenon where a robot may form a belief over a human's mental model of the robot. This meta modeling is defined as second-order mental modeling which enables robots to estimate how a human's mental model is

affected by its own behavior [49]. Thus, current work in mental modeling for human-robot teaming can be broadly classified into first-order (or standard) or second-order mental models.

We can see how effective mental models correlate with team functioning: team members predict what their teammates will do or need, facilitating the coordination of actions. Prior studies in the human factors community demonstrate a positive relationship between team performance and similarity between the mental models of team members [40, 50, 51]. This implies that shared understanding of the team is a crucial factor of effective team performance (i.e., team members should have a shared mental model). Shared Mental Model (SMM) theory states that team members should hold compatible mental models that lead to common expectations for shared task execution to avoid failure [52, 53]. To summarise, if a mental model helps in describing, explaining, and predicting the behavior of a system, a shared mental model serves the purpose of describing, explaining, and predicting the behavior of a team.

2.3 Mental Models in Human-Robot Teaming

Teamwork is the collaborative effect of a group's effort toward achieving a common goal[54]. In the mental modeling literature, collaborative tasks are often broken up into smaller submodels representing components of effective teamwork, such as models of task procedures and strategies, models of inter-member interaction and information flow, or models of individual team member skill and preferences [40].

These various types of mental models and their incorporation of shared knowledge in teams help in achieving characteristic traits such as fluent behavior between teammates, quick adaptation to changing task demands, trusting collaborators with roles and responsibilities, effective communication, and decision making in time-critical applications. Several studies in human-robot collaboration have attempted to elicit these positive qualities through the use of mental models. In this section, we present a systematic characterization of desirable traits which can be achieved through mental modeling in human-robot teaming:

- **Fluent behavior**: Fluency, as defined by Hoffman, is a "coordinated meshing of joint activities between members of a well-synchronized team" [55]. This quality of interaction, collaborative fluency, intuitively means human and robot are well-synchronized in timing, they can alter plans and actions appropriately, and often without much communication.

- **Adaptability**: During collaboration, plans change, and team members (both human and robot) should be able to alter their plans and actions appropriately and dynamically as needed. Previous studies show that shared or common mental models can be leveraged for changing task demands for quick adaptation in a team [52, 56].

- **Trust building**: Trust is a critical element for the success of a team. In human-robot interaction, studies show that people trust a collaborative robot when they can discern its role and responsibility, have confidence in its capabilities, and possess an accurate understanding of its decision-making process (a shared mental model) [6, 24].

- **Effective communication**: Information exchange, either verbal or non verbal, is pivotal for collaboration. A collaborative agent can leverage mental models to warn its human teammate about potential failures or ask for help when it is unable to complete a task [20, 57].

- **Explainability**: Knowledge sharing and expectation matching also have importance for behavior explainability [58, 59, 60]. The recent surge in popularity of explainable AI (xAI) has shown the crucial importance of agents' ability to explain their decision-making process, leading to improved transparency, trust, and team performance.

2.4 Mental Model Methodologies

In this section, we discuss successful methods for mental modeling in human-robot teaming contexts. We organize the literature into three categories: first-order (or standard) mental models, second-order mental models, and shared mental models.

2.4.1 First-order Mental Models

In first-order mental models, robots model the behavior of human collaborators to infer their beliefs, intentions, and goals, for the purpose of predicting their actions. Usually, such modeling can be functionally broken down into two steps which a framework must resolve: 1) the human's reward function (which motivates the human's behavior in the world), and 2) a planning algorithm which connects that inferred reward function to robot behavior [61].

One of the simplest approaches is based on the principle of rationality [62, 63]: the expectation that agents will plan approximately rationally to achieve their goals, given their beliefs about the world (i.e., they will take actions that maximize their expected reward). One way to infer a human's reward function is to observe their behavior through inverse reinforcement learning (IRL). For example, the widely used maximum entropy IRL formulation optimizes a model to fit a reward function that incentivizes a human demonstrator's actions exponentially more than unobserved actions [64, 65].

A similar approach to inferring a human's reward function is through inverse planning. Baker et al. propose a computational framework based on Bayesian inverse planning for modeling human action understanding. They modeled human decision making as rational probabilistic planning with Markov decision processes (MDPs), and inverted this relation using Bayes' rule to infer agents' beliefs and goals from their actions (running the principle of rationality in reverse) [66, 67]. They were able to extend this method to a Bayesian model of Theory of Mind (BToM), which provides the predictive model of belief and desire-dependent action (the ToM capacity of the collaborative human) as a Partially Observable Markov Decision Process (POMDP) [68], and reconstructs an agent's joint belief state and reward function using Bayesian inference based on observations of the agent's behavior [69, 70].

From a planning and decision-making point of view, the noisy rational choice model (also known as Boltzmann rational) [71, 72] is a popular method in robotics where actions or trajectories are chosen in proportion to their exponentiated reward. Here, it is assumed that the collaborative

agent has access to some underlying human reward function (usually inferred through IRL or inverse planning approaches). The human is modelled to act rationally with the highest probability, but with a non-zero probability of behaving sub-optimally [49, 73, 74, 75, 76].

Humans frequently deviate from rational behavior due to specific biases such as time pressures, loss aversion, and the like [36]. Furthermore, they are limited in cognitive capacity, which leads to forgetfulness, limited planning horizons, and false beliefs. Some recent methods attempt to introduce these inconsistencies to the rational model assumption [77]. Nikolaidis et al. gave a Bounded-Memory Adaptation Model, which models humans as boundedly rational, subject to memory and recency constraints, through a probabilistic finite-state controller that captures human adaptive behaviors [48]. Kwon et al. used a risk-aware human model from behavioral economics (Cumulative Prospect Theory) for modeling loss aversion behaviors of humans under risk and uncertainty [78].

Another recent approach for human behavior modeling is the Reward Augmentation and Repair through Explanation (RARE) framework for estimating and improving a collaborators' task understanding. Here, Tabrez et al. provided a computational framework for human reward function estimation via a set of possible Hidden Markov Models (HMMs) [20], representing a task's reward function and partially deficient variants (e.g., missing reward information). The collaborative agent must infer the most likely HMM for explaining the teammates' behavior, which in turn indicates a plausible underlying reward function for explaining the human's actions. For more details on inferring human intent and predicting behavior in human-robot collaboration scenarios, we direct readers to the recent comprehensive survey by Hoffman et al. [17].

2.4.2 Second-order Mental Models

The concept of a second-order mental model is related to a recursive type of reasoning modeled by game theorists ("I believe that you believe that I believe...") which can be extended to a possibly infinite reasoning process [79, 80]. The second-order mental model is one step deeper in behavior modeling (i.e., a robot forming a belief over a human's model of the robot). Second-order mental

models enable robots to possess more predictable and explicable behavior, as the effects of their actions on another agent's perception of them is included in the model.

Work by Huang et al. modeled humans as learning a robot's objective function over time by observing its behavior using Bayesian IRL, an inversion of typical IRL paradigms where a robotic agent attempts to infer human objective functions. To account for noisy learning behavior from humans, the authors utilize approximate-inference models. Using this insight, an agent can plan for actions that communicate to the human so as to be maximally informative, better enabling humans to anticipate what the robot will do in novel situations [81].

Another approach that has shown promise is the Interactive POMDP (I-POMDP) framework, which modifies a traditional single-agent POMDP to include other agents by creating the notion of an interactive state. An interactive state encapsulates both the environment state and the modeled belief state attributed to another agent. Brooks and Szafir use this I-POMDP framework [82] for performing Bayesian inference of second-order mental models. They estimate the human's Q-function (a function that helps determine the optimal action given an interactive state) through IRL and use it to infer the human's belief state about the agent, by comparing it with the human's actions assuming a Boltzmann rational behavior model [49].

2.4.3 Shared Mental Models

Shared mental models enable team members to draw on their own well-structured common knowledge as a basis for selecting actions that are consistent and coordinated with those of their teammates. They are strongly correlated to team performance [40]. In this section we focus on methods employed for establishing a shared understanding between teammates.

One well-known approach in HRI inspired by SMM is work on human-robot cross-training by Nikolaidis and Shah, which focuses on computing a robot policy aligned with human preference by iteratively switching roles (between a human and a robot) to learn a shared plan for a collaborative task [83]. Hadfield-Menell et al. approached SMM as a value alignment problem, ensuring that the agents behave in alignment with human values. They utilize a cooperative inverse reinforcement

learning (CIRL) formulation, where a robot maximizes a human teammate's unknown reward in a cooperative, partial information game. They show that solutions within this formalism result in active teaching and active learning behaviors [84].

Nikolaidis et al. also propose a game-theoretic model of a human's partial adaptation to a robot teammate. This method assumes the robot agent knows a "true" utility function for the team, and the human is following a best-response strategy to the robot action based on their own, possibly incorrect reward function. The robot uses this model to decide optimally between revealing information to the human and choosing the best action given the information that the human currently has [56].

From these well-known models, we can see that establishing a shared mental model requires communication between agents (except the cross-training method, where agents learn each other's responsibilities by switching roles). We can separate these communication strategies into two categories: implicit (e.g., using movement or motion) and explicit (e.g., verbal explanations).

Implicit Communicative Models. A popular principle in motion planning for expressing intention to a collaborator is the notion of legibility. Dragan et al. developed a formalism to mathematically define and distinguish predictability (predicting a trajectory given a known goal) and legibility (predicting a goal given an observed trajectory) of motion based on a rational action assumption for the collaborative human [76]. Kulkarni et al. generate explicable robot behavior by learning a regression model over plan distances and mapping them to a labeling scheme used by a human observer, minimizing divergence between the robot's plan and the plan expected by the human [85].

Another mode of implicit communication is through gesture and non-verbal expression. One example of this is work by Lee et al. which uses a BToM approach to model dyadic storytelling interactions [86]. They propose a method for a robot to influence and infer the mental state of a child while telling it a story, specifically estimating the child's degree of attentiveness towards the robot. They model emotion expression as a joint process of estimating people's beliefs through

inference inversion using a Dynamic Bayesian Network (DBN), and subsequently produce nonverbal expressions (speaker cues) to affect those beliefs (attention state).

Explicit Communicative Models. Model reconciliation processes try to identify and resolve the model differences of a collaborator through explanations, thereby establishing a shared mental model. These processes lead to predictable behavior from the collaborative agent: a consequence of explainability [87, 88, 89]. Briggs and Scheutz's recent work provides a formal framework to correct false or missing beliefs of collaborators in a transparent and human-like manner by using adverbial cues, adhering to Grice's maxims [90] of effective conversational communication (quality, quantity, and relevance) [91]. Additional recent works also address the generation of these explanations, seeking output that is optimal with respect to various quantitative and qualitative criteria including selectivity, contrastiveness, and succinctness [18, 24, 31, 92, 93].

The major contributions of this thesis focus on expanding effective methods for mental model reconciliation by leveraging multimodal communication. This work enables robots and autonomous agents to generate real-time multimodal communication, such as natural language and AR-based visual explanations, within partially observable and complex human-robot collaboration scenarios. Additionally, it posits that different modalities of communication have different utilities depending on the situation and argues for leveraging various modalities, augmented with novel interfaces, to facilitate fluent communication between human-robot teams.

2.5 Evaluation Methods

In this section, we discuss evaluation methods employed in human-robot teaming for each of the desirable traits characterised in Section 2.2.

Team Fluency. Fluency, the metric for well synchronized meshing of joint actions between humans and robots, is difficult to measure and optimize in practice [94]. Hoffman and Breazeal demonstrated that fluency is a distinct construct to efficiency through a user study involving an anticipatory controller (when the robot anticipated participants' actions, task efficiency was not improved, but participants' sense of fluency was increased) [95]. For team fluency, there exist a

number of validated subjective metric scales, as well as commonly used objective measures, such as human and robot idle time, fraction of time spent concurrently working between agents, and delay times between one agent finishing a precursor task and another agent resuming that task [55].

Adaptability. Shared mental models offer a mechanism for adaptability: quick, on the fly strategy adjustments by a team. As adaptability is intrinsically linked to performance, the majority of measures are objective, often treating an adaptable controller as an independent variable to compare alongside other controllers. Specific objective measures vary with the formulation used, including mean reward accrued [56] and similarity metrics between human and robot notions of "correct action sequence" in an evolving task [83]. Though there is a notable lack of validated subjective measures for agent adaptability in HRI, many studies utilize subjective metric scales for correlated measures such as team fluency and trustworthiness [55, 83]. Nikolaidis et al. have additionally showed that accounting for individual differences in humans' willingness to adapt to a robot is positively correlated with trust [48].

Team Trust. Shared mental models promote trust and reliability by alleviating uncertainty in roles, responsibilities, and capabilities while working in a team. Lee and See proposed a three dimensional model wherein trust is influenced by a person's knowledge of what the robot is supposed to do (purpose), how it functions (process), and its performance [96]. Based on previous studies, robot performance is considered to be the most influential factor for trust [97], likely due to the importance of the agent's ability to meet expectations [98]. Other factors with positive relationships to trust are minimizing system fault occurrence, system predictability, and transparency [99]. Most subjective measures for trust in HRI research are newly created to match individual study requirements and lack the rigor in development and validation available in standardized scales from the human factors community. Some well-known standardized scales with high potential for use in HRI to evaluate a user's trust perception of an agent are the HRI Trust Scale, Dyadic Trust Scale (DTS), and Robotic Social Attributes Scale (RoSAS) [99, 100].

Effective Communication. Previous studies show that information exchange and effective communication are important for building trust between team members. These communications

can be verbal (explicit) or nonverbal (implicit), as seen in Section 2.4. For explicit models, the following qualities have been found to be positively correlated with trust and teamwork: task-related communications, contrastive explanations expressing model divergence, and user & context dependent information (such as providing technical information to an expert, and accessible information to a lay-user) [101, 102, 103]. For implicit models, such as those aimed at plan legibility and explicability, self-reported understanding of a robotic agents' behavior or goal is a common evaluation metric. Additionally, subjective metrics are often crafted for individual study requirements, aimed at uncovering related traits like robot trustworthiness [76, 104, 105].

Explainability. Explainability deals with the understanding of the mechanisms by which a robot operates and the ability to explain robots' behavior or underlying logic [20, 92]. Existing works in explainable AI assess the effects of explainability through self-reported understanding of the agent behavior, successful task completions, system faults, task completion time, number of irreparable mistakes, and trust in automation. A survey by Walkotter et al. described three categories of measures for evaluating the effectiveness of explainable architectures (in descending order of importance): 1) Trust (willingness of users to agree with robot decisions through a self-reported scale), 2) Robustness (failure avoidance during the interaction), and 3) Efficiency (how quickly tasks are completed) [106]. Two primary standardized scales for measuring explainability are from Hofman et al. [107] and Silva et al. [108]. Silva et al. created a 30-question survey with items intended to measure the simulatability, transparency, and usability of the agent's explanations. In contrast, Hoffman et al. provided a standardized scales to evaluate the following concepts: (1) the goodness of explanations, (2) whether users are satisfied with explanations, (3) how well users understand the AI systems, (4) how curiosity motivates the search for explanations, (5) whether the user's trust and reliance on the AI are appropriate, and (6) how the human-XAI work system performs.

2.6 Emerging Fields & Discussion

Mental models have proven beneficial for many human-robot teaming applications such as assistive and healthcare robotics [109], social path planning and navigation [110], search and rescue [111], and autonomous driving [78, 112]. In this section, we describe a selection of more recent emerging use cases of mental models in HRI.

Though robots have been fixtures in industrial applications since the 1970s [113], the factory of the future is likely to utilize robots for a much broader range of tasks, and in a much more collaborative manner, enabled in part through the use of recent developments in mental models. Many of these potential robot tasks intrinsically require operation in proximity to humans, raising issues of safety and efficiency. Recent work by Unhelkar et al. provides a framework for human-aware task and motion planning in shared-environment manufacturing [114]. Additional research in this area focuses on the problem of task scheduling for safely and effectively coordinating human and robot agents in resource-constrained environments [19, 115]. Another recent development has been towards the generation of supporting behavior for improving human collaborators' task performance. These supportive behaviors do not directly contribute to a task but instead alleviate the cognitive and kinematic burdens of a collaborating human (e.g., fetching tools or stabilizing objects during assembly) [87, 116].

Furthermore, developments in augmented reality (AR) technology have shown promise for industrial HRI applications. AR represents a novel modality of model communication for human-robot collaboration, wherein details of a robot's plan or decision making process are visualized and presented to a human teammate as holographic imagery overlaid onto the robot itself, viewed through a head-mounted display. Notable work in this area has focused on visually conveying robotic motion intent during human-robot teaming tasks with AR, both for robotic manufacturing arms [117], and mobile robots [118], a technique which has been shown to broadly increase objective measures of task accuracy and efficiency, as well as subjective perceptions of robot transparency and trustworthiness. Recent work has explored the inclusion of human-to-robot communication

features on top of AR visualization, allowing human teammates to diagnose problems with and modify a robot's plans or internal models during collaboration [119, 120, 121].

With the currently observed rate of increase in agents' capability for social behavior and natural language generation, important problems surface regarding robot ethics and norms [122, 123], particularly in cases of policy elicitation (manipulating the human in the hopes of achieving some greater good). These behaviors and capabilities induce perceptions of a moral and social agency in robots similar to human standards of morality [124]. In reality, such actions/behaviors do not embody any maliciousness but rather emerge due to necessity of situation and cooperation. Some major challenges within this domain of problems include establishing moral norms during collaboration, anticipating possible norm violations, attempting to prevent them while executing, and if norms are eventually violated, taking mitigating actions to create transparency and user awareness (such as providing justifiable explanations communicating the robot's decision-making processes or capabilities) [125, 126].

Conclusion: As evidenced by the emerging application areas found within human-robot teaming literature, mental models continue to be developed and applied in novel ways. Research in human-robot interaction is rapidly evolving and expanding into new application areas, so this list is far from exhaustive. In this chapter, we have provided a general overview of mental models as applied to human-robot teaming: formalisms which have proven to be significantly beneficial for fluent collaboration and cooperation between teammates.

The next chapter of the thesis will focus on explainable robotic coaching to achieve mental model alignment via natural language communication. Additionally, it will introduce the concept of behavior manipulation, also known as **policy elicitation**. This refers to a class of problems in human-robot teaming wherein an agent must guide humans towards an optimal policy, or away from potential failure states, to successfully complete a task, either through implicit or explicit communication [18, 20, 127].

Chapter 3

Communication and Explanations for Mental Model Reconciliation via Reward Coaching

"We do not learn from experience... we learn from reflecting on experience."

— John Dewey, *Experience and Education*

This chapter presents methodologies for enabling robots to effectively communicate their decision-making rationale and operationalize robotic coaching using natural language explanations. It is divided into two subchapters. The first presents a novel framework for explainable robotic coaching and justification, aiming to transform robots into competent coaches using explainable AI to establish shared mental models among teammates. The second explores operationalizing robotic coaching and introduces the concept of policy elicitation, where an autonomous agent guides humans towards optimal policies or away from failure states through explicit communication to complete tasks successfully.

3.1 Part 1: Framework for Robot Coaching and Justification

In this subchapter, we present a novel mechanism for enabling an autonomous system to detect model disparity between itself and a human collaborator, infer the source of the disagreement within the model, evaluate potential consequences of this error, and finally, provide human-interpretable feedback to encourage model correction. This process effectively enables a robot to provide a human with a policy update based on perceived model disparity, reducing the likelihood of costly or dangerous failures during joint task execution. This chapter makes two contributions

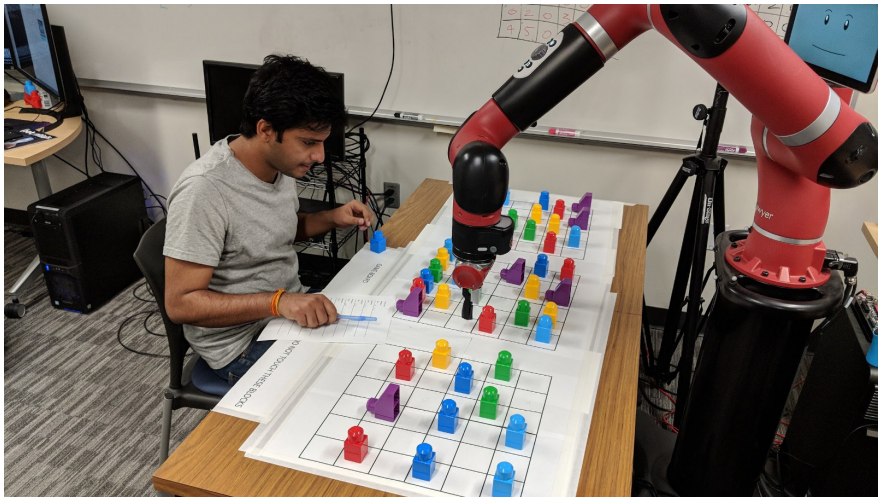


Figure 3.1: A participant plays a collaborative, color-based Sudoku variant with a robot during a human subjects study evaluating the proposed framework. Using RARE, the robot is able to identify, indicate, and explain potential failure modes of the game based on the human's predicted understanding of the game's reward function.

at the intersection of explainable AI (xAI) and human-robot collaboration: 1) The Reward Augmentation and Repair through Explanation (RARE) framework for estimating task understanding and 2) A human subjects study illustrating the effectiveness of reward augmentation-based policy repair in a complex collaborative task.

3.1.1 Introduction

Shared expectations are crucial for fluent and safe teamwork. Establishing a common mental model of a task is essential for human-robot collaboration, where each team member's skills and knowledge may be combined to accomplish more than either could in isolation [128, 129, 130]. However, gaining insight into a collaborator's decision-making process during task execution can be prohibitively difficult, requiring the agent to have the capability to perform policy explanation [92]. Further, taking corrective actions when a team member's comprehension of the task doesn't match your own requires one to not just indicate a problem with the policy, but also to identify the root cause of the incongruousness.

Within society, the roles and responsibilities being assigned to robots have grown increasingly complex, reaching the boundaries of social integration. As this continues, it is reasonable to assume

people will increasingly turn towards robots for completing important collaborative tasks with real consequences of failure, such as search and rescue [131], housekeeping [132], and personal assistance for the elderly [133, 134]. Providing these autonomous systems with the ability to identify and explain potential failures or root causes of sub-optimal behavior during collaboration will be essential to establishing appropriate levels of trust and reliance, while simultaneously improving the task understanding and performance of human operators.

Consider the problem of resource allocation and asset tasking during a collaborative search and rescue operation, where a human operator is commanding a fleet of UAVs. If the human provides a sub-optimal flight plan to an agent that provides poor coverage or exceeds its flight range, a system that could both generate human-interpretable feedback indicating the potential failure mode associated with the human's action and provide a justifying explanation would be far more useful than one that could not. One might expect such a capability to improve both operator task proficiency and failure rates.

To provide usable feedback for avoiding sub-optimal behaviors expected of a collaborator, we introduce a framework that leverages the assumption that sub-optimal collaborator behavior is the result of a misinformed understanding of the task rather than a problem with the collaborator's rationality. In terms of a task defined through a Markov Decision Process, a human's poor action selections should be attributable to a malformed reward function rather than a malformed policy search algorithm. Building on this assumption, we believe a useful autonomous collaborator should be able to 1) infer the most likely reward function used as a basis for a human's behaviors; 2) identify the single most detrimental missing piece of the reward function; and 3) communicate this back to the human as actionable information.

Toward this goal, we propose **Reward Augmentation and Repair through Explanation** (RARE), a novel framework for improving human-robot collaboration through reward coaching. RARE enables a robot to perform policy estimation during a collaborative task and offer corrections to a teammate's mental model during joint task execution. Our model estimates the most likely reward function that explains the collaborator's behavior and provides a repairing explanation

meant to enable the collaborator to update their reward function (task comprehension) and policy (behavior). The two primary contributions of our work are:

- Reward Augmentation and Repair through Explanation (RARE), a novel framework for understanding and correcting an agent's decision-making process, which estimates an agent's understanding of a domain's reward function through their behavior and provides corrective explanations to repair detected issues.
- A human subjects study-based evaluation of RARE, showing both the technical feasibility of the approach alongside empirical results illustrating its effectiveness during a complex human-robot collaboration.

3.1.2 Background and Related Work

Much of the recent work in human-robot collaboration focuses on the common goal of making robots a more acceptable, helpful, trustworthy, and functional part of our day-to-day life. Throughout the established literature on human-robot collaboration, a majority of the attention has been placed on providing capabilities to enable robots to adapt to their human collaborators, as opposed to providing them with the tools needed to improve their human collaborators' behaviors for more productive joint task execution.

One important trend in human-robot collaboration has been to improve robots' awareness of human behavior [46, 135, 136]. These approaches primarily focus on enabling a robot to successfully adapt and perform tasks in the presence of humans rather than enabling them to collaborate on equal footing with people. An effective approach to collaboration has been to enable the robot estimate a human collaborator's belief [137] in order to plan 'in their shoes', allowing for a better understanding of their decision-making process and the factors influencing their choices. Recent work [138] has used Inverse Reinforcement Learning (IRL) [64] to infer human behavior given a known goal. This work assumes the human holds an imperfect dynamics model for the domain, and creates a shared control scheme to invisibly correct the disparity. As our approach attributes

suboptimal behavior to a human's imperfect reward model, we find applicability to scenarios (such as cognitive tasks) where shared control isn't a viable solution. Unfortunately, existing approaches do not provide mechanisms where this perspective-taking can be used to improve a human's performance and awareness on a task — rather, they mainly focus on mechanisms for allowing a robot to adapt to a human. Work by Imai and Kaneko has provided a method to estimate a human's false beliefs about a domain [139], with the intent to allow a robot to dispel said beliefs. Work by Faulkner et al. models human belief to generate minimal communication [140], enabling a robot to effectively ask for help from a human oracle, but does not investigate the reverse scenario of providing succinct help to a human agent. Implicit communication [141, 142] has also been investigated, utilizing a robot's actions to provide actionable signal about its intent in collaborative scenarios.

One popular approach is to develop a "theory of mind" about one's collaborator [44, 45, 46, 47] to effectively understand their knowledge state, goals, and beliefs. Work by Devin and Alami [44] estimates the information the human might be missing to minimize the conveyance of unnecessary information. In work by Leyzberg et al. [26], it is shown that personalized interactions lead to better results, while in [48] trust is better preserved and maintained by performing actions that respect a human's preferences.

During collaboration, interruptions are necessary for effective resynchronization of expectations. A great deal of work has been performed to study how [143] and when [144, 145, 146] an autonomous agent should interrupt a teammate, how to personalize interruptions [147], and even how interruptions can cause more errors in skill-based tasks [148]. **Our work addresses a crucial technical gap as it not only estimates a collaborator's belief about the reward function of their current task, but also infers the root cause for inaccuracies encoded in said belief.** Doing so provides the infrastructure needed for achieving the autonomous repair of a collaborator's policy through explanations generated online during task execution intended to illustrate and eliminate their root cause.

3.1.3 A Framework for Reward Augmentation and Repair through Explanation

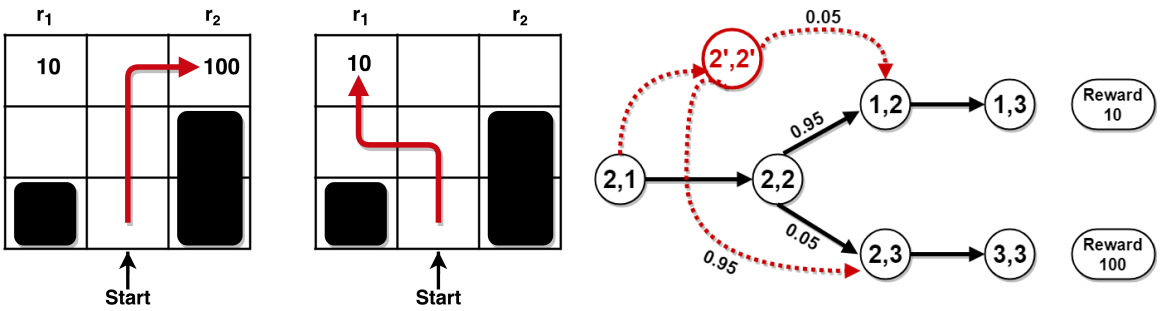
In this section we detail the theoretical framework of **Reward Augmentation and Repair through Explanation** (RARE), wherein we utilize a Partially Observable Markov Decision Process (POMDP) coupled with a family of Hidden Markov Models (HMMs) to infer and correct a collaborator's task understanding during joint task execution. The central insight underpinning the proposed method is that sub-optimal behaviors can be characterized as an incomplete or incorrect belief about the reward function that specifies the task being performed. By proposing potential (erroneous) reward functions and evaluating the behavior of a virtual agent optimizing its policy using these functions, our approach allows a robot to determine potential sources of misunderstanding. Once a plausible reward function is discovered that explains the collaborator's behavior, a repairing explanation can be generated and provided if the benefit of correction outweighs the consequences of ignoring it.

The framework can be characterized through three interconnected components responsible for: 1) estimating a collaborator's comprehension of a domain's reward function; 2) determining a policy for trading-off between collaborative task execution and intervention; and 3) formulating corrective explanations for reward function repair. For the remainder of the section, we focus on the use case where the collaborating agent is a human and the agent employing RARE is a robot jointly executing a task with them.

emphEstimation of Reward Comprehension

The core insight of RARE is that sub-optimal behavior is an indicator of a malformed reward function being used by an otherwise rational actor. Thus, if it is possible to determine which reward function the actor is using, it will be possible to identify problematic misconceptions that may contribute to adverse behavior. As a result of this formulation, RARE necessarily assumes that the agent implementing it has a complete specification of the domain's true reward function.

To determine which components of the reward function the human collaborator is using, RARE utilizes an HMM that incorporates both state features of the world ("world features") and



(a) Task executions given two different comprehensions of a gridworld domain's reward function.

(b) Incorporation of reward comprehension to infer an agent's likely policy. State (2,2) indicates partial knowledge of the domain reward function, while (2',2') indicates full knowledge.

Figure 3.2: An example of two possible comprehensions of a domain's reward function. (a)-left: The agent knows the true reward of the domain. (a)-right: The agent does not know about the +100 reward, and behaves rationally given this malformed reward function. (b): Latent reward comprehension variables differentiating state (2,2) and (2',2') provide a hypothesis to better explain the agent's behavior, distinguishing between the scenarios represented in (a)-left and (a)-right.

latent state features that indicate knowledge of corresponding components of the domain's reward function ("comprehension features"). In the example shown in Figure 3.2, the reward function has two components: a +10 reward for entering the top left cell and a +100 reward for entering the top right cell. The transition probabilities of a given state are directly computed from a policy trained on the reward function specified by the values of the comprehension features in the state.

We define an augmented HMM (RARE-HMM) as the 7-tuple $\lambda = \{S, O, M, \pi, A, B, \tau\}$ that estimates the likelihood of a state-action trajectory of an observed agent given a particular reward function, where:

- $S = s_0, s_1, ..., s_N$ is the finite set of states the observed agent can be in.
- $O = o_0, o_1, ..., o_M$ is the finite set of possible observations, which correspond to the effects of the action most immediately taken by the observed agent.
- M is a Markov Decision Process (S, A, T, R) where S is the set of states in the MDP, A is the set of actions an agent may take $(A = O)$, T is a stochastic transition function describing the action-based state transition dynamics of the model, and R is a reward

function. Intuitively, M serves as a simulator for an agent in the task domain.

- π is a policy trained to maximize reward in M.

- A is a stochastic transition matrix, indicating the transition probability from state i to state j:

  $A_{i,j} = P(q_t = s_j | q_{t-1} = s_i)$, where $0 \leq i, j \leq N-1$, q_t is the state at time t, and $\forall i \in [0, N], \sum_{j=0}^{N} A_{i,j} = 1$. These probabilities are drawn directly from the composition of the transition dynamics function M_T and π. In other words, A represents the transition likelihoods for an agent following policy π in M.

- B is the stochastic observation emission matrix, indicating the probability of getting observation j at time t in state i: $B_{i,j} = P(v_t = o_j | q_t = s_i)$, where $0 \leq i \leq N$, $0 \leq j \leq M$, and v_t is the observation emitted at time t. $\forall i \in [0, N], \sum_{j=0}^{M} B_{i,j} = 1$.

- τ is the distribution describing the probability of starting in a particular state $s \in S$ such that $\sum_{i=0}^{N} \tau_i = 1$.

Specifically, RARE utilizes a set of such HMMs Λ, where each member $\lambda \in \Lambda$ uses a unique reward function.

Collaborative Task Execution and Reward Repair.

For a given collaborative task, we define the RARE agent's behaviors with a policy that solves an augmented POMDP (RARE-POMDP) defined by the 6 tuple: $(S, A, T, R, \Omega, \mathcal{O})$ where:

- S is the set of world states, consisting of both traditional features W ("world features") and additional latent features C indicating the collaborator's understanding of the domain's reward function ("comprehension features"). We formulate the set of comprehension features as a vector of boolean variables indicating whether a particular component of the reward function is known by the collaborator.

$$S = \begin{bmatrix} W \\ \_ \\ C \end{bmatrix}, W = \begin{bmatrix} x \\ y \\ \vdots \end{bmatrix} C = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}$$

- A is the set of actions, consisting of both task-specific physical actions and reward repair-specific social actions.
- T is a transition function specifying state transitions as a function of action performed. As RARE models a collaborative process, the dynamics introduced by the collaborator's actions are also represented within this function, but are assumed to be known given known comprehension features (i.e., if the agent's reward and policy are assumed to be known, its behavior in a given state is also known).
- R is a reward function specifying the value of executing an action in a given state.
- Ω is the set of all possible observations. In a RARE-POMDP, each observation corresponds to a particular RARE-HMM being the most likely explanation for a collaborator's behavior, signaling the current state of their reward comprehension (i.e., their understanding of the reward function).
- $\mathcal{O}$ is a function describing observation emission probabilities for a given state. In RARE, the emission function must be designed to encourage congruence between a state's comprehension features and the RARE-HMM with the corresponding reward function in Ω. In other words, a RARE-HMM has higher likelihood if its reward function contains the components indicated by the current state's comprehension features.

The observation emission function presents an important design decision for implementing a RARE-POMDP in a given domain. This function provides a link between the RARE-HMMs, each representing an agent's expected behavior given a particular understanding of a reward function, and the RARE-POMDP that is being solved to maximize the success of the collaboration. In

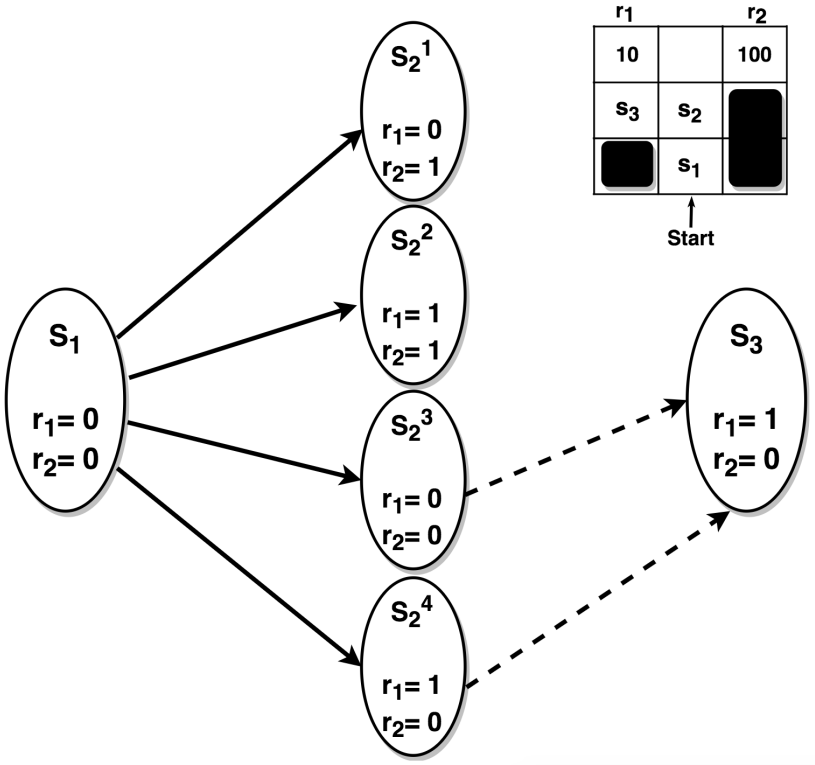


Figure 3.3: Partial visualization of comprehension features for a gridworld domain with two reward factors, one at each terminal reward state. Four variants of s_2 are shown, each indicating a different level of reward function awareness. Observing an agent transition from state s_2 to s_3 provides evidence suggesting they may not know about the larger reward r_2 in the top-right, but do know about reward r_1.

this work, we propose a softmax scoring function based on the likelihood of the collaborator's action sequence for each potential RARE-HMM. For a given observed collaborator trajectory T, RARE-HMM/observation $o_i \in \Omega$ and state $s \in S$, we propose $\mathcal{O}$ such that:

$$P(o_i|s) = \frac{\exp(P(T|o_i))}{\sum_{j=0}^{|\Omega|} \exp(P(T|o_j))}$$

Intuitively, this choice of $\mathcal{O}$ enforces that the RARE-POMDP's estimate for which reward function the collaborator is following is proportional to the likelihood that their behavior was informed by a policy derived from it. In applications where there is not a 1-to-1 correspondence between available RARE-HMMs and potential reward functions (i.e., there are not 2^n RARE-HMMs defined for n reward function components), a more clever approach may be merited.

The RARE-POMDP introduces the opportunity for the agent to make the decision to execute social actions aimed at better informing a collaborator about the domain's reward function. In other words, the agent may execute a communicative action to explicitly inform a collaborator about part of the reward function, directly changing the value of a latent comprehension feature (e.g., the knowledge of r_2's existence in Figure 3.3). Even though such an action may not directly advance the task toward completion, it may invariably result in higher net reward, as it can improve the collaborator's policy by informing them of high reward states or harshly penalized states that may lead to task failure.

Explanation Generation

The RARE framework allows an agent to estimate a collaborator's reward function during joint task execution. This is a powerful piece of information, but it is far more useful in a collaborative context when paired with actions that enable one to augment a collaborator's understanding of the task. RARE uses this information to decide what and when to communicate, updating the collaborator's reward function and policy. For our application domain, we propose an algorithm (Algorithm 1) that autonomously produces statements capable of targeted manipulation of a collaborator's comprehension features based on anticipated task failures. Future work may provide similar algorithms for providing information about non-terminal state rewards or for more generally suggesting collaborator reward function updates.

Intuitively, Algorithm 1 performs a forward rollout of a policy trained on the estimated human reward function, which may contain a subset of the information (factors) of the true reward function known to the RARE agent. As in Figure 3.3, the collaborator may only know of r_1, so we say it is missing the reward factor r_2. Upon completing this rollout, we also run forward rollouts for policies trained on reward functions that include one more reward factor than the human's (Figure 3.2). This step allows the RARE agent to find the most valuable single reward update to provide the collaborator, updating their policy by changing one reward factor at a time, following an iterative interaction pattern previously validated within HRI [149]. Finally, the update is serialized using

Algorithm 1: Augment Terminal-State Reward Comprehension

Input: Factored Reward Function R, Set of Policies Π Trained on Power Set of R, Estimated Human Reward Function R_h, Domain MDP $M = (S, A, T)$, Current state s_c

Output: Semantic Reward Correction

$r_c \leftarrow 0$; // Cumulative reward
$s' \leftarrow \emptyset$;
// Simulate existing human policy
$\pi_h \leftarrow$policy trained on R_h;
while s **is not terminal do**
    // Perform forward rollout of π_h
    $s' \leftarrow M_T(s, \pi_h(s))$;
    $r_c \leftarrow r_c + R(s, \pi_h(s), s')$;
    $s \leftarrow s'$;
$s_{h,terminal} \leftarrow s$; // Terminal state of human policy
$r_h \leftarrow r_c$;
// Find best single-comprehension-change
$\Pi_1 \leftarrow \{\pi \in \Pi \mid \pi$ trained on $R_1 \in R$ s.t. R_1 contains 1 additional factor of R^* than $R_h.\}$;
$\pi_c \leftarrow \emptyset$;
$r_\pi \leftarrow r_h$;
for $\pi \in \Pi_1$ **do**
    $s \leftarrow s_c$;
    $r_c \leftarrow 0$;
    while s **is not terminal do**
        // Perform forward rollout of π
        $s' \leftarrow M_T(s, \pi(s))$;
        $r_c \leftarrow r_c + R(s, \pi(s), s')$;
        $s \leftarrow s'$;
    if $r_c > r_\pi$ **then** $r_c \leftarrow r_\pi, \pi_c \leftarrow \pi$;
feedback $\leftarrow$ "If you perform {describe_action(π_h)}, you will fail the task in state {describe_state($s_{h,terminal}$)} because of {describe_reward(diff(R_h, R_π))}";
return feedback

designer-specified action [150], state [92], and reward factor description functions.

3.1.4 Experimental Validation

To quantify the viability and effectiveness of RARE within a live human-robot collaboration, we conducted a user study wherein participants had to solve a complex collaborative puzzle game – a color-based variant of Sudoku – collaboratively with a Rethink Robotics Sawyer manufacturing robot. In the sections that follow, we present results characterizing participants' perception

of a RARE-enabled robot that provides guidance during complex collaborations to prevent task failure. Failure prevention was attempted by the robot by means of verbal interruptions taking place between the human's selection of a color to play and the human's placement of that color. Additionally, we investigate the role that justification plays when providing advice that directly alters the collaborator's understanding of the game.

Participants were recruited into one of two treatments that determined what the robot would communicate when interrupting a human who is about to play a move that leads to failure: a failure identification-only condition ('control') where future failures are identified but not explained, and an experimental condition ('justification') where future failures are both identified and explained to the collaborator. Participants were assigned to a third, implicit baseline condition ('no interruption') when no failures were detected and the robot did not interrupt the game.

Hypotheses We conducted a human-subjects study to investigate the following hypotheses regarding RARE's application within a live human-robot collaborative puzzle-solving task:

- **H1**: Participants will find the robot more helpful and useful when it explains why a failure may occur (i.e., participants in the 'justification' condition will find the robot to be more helpful than in 'no interruption' condition and control condition.
- **H2**: Participants will find the robot to be more intelligent when it gives justifications for its actions as compared to the other conditions.
- **H3**: Participants will find the robot more sociable when it provides justifications for its failure mitigation than when it doesn't.

Experiment Design

To evaluate our hypotheses, we conducted a between-subjects user study using a color-based collaborative Sudoku variant played on a table with a grid overlay using colored toy blocks. Study participants were assigned into one of three conditions:

- **Control**: The robot interrupts users that are about to make erroneous block placements, indicating to them that it will cause task failure.
- **Justification**: The robot interrupts users about to make erroneous block placements, indicating that it will cause task failure and explaining which game constraint will inevitably be violated.
- **No Interruption**: An implicit condition for participants that do not commit any errors and experience interruptions by the robot.

During the game, participants place blocks concurrently with the robot (i.e., without turn-taking), until the board is filled. Participants were required to place blocks successively in the grid cells most proximal to themselves, enforcing that the final row for both human and robot were adjacent (the middle of the board). As in Sudoku, certain blocks were pre-placed on the board to limit the solution space of the task.

The robot was pre-trained on all possible solutions for the game board, making it an expert on the task. Human participants were not exposed to the board before beginning the task, and as such could be considered novices trying to solve the game online — making them susceptible to errors. During gameplay, the robot is able to verbally interrupt the human player before they place a block that will make the game impossible to solve, with the opportunity to provide feedback that may avoid task failure.

Rules of the Game

Participants must collaboratively solve a color-based 6x6 cell Sudoku variant (Figure 3.4), by placing colored blocks on the table until the grid is filled. There were six unique colors of block available, with a large supply of all colors available to each player. Both the participant and robot were required to place blocks from right to left, nearest-row to farthest-row, enforcing the constraint that the middle of the board is filled last (where the need for coordination is maximized). The game has two major constraints (Figure 3.5) limiting the gameplay decisions of both the robot and

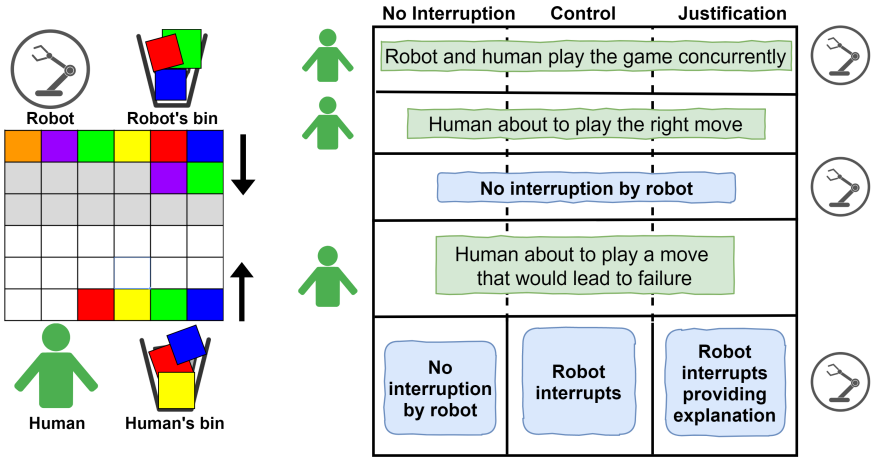


Figure 3.4: (Left) Board layout for the collaborative color-based Sudoku variant. Each player concurrently fills in the three rows closest to them with colored blocks, respecting the game's constraints. The adjacency of each player's final row introduces non-trivial coordination requirements. (Right) Diagram of game flow across the three experimental conditions.

the participant.

- Row Constraint: The first constraint restricts each row of the game board to have only one of each color type.
- Adjacency Constraint: The second constraint requires that no block may have a neighbor (assuming an 8-connected grid) of the same color.

The robot and participant solve the game concurrently and independently of each other's pacing. We enforced the restriction that players must solve the row closest to them in full before moving on to successive rows, as this introduces complex coordination requirements early in the game, as early decisions will have non-obvious effects on allowable middle-row configurations. In other words, blocks placed by the robot in its third row will invariably restrict the gameplay of the participant and vice versa. Per the design of the study, the robot analyzes the gameplay decisions of the participant online and generates an interruption should they make a move that violates the constraints or inhibit successful game completion.

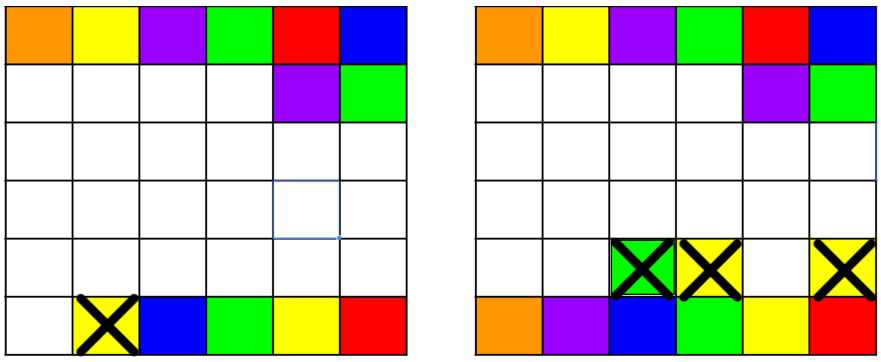


Figure 3.5: Two types of violation that can occur during gameplay. Left: All colors within a row must be unique. Right: No color can be next to itself.

Study Protocol

Before the start of the experiment, informed consent was obtained and participants were educated about the rules of the game. We administered 1-move test puzzles, illustrating specific scenarios possible within the Sudoku variant, to verify their understanding of the game's rules and various constraints.

Both participants and the robot were both free to place blocks on the board as quickly as they were able. To play a move in the game, participants were required to: 1) Move a block from their block supply (the left-most grid of blocks in Figure 3.1) to a staging area (the white area directly in front of participant); 2) Solve a distractor task; and 3) Either place the staged block onto the game board or return it to the block supply and return to step 1. The staging dynamic was implemented to provide the robot with a brief moment within which it could interrupt the participant should their choice of block be determined to cause an inevitable task failure. We utilize multiplication problems as distractor tasks, though the correctness of the participant's answers was not verified.

Any blocks placed on the game board were considered final and could not be changed. If the human placed a block that prevented the game from being completed, the robot would halt the game by saying, *"I am sorry, the game cannot be solved now."* Otherwise, gameplay continued until the human and robot both solved their respective sides of the board.

At the conclusion of the game, participants were lead away from the game board to com-

plete the a post-experiment survey and exit interview. Following the experiment, a comprehensive analysis of the dependent variables using objective measures (e.g., task completion time, idle time and number interruption) and subjective measures (e.g., Likert scale, open-ended survey questions) were used to assess the overall effectiveness of the proposed approach.

Implementation

Sawyer picked blocks from its supply and placed them on the board according to the game's rules. Concurrently, the robot controller implemented RARE, which monitored the current board state and human's actions, occasionally performing verbal interruptions according to the condition being run. For this game, we abstracted reward into three classes for comprehension variables: row constraint, adjacency constraint, and victory. Human-understandable feedback was generating using these with Algorithm 1. To make the game solvable quickly, we used an algorithmically predetermined board configuration to minimize the reachable states, accelerating exploration of the solution space.

Measurement

Our IRB-approved study was completed by 26 participants recruited from a university population. Participants' reported gender skewed male (65% male), and ranged in age from 18 to 30 (M = 21.87; SD = 2.93). All participants came from STEM backgrounds, and their familiarity with robots was relatively high (M=5.08, SD=1.28) on a scale from 1 to 7.

An exit-questionnaire was administered to participants after the conclusion of the game. The questionnaire was developed using questions derived from established collaborative robotics questionnaires [151, 152]. Participants were asked to rate their opinion and experience with Sawyer in 7-point Likert-scale items. Three concepts were identified which form the basis of our hypotheses, based on the previous study of shared autonomy and mixed observability of human and the agent: *Helpfulness, Sociability and Intelligence.* To determine these concepts, we first extracted the latent factors using principal component analysis (PCA). The identified factors were reduced to 11 using

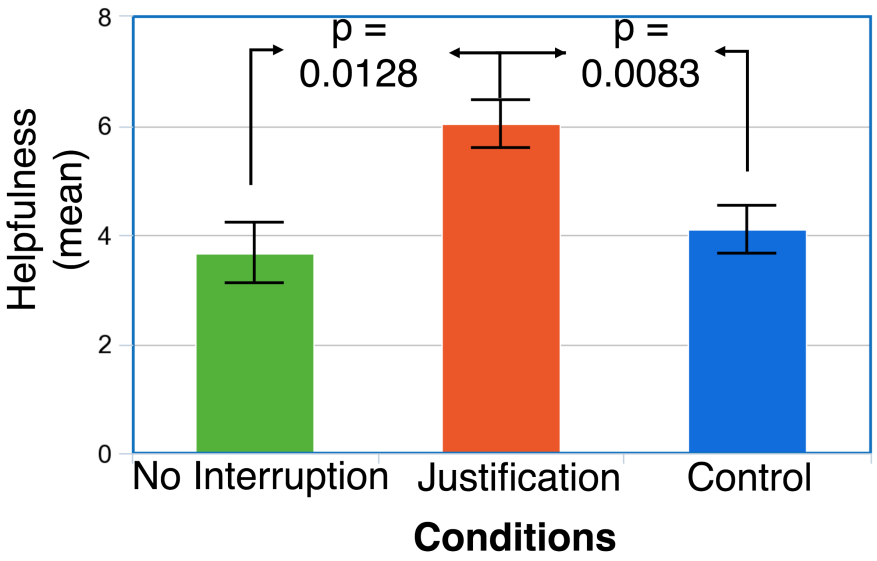


Figure 3.6: Mean ratings of Helpfulness across three experimental conditions. Tukey's HSD test shows a statistically significant difference between all three conditions.

the Kaiser criteria, selecting factors with eigenvalues greater than 1. To spread variability more evenly across each factor, we calculate the loadings of each variable on each factor and applied varimax rotation. To identify the items that can be combined to construct a valid scale, we applied a cutoff point of correlation $r > 0.6$ to the factor matrix.

Sociability was comprised of questions measuring participants' opinions about Sawyer with respect to the interaction's naturalness, pleasantness, and comfort ($\alpha = 0.8557$).

Helpfulness was comprised of questions measuring participants' opinions about how useful and informative Sawyer was during the interaction and its ability to help ($\alpha = 0.83$).

Intelligence was comprised of questions measuring participants' opinions about how intelligent and knowledgeable Sawyer was ($\alpha = 0.8734$).

3.1.5 Results and Discussion

Analysis

There were no anomalies or outliers detected in our combined data set for any of the three concepts,

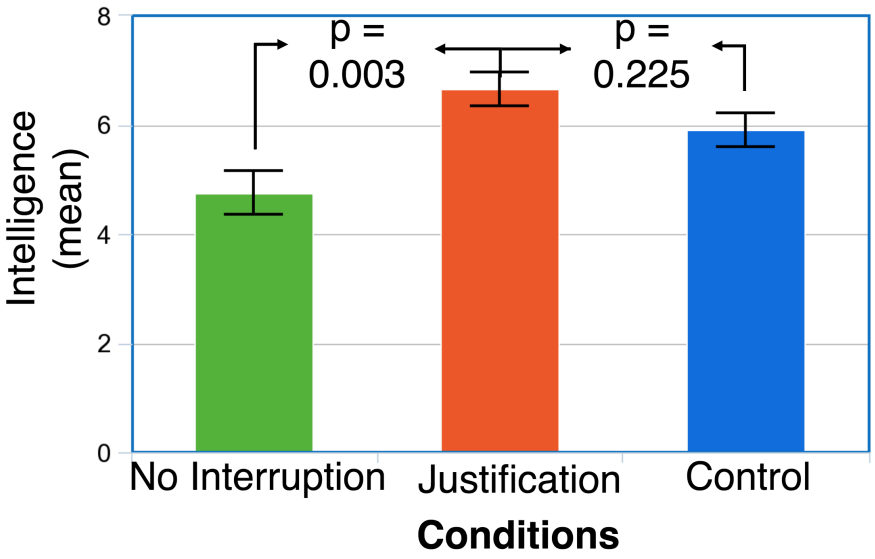


Figure 3.7: Mean ratings of Intelligence across three experimental conditions. Tukey's HSD test results show a significant effect between the justification condition and the no-interruption conditions, but not between no-interruption and control.

but the datasets were positively skewed. We did not observe any multimodalities in the distribution of data. We conducted an ANOVA to test effects across our experimental conditions with respect to perceptions of *Sociability, Helpfulness, and Intelligence.*

We found a significant effect from the 'justification' condition on perceived helpfulness ($F(2, 23) = 7.23, p < 0.004$), **confirming H1**. Post-hoc comparisons using Tukey's HSD test (Figure 3.6) revealed that the justification condition resulted in a significantly different level of Helpfulness as compared to the control condition ($p < 0.009$) and the no-interruption condition ($p = 0.013$).

We also found a significant effect caused by the justification condition on our measure for intelligence ($F(2, 23) = 6.99, p < 0.005$), **confirming H2**. Post-hoc comparisons using Tukey's HSD test (Figure 3.7) revealed that the justification condition resulted in a significantly different level of our perceived intelligence measure as compared with the no-interruption condition ($p = 0.003$), but not with the control condition ($p = 0.225$). Hence, we cannot dismiss the null hypothesis that a robot notifying a collaborator of a bad action choice may not be differently perceived if it also offers justification for its advice.

No significant effects were found with respect to perceptions of sociability as a function of

experimental condition ($p = 0.1$), thus **we cannot validate H3**.

Objectively, we observed that there were more terminations of the game during the control condition as compared to the justification condition (8/10 vs 2/10) which we did not anticipate when designing our experiment. As the robot preempts human actions that would lead to task failure in both conditions, we anticipated that the our control condition (notification of inevitable failure without justification) might lead to longer completion times. To understand the behavior of participants who ignored the robot's warning, we looked to the open-answer questions in our exit survey.

One of the two participants that had their game terminated due to invalid block placement in the justification condition indicated that they were too involved in the game and did not listen to Sawyer's advice and warnings:

"I was so much involved in completing the game, I completely missed [the warning] from the robot — I just heard some sound from the robot and did not realize what it was saying..."

The other participant indicated that they started to think of Sawyer as a competitor and did not listen to its advice, despite being briefed on the collaborative nature of the game at the onset of the experiment:

"As soon as the game began, I forgot it was a collaborative game and I became competitive and was not sure of advice given by Sawyer"

In the control condition, the survey responses painted a clear picture for the terminations — **participants did not trust Sawyer when it indicated that the human was about to make a move that would cause the task to eventually fail, when it did so without further explanation**. They were confused why the move was not valid, even though it looked valid to them. They were skeptical with respect to Sawyer who was not providing accompanying justification for its judgment of their move, as evidenced by the following quotes from participants' survey responses:

- *"Sawyer wasn't forceful enough and was not giving me the reasons why the move was wrong.*

So I couldn't trust him"

- *"Response looked like hard coded and I did not find the reason to think that Sawyer was addressing to me"*
- *"I felt that Sawyer was a robot that is good but I didn't know what his purpose was ... I feel he should have been more forceful in stopping me doing the wrong moves."*
- *"I did not believe it as it did not give details regarding the wrong step"*

We also found evidence in the post-experiment surveys supporting the notion that **providing justification alongside reward feedback leads to a more positive user experience**. Many participants found easier to trust Sawyer when it was providing an explanation alongside its advice. We also saw evidence that the behavior in the justification condition was affecting the way participants played, an important result.

- *"He was forward predicting the movement of the game and telling me why my move was not right even though it was the right move. I was able to trust him easily when he gave the reasons"*
- *"It helped me make sure that I made the correct decisions"*
- *"I learnt to think of moves ahead when Sawyer helped me once with the game."*
- *"Sawyer's input made me question my understanding of the game"*

Thus, we can conclude from the qualitative and quantitative results of our user study that RARE provides tangible subjective and objective benefits during human-robot collaboration. Our experimental results further show improvements beyond standard failure mitigation techniques. Our results highlight that justification is an important requirement for a robot's corrective explanation. Hence, we validate that our contribution is not a solution in search of a problem, but addresses an important, underexplored capability gap in the HRI and Explainable AI literature.

Opportunities for Future Work

Our proposed framework allows an agent to estimate and provide corrections to a collaborator‘s reward function during joint task execution. RARE's effectiveness stems from its ability to discover the root cause for an agent's suboptimal behavior and provide targeted, interpretable feedback to address it. One of the drawbacks of RARE is that the formulation of reward factors by way of comprehension features causes the state space to explode combinatorially, with non-trivial reward functions causing RARE to easily become intractable.

There are many potential approaches for addressing this problem of scalability: 1) Attention mechanisms and priors to reduce comprehension features (i.e., making a priori assumptions about what one's collaborator knows); 2) State abstractions to reduce state space [153]; and 3) Reward function abstractions (i.e., removing the naive independence assumption of rewards across states), approximations/simplifications, or using a subset of potential reward function candidates.

Furthermore, in our implementation the RARE framework estimates only **missing rewards** from the user‘s comprehension of a domain's true reward function. We are not considering the cases where the user has an imagined reward not truly present in the true reward function, or in other words, where the user erroneously includes incorrect or non-existent reward signal in their comprehension of the domain.

Based on the exit interviews of participants who ignored the robot's advice due to over-engagement in the game, where participants said they were too busy to listen, a promising direction for future work also includes investigating different modalities for conveying reward repair information (e.g., incorporating nudging theory for non-invasive corrections).

Finally, we have considered only a single RARE agent (expert) and a collaborator (novice). Natural extensions of this work include relaxing assumptions about the RARE agent's knowledge of the true reward function (e.g., can RARE be improved to enable two RARE agents with complementary reward functions learn a stronger joint reward function from each other's feedback) or extending the work to larger teams.

3.1.6 Conclusion

In this subchapter, we proposed Reward Augmentation and Repair through Explanation, a novel framework for estimating and improving a collaborator's task comprehension and execution. By characterizing the problem of suboptimal performance as evidence of a malformed reward function, we introduce mechanisms to both detect the root cause of the suboptimal behavior and provide feedback to the agent to repair their decision-making process. We conducted a user study to investigate the effectiveness of RARE over a standard failure mitigation strategy, finding that **RARE agents produce more successful collaborations and are perceived as more helpful, trustworthy, and as a more positive overall experience.**

3.2 Part 2: Policy Elicitation via Semantic Reward Coaching

This subchapter builds upon the RARE framework introduced previously, addressing its limitations and presenting a nuanced approach to generating human-centric explanations (i.e., making it easier for novice users to understand and therefore coach) and enabling better mental model alignment in robotic coaching. RARE corrects a single instance of suboptimal human action at a time, which can be tedious and time-consuming for human collaboration. Furthermore, RARE does not consider the recipient's world model, leading to the generation of uninterpretable explanations. This leads to two critical issues:

- **Scalability Concerns**: Explanation generation for reward descriptors within RARE is exponentially dependent on the state space and domain size, making it less feasible for larger, more complex systems [20, 92].

- **Lack of User-centric Explanation**: The explanations provided by RARE are agnostic to user preferences and understanding levels. To effectively reconcile the mental models of humans and robots, explanations need to be more tailored and nuanced. A user-centric approach to explanations should have two primary characteristics: firstly, it should identify and address the delta – the differences and discrepancies – between the human model and the robot model, understanding where the misunderstanding or misalignment lies [18, 101]; secondly, the explanations should be crafted with an understanding of the human user's existing knowledge and mental model, making the explanations more intuitive and easier for the user [30, 31, 154].

Therefore, this subchapter focuses on human policy coaching, a methodology that can also be extended to agent-to-agent policy coaching. This approach is not only more scalable but also enables the planner to generate user-centric explanations. Here, we present **S**ingle-shot **P**olicy **E**licitation for **A**ugmenting **R**ewards (SPEAR), a novel sequential optimization algorithm that uses semantic explanations derived from combinations of planning predicates to augment human agents' reward

functions, driving their policies to exhibit more optimal behavior by modeling humans as RL agents and reconciling disparities in their reward function.

3.2.1 Motivation & Background

Autonomous systems have been shown to improve human performance across a multitude of tasks by imparting useful knowledge or motivating positive behavioral changes [26, 155, 156]. As is the case in the domains of human-AI tutoring and coaching, this is accomplished primarily using natural language explanations [20, 157]. Unfortunately, the process of generating concise and informative explanations capable of eliciting desired changes is a difficult task, as it requires both insights into a human collaborator's decision-making process and the ability to determine and convey important information [12, 22, 158]. Similarly, for autonomous robots operating with different state representations (e.g., different embodiments or sensors), policy repair requires a common ground (language) to communicate updates for efficient behavior modification during the task. Despite their generalizability, large language models currently fall short in planning and reasoning capabilities and lack real-world grounding, which limits their ability to provide reliable and factual explanations in high-stakes scenarios [159, 160].

In this work, we characterize the problem of semantically manipulating human behavior through external advice, especially when a human teammate displays sub-optimal behaviors. We model the human agent as a reinforcement learner, whose sub-optimality is attributed to a misguided reward function rather than a faulty policy search algorithm [20]. Our proposed solution empowers autonomous agents to: 1) Infer the reward function driving human agent's behavior; 2) Identify divergences between human reward function and their own; and 3) Offer advice that concisely and efficiently rectifies these differences, enabling human policy repair.

An illustrative scenario we use to demonstrate the importance and practicality of this work is routing people during an emergency building evacuation, where inhabitants are unaware of the precise nature of the emergency (Fig. 3.8-right). Even if people are capable of navigating towards the nearest exit, uncertainty about the nature of environmental hazards could be disastrous. Adding

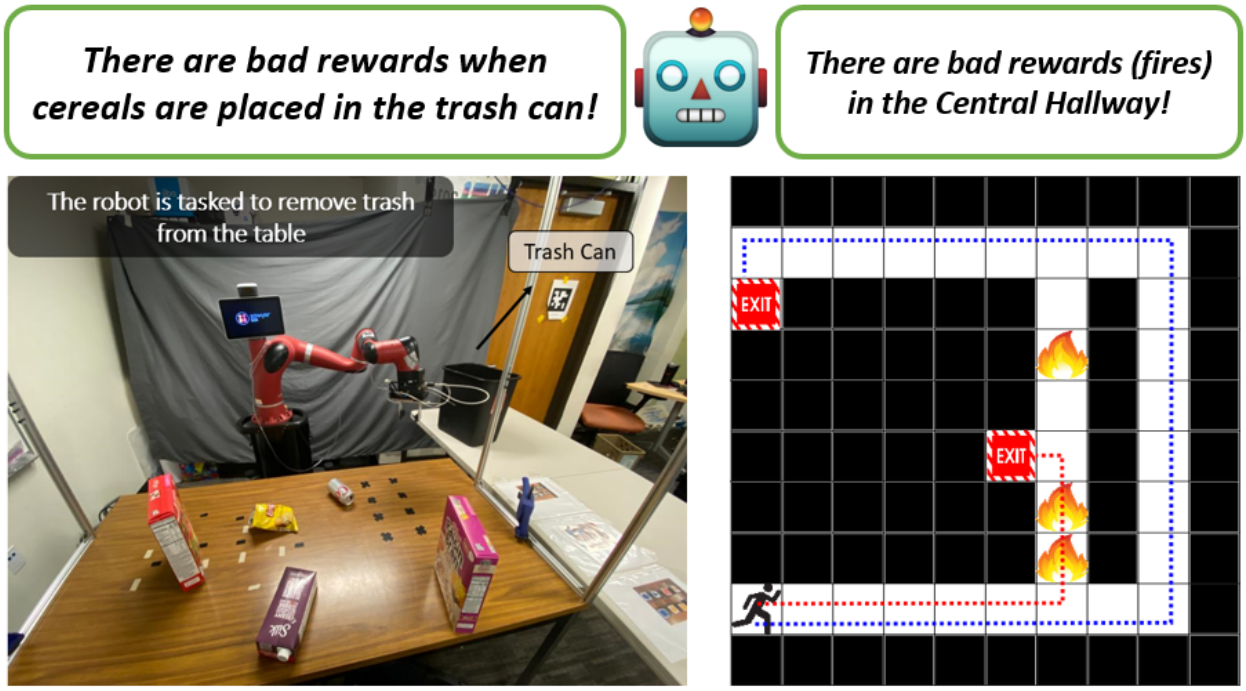


Figure 3.8: (Left) The robot is cleaning a tabletop, but the human observer mistakenly thinks all objects will be thrown out, requiring clarification. (Right) A human tries to exit a building during a fire, unaware of hazard locations. With SPEAR, these policies can be repaired (or justified), using natural language updates to produce more optimal behavior.

to the complexity of time-critical, high-consequence scenarios like this one, it is essential that any advice or instructions account for the recipient's knowledge of the world, and therefore must also consider tradeoffs between accuracy, specificity, and interpretability [161, 162]. As an example, someone visiting a building for a meeting may not know how to change their evacuation plan when told "**There's a fire near Conference Room 3,**" but may be able to adapt their plan if told "**the north half of the building is on fire.**" Even though the latter phrase may not communicate the most accurate representation of the hazard, it is more easily comprehensible to someone who is less familiar with the building. Thus, autonomous systems aiming to offer useful feedback must explicitly consider the complexity of their own explanations and the knowledge held by those they attempt to help.

To achieve this, we present **S**ingle-shot **P**olicy **E**licitation for **A**ugmenting **R**ewards (SPEAR), a novel integer programming-based algorithm for generating reward updates in the form of natural

language advice to improve the policies of human collaborators. SPEAR enables an autonomous robot to utilize knowledge about the beliefs and goals of its collaborators (whether they are humans or other robotic agents) to identify inaccuracies in their models, generating targeted, interpretable guidance for updating their reward functions (and thus, policies) during a task. Key to its effectiveness, SPEAR generates feedback with levels of specificity appropriate to the human agent's understanding of the world. The four primary contributions of our work are:

- A characterization of the **policy elicitation** problem domain.
- SPEAR, a novel algorithm for improving task performance through semantic elicitation of others' policies.
- An integer program enabling semantic communication of state space regions that scales linearly with predicate count (an improvement over exponential, exact methods [163]).
- An experimental validation and performance analysis of SPEAR, and a human-subjects study to validate the utility of the explanations generated by our algorithm.

3.2.2 Background and Related Work

Human-Centered Explainable AI. As autonomous systems become increasingly capable decision makers, xAI has emerged as a necessary component for fielding safe autonomous systems. Explainable AI can help bridge the gap between human and autonomous agents by making complex models more understandable, allowing for faster debugging and failure recovery, ultimately improving transparency, trust, and team performance [6, 7, 8]. Research in xAI has primarily targeted algorithm transparency for developers, aiding in model debugging and behavior prediction [7, 29]. Though these approaches are beneficial for experts, such methods might restrict end users who engage with these models and directly experience the consequences of failure [3, 30].

A popular approach is to use post-hoc explanation methods on RL and/or neural network controllers to enhance interpretability, enabling both developers and novices to understand and

debug models during system failures, as well as assist in decision-making [22, 164]. Some examples include generating flowcharts or recipe-like instructions for users [165], employing decision tree-based RL models [166, 167], providing case-based explanations for visualizing class boundaries [168], and using counterfactual explanations to infer a causal link between input and output models [169].

Others have explored the generation of different types of explanations based on user preference [101, 170]. Work by Briggs and Scheutz explored adverbial cues informed by Grice's maxims of effective conversational communication (quality, quantity, and relation) [90] to transparently track and update mental models of collaborators [91]. Others have leveraged abstraction in explanations [22], allowing for simplified and more useful explanations when an agent's decision-making model is too complex for the observer to comprehend. Our proposed approach generates explanations using overstatement or understatement to abstract detail when necessary, enabling agents to provide helpful feedback even with limited common language.

Explanations for Model Reconciliation. Previous research has demonstrated that explanations bring transparency and also play a functional role in synchronizing expectations during misalignments between human and robot agents [12, 22, 171]. Moreover, people tend to trust autonomous agents more when they have a clear understanding of the robot's capabilities and decision-making process [8]. An effective approach for establishing shared mental models in human-robot collaboration has been to use natural language to explain robots' behavior or underlying logic [20, 163]. Hayes and Shah [163] approached the problem of state region description as a set cover problem, trying to find the smallest logical expression of predicates that succinctly describe a target state region. While original, their method's exponential memory and runtime relative to domain and predicate size limits its real-world applicability. This method was further adapted to multiagent RL environments for policy summarization and query-based explanations [172]. Our approach builds on these formulations and enables real-time applicability in most real-world problems.

Furthermore, though these techniques focus on improving an agent's transparency and behavior using explanation, they don't account for collaborators' level of knowledge or need for the

information. Recent research has leveraged value of information (VOI) to determine when and what information to communicate during collaborative decision-making [173, 174]. Luebbers et al.'s framework, grounded in VOI theory, allows robots to strategically time justifications during periods of misaligned expectations for greater effect. This approach improves performance, assists users in making informed decisions, and promotes higher interpretability [175].

In this work we focus on **policy elicitation**, a process through which feedback is crafted and given to another agent, in the form of reward function updates during task execution, such that they change their behavior to match the desired policy. By providing reward information for targeted regions of state space through explanations (symbolic updates), we can modify a collaborator's reward function using semantic descriptions.

3.2.3 Policy Elicitation via Social Manipulation

The goal of policy elicitation is to cause a behavior change (policy update) in another agent through some form of communicative act. To effectively collaborate with others and coach them towards more optimal policies, it is essential that these communications are intelligible [20, 176, 177], but directly communicating states (i.e. the feature vector itself) relies on there being an efficient mechanism to communicate that information quickly. These criteria are unlikely to hold with humans or heterogeneous robotic agents (not all agents will have the same state representations) as the intended recipients. Our work generates state space-agnostic natural language descriptions of state regions and corresponding reward information (i.e., abstracting the low-level state space), allowing agents to update their reward functions (and policies).

Planning Predicates. We define a **base predicate** to be a pre-defined boolean state classifier (as found in traditional STRIPS planning [178]) with associated string explanation (e.g., in_central_hallway(x) $\rightarrow$ "X is in the central hallway."). To represent intersections of predicates (e.g., "in the central hallway" AND "has fire extinguisher"), we introduce **composite predicates**, which consist of multiple base predicates and evaluate to true if and only if all base predicate members evaluate to true. Predicates may also have a cost associated with them (e.g., how long

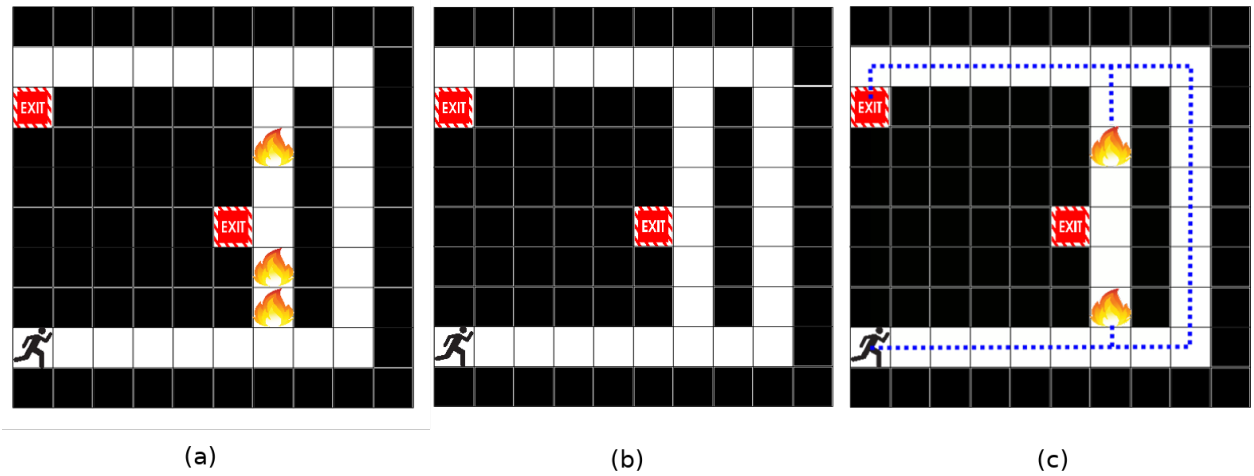


Figure 3.9: Rollouts of the human's estimated policy are used to identify suboptimal behavior. (a) The true environment that the agent is acting in. (b) The environment that the human believes it is acting in. (c) A rollout of the human agent's estimated policy reveals the minimal amount of hazardous states that need to be communicated for policy repair.

they take to communicate) to assist the optimizer with generating more desirable solutions. In SPEAR, predicates are composed in disjunctive normal form to communicate state regions needing reward updates to improve the collaborator's task performance (Figure 3.10).

We characterize the problem of repairing or otherwise manipulating an agent's policy as one of identifying and reconciling divergence between a source and target reward function. Our proposed algorithm relies upon the following assumptions: 1) agents can understand the natural language description of the predicates (mapping string to predicate); 2) agents are operating using a functional planner or policy learning algorithm informed by a reward function (or something analogous); and 3) agents' suboptimal action selections are attributable to a malformed reward function rather than a malformed policy search algorithm or inaccurate dynamics model of the environment [20, 73].

3.2.4 Approach

In this section, we specify the problem of reparative policy elicitation: repairing or manipulating an agent's policy by communicating corrections for model divergences. Our approach to

policy elicitation uses three components that: 1) estimate the agent's reward function; 2) determine important reward function disparities that prevent desirable behavior; and 3) determine which states to provide reward updates for and communicate a corrective explanation. For clarity, we will use the terminology of an 'expert' to refer to an agent who is initiating communicative action, and a 'novice' to refer to an agent or human whose policy is being corrected.

3.2.4.1 Belief Estimation with Active Observation

We formulate our domain as a Markov Decision Process (MDP)[179], wherein an agent acts to maximize an expected reward. M is a MDP defined by the 4-tuple (S, A, T, R) where S is the set of states in the MDP, A is the set available actions, T is a stochastic transition function describing the model's action-based state transition dynamics, and R is the reward function $R : S \times A \times S \rightarrow \mathbb{R}$. Intuitively, M serves as a simulator for an agent in the task domain.

Since the novice human collaborator's internal policy is latent from the perspective of the expert agent, we perform belief estimation (inferring novice agent's policy) of the human's most likely reward function R_h based on the information they can observe, and derive their policy π_h^* assuming that humans optimize expected reward given their current knowledge of rewards: a common practice within inverse reinforcement learning and preference learning literature [17, 20, 73]. Because the only reward information humans receive is communicated either via the expert agent or through observing human behavior, we update the human's reward function R_h and resultant policy π_h^* whenever the agent provides a communicative update or based on human's past actions, similar to [175].

3.2.4.2 Finding Important Model Divergences

Once we have our belief of the novice human agent's possible reward function R_h, we identify divergences between the optimal policy π^* and the policy of the human π_h^* that would cause a reduction in the human's expected cumulative reward. We do this by comparing policies trained on R and R_h, where R is the true reward function of the domain and R_h is the reward function

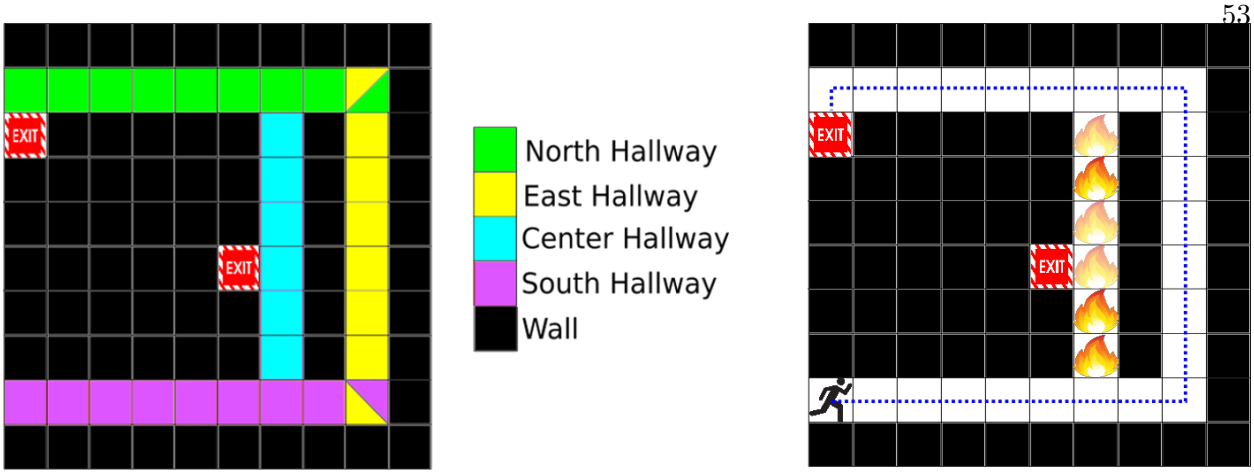


Figure 3.10: We ground reward function updates in language using a Boolean algebra over predicates. (a) A domain map with an overlay indicating states where various predicates are true. (b) After communicating "The center hallway is on fire," a belief update for the human shows potential fire locations, though language imprecision means some of the fires (shown as faded) do not exist. Despite the imprecision, the optimal policy is elicited from this update.

used by the novice human. We then identify a set of states $\bar{S}$ for which we communicate updated reward information to augment R_h, such that a more optimal policy (closer to the expected reward of π^*) is elicited (Figure 3.10b).

3.2.4.3 Communicating State Regions

We approach the problem of efficiently describing state regions as a set cover problem, trying to find the smallest logical expression of communicable predicates to succinctly describe target states as in Hayes and Shah [163]. Unlike prior work, we solve for the minimum set cover of the targeted state region using an integer program formulation that admits approximate solutions as well. The inputs to our IP formulation include:

- A set of state indices $\bar{S} = \{s_1, s_2, \cdots, s_{|\bar{S}|}\}$ that correspond to states with expected reward function divergence that need to be communicated for policy repair.
- A set of communicable predicates $\bar{P} = \{p_1, p_2, \cdots p_{|\bar{P}|}\}$. The following is necessary for a solution to exist:

$\exists\ \bar{Q} \subseteq \bar{P}$, such that $\forall\ s \in \bar{S}$, s is covered by a predicate in $\bar{Q}$, — for every state that needs to be covered ($\bar{S}$), there exists a non-empty subset of predicates $\bar{Q}$ that can cover it.

- A set of costs $\bar{C} = \{c_1, c_2, \cdots, c_{|\bar{P}|}\}$, such that $\forall\ c \in \bar{C}, c \neq 0$ — every predicate has non-zero cost associated with using it to update the human's reward function R_h.
- The desired trajectory for the human $\bar{O} = \{o_1, o_2, \cdots, o_{|\bar{O}|}\}$, where o_i describes the state achieved after taking the i^{th} action following the optimal policy π^* from the start state.

The cost for each predicate can be customized per task, as many factors may influence the cost of a predicate. One such criteria for defining cost can be the length of the string describing the predicate. Such a criteria could generate more easily understood explanations by imposing penalties for being too verbose.

A solution to the policy elicitation problem consists of selecting predicates to communicate reward information about specific state regions such that a more optimal policy is produced within some ϵ bound of the optimal policy's expected reward, $|\mathbb{E}_{\pi^*}(R) - \mathbb{E}_{\pi_h^*}(R_h)| \leq \epsilon$. To minimize this objective while satisfying all the constraints, we define the mathematical formulation of our IP, which we refer to as **SPEAR-IP**, below:

$$min \sum_{j=1}^{|\bar{P}|} c_j x_j + L \sum_{k=1}^{|\bar{O}|} \sum_{j=1}^{|\bar{P}|} v_{kj} x_j \quad \text{subject to} \tag{3.1}$$

$$\sum_{j=1}^{|\bar{P}|} u_{ij} x_j \geq 1 \quad \forall\ i \in [1, |\bar{S}|] \tag{3.2}$$

where we define the $|\bar{S}| \times |\bar{P}|$ matrix U by

$$u_{ij} = \begin{cases} 1, & \text{if } s_i \text{ is covered by } p_j \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

and, the $|\bar{O}| \times |\bar{P}|$ matrix V by

$$v_{kj} = \begin{cases} 1, & \text{if } o_k \text{ is covered by } p_j \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

L is a large constant that acts as a penalty term and soft constraint violation indicator. The indices i and k are in sets $\bar{P}$ and $\bar{O}$ respectively, and $x_j \in \{0, 1\}$ indicates whether the predicate p_j is included ($x_j = 1$) or excluded ($x_j = 0$) from the set cover. Equation 3.1 minimizes the cost of the set cover while prioritizing completeness. The first term of Equation 3.1 minimizes the total cost of the chosen predicates, while the second term penalizes the objective function for any overlap of the selected set cover with the desired path when communicating a negative reward (this can be inverted for positive reinforcement). Equation 3.2 provides a hard constraint for the inclusion of states from $\bar{S}$ in the set cover. Equation 3.3 defines the elements of matrix U, which encapsulates cover constraints from $\bar{S}$ (inclusion of all states). Equation 3.4 defines the elements of matrix V using the desired trajectory $\bar{O}$, encapsulating the requirements for eliciting the desired policy.

The second term from the objective of Equation 3.1 can be removed to enforce a hard constraint to find an exact set cover that excludes states on the desired path. However, this approach restricts the capability to find solutions that would provide a **near** optimal policy update. This second term leads to **three possible cases**: 1) Set cover solution with a low cost (cost $< L$); 2) No solution for the set cover; and 3) Set cover solution with high cost (cost $> L$).

Cases 1 and 2 are straightforward and describe the ability of SPEAR-IP to solve the set cover. Case 3 is interesting and provides the option of further exploration to find an alternate solution. Using the set cover from Case 3, we can identify overlapping states with the desired path. By penalizing these states in the true reward function R, we can simulate an alternate desired policy, effectively coming up with a contingency for imprecise language. This process of penalization is repeated until either Case 1 or 2 is reached.

SPEAR-IP is a specialized variant of the NP-hard set cover problem, with added constraints and objectives. The runtime varies based on problem characteristics and solver efficiency [180]. We

discuss the runtime in Section 3.2.5.2.

Algorithm 2: Single-shot Policy Elicitation for Augmenting Rewards (SPEAR)

Input: MDP (S, A, T, R), Min. Reward Threshold R_L, Agent Reward Function R_h, Current state s_c, Num. Rollouts k

Output: Semantic Reward Correction

$\bar{S} \leftarrow \emptyset$; $L \leftarrow$large scalar value;
$R_h^* \leftarrow R_h$; // Best possible agent reward function
for rollout in range(1 to k) do
    $r_c \leftarrow 0$; // Cumulative reward
    $\pi_h \leftarrow$policy trained on R_h;
    // Find states to update for best possible R_h^*
    $s \leftarrow s_c$;
    while s **is not a terminal state do**
        // Perform forward rollout of π_h
        $s' \leftarrow T(s, \pi_h(s))$;
        $r_c \leftarrow r_c + R(s, \pi_h(s), s')$;
        $s \leftarrow s'$;
        if $r_c \leq R_L$ **then** // Reward too low
            $\bar{S} = \bar{S} \cup s'$; // Track state for later
            $R_h^*(s, \pi_h(s), s') \leftarrow R(s, \pi_h(s), s')$;
            break;
$\pi_h^* \leftarrow$policy trained on R_h^*; $\pi^* \leftarrow$policy trained on R;
Set_Cover, Objective $\leftarrow$ predicate_selection$(\bar{S}, \pi_h^*, ...)$;
if objective is no_solution then exit;
if objective $\geq$ L (from Eq. 3.1) then
    for rollout in range(1 to k) do
        $s \leftarrow s_c$;
        while s **is not terminal do**
            // Perform forward rollout of π^*
            $s' \leftarrow T(s, \pi^*(s))$;
            if $s' \in$ **Set_Cover then** $R(s, \pi^*(s), s') \leftarrow -L$;
            $s \leftarrow s'$;
    go to 1;
else
    feedback $\leftarrow$ "There's a bad reward in **Set_Cover**."
    return feedback

3.2.4.4 Algorithm

Algorithm 2 details how SPEAR identifies communicative predicates. We build intuition for this using our emergency evacuation example from Figure 3.8-right. Given an estimate of the

novice human agent's reward function, SPEAR communicates a reward update in an attempt to elicit the best possible policy. To achieve this, (Line 3-16) we perform multiple forward rollouts of the novice agent's policy π_h derived from R_h and (Line 13-16) compare the accumulated expected reward to the reward threshold R_L. The moment this threshold is crossed, the reward from that transition is determined to be relevant for updating R_h and the state is added to the set cover. The threshold R_L is domain-specific and depends on the threshold for "failure", which will vary across reward functions and the meaning of reward. Note that suboptimal does not necessarily imply failure: failure is a subjective distinction that the domain designer must make.

This can be easily illustrated for our example, where Figure 3.9b shows the belief of a novice human trying to evacuate the building. Figure 3.10b gives insight into how updating the human's reward function can result in an optimal policy even if the human's belief about the fires doesn't truly match the environment. Likewise, for a different building layout, a policy causing a human to take a longer path than optimal may still be considered both successful and an acceptable improvement if it causes the human to safely reach an exit.

After finding the states responsible for meaningful reward divergence, we find a minimal set cover of communicable predicates that map onto these states. This is achieved through **SPEAR-IP**, discussed in Section 3.2.4.3. We use an off-the-shelf optimizer [181] to solve SPEAR-IP (Line 18), where the *predicate_selection* method takes in the states to cover ($\bar{S}$) and the best agent policy π_h^*. This gives a set of communicable predicates and a final cost using Algorithm 3.

Line 19 checks whether or not a solution exists for a given set of predicates. In lines 20-28, our algorithm evaluates case 3 to determine if alternate solutions exist. In lines 23-27, SPEAR performs multiple forward rollouts of the optimal policy to find states responsible for a high cost. In line 26, these states are penalized in the true reward function (R) to incentivize the algorithm to find an alternate solution which avoids these states. This enables SPEAR to explore alternate solutions, making it more robust in applications where the available predicates are insufficient, overcoming barriers due to imprecise or unavailable language.

In line 28, now that all the appropriate states are penalized, the algorithm repeats the

whole procedure from the beginning with a modified R, continuing this process until it finds a low objective solution or no solution (case 1 or 2). Finally, in line 30, the update is serialized as semantic feedback using the *Set_Cover*. This feedback generation strategy uses negative reward to drive a novice human agent's policy away from undesirable states, as improved policies can then be elicited through the exclusion of states along the human's (hypothesized) originally intended path. While similar outcomes can be achieved via positive reward, a state exclusion-based strategy generally allows for the use of less precise predicates.

In Algorithm 3, we produce the set cover for communicating the reward update. Lines 4-7 evaluate states we want the novice agent to traverse (the desired trajectory $\bar{O}$) by performing a forward rollout of the best attainable policy π_h^*. In lines 8-13, the matrices U and V from Equation 3.3-3.4 are defined, which form the basis of the constraints governing the inclusion of states to cover and exclusion of optimal states (when giving information about negative reward) in the set cover respectively. Finally, in line 15, **SPEAR-IP** is solved using the matrices U and V to give *Set_Cover* and *Objective*.

3.2.5 Experimental Evaluation

To demonstrate the utility of our algorithm and validate the effectiveness of its generated explanations, we present a series of algorithmic evaluations and human-subjects user studies. This section focuses on an empirical analysis of SPEAR's algorithmic performance, considering domain size and predicate count in a simulated emergency evacuation scenario. In Sections 3.2.6 and 3.2.7, we present two separate human-subjects studies aimed at evaluating the usefulness and applicability of explanations generated by SPEAR across various applications, targeting expert and novice users respectively.

3.2.5.1 Evaluation Domain

In this scenario, our expert autonomous agent must help a human agent escape from a smart building in which a series of fires has broken out, as shown in Figure 3.9. While people

Algorithm 3: Predicate selection (Minimal set cover)

Input: Set of States to cover $\bar{S}$, Agent policy π, MDP (S, A, T), Set of predicates $\bar{P}$, Current state s_c

Output: Set_Cover and Objective (high, low, or no solution)

Set_Cover $\leftarrow \emptyset$; // Predicates in min set cover
$O \leftarrow \emptyset$; // States we don't want to cover
$s \leftarrow s_c; O \leftarrow O \cup s$;
while s **is not terminal do**
    // Perform 'optimistic' forward rollout of π
    $s \leftarrow$ most likely transition from $T(s, \pi(s))$;
    $O \leftarrow O \cup s$; //append occupied states
//Define matrix U to be $|\bar{S}| \times |\bar{P}|$ matrix s.t.
for $i \in [1, |\bar{S}|]$, $j \in [1, |\bar{P}|]$

$$u_{ij} = \begin{cases} 1, & \text{if } s_i \in \bar{S} \text{ is covered by } p_j \in \bar{P} \\ 0, & \text{otherwise} \end{cases}$$

//Define matrix V to be $|\bar{O}| \times |\bar{P}|$ matrix s.t.
for $k \in [1, |\bar{O}|]$, $j \in [1, |\bar{P}|]$

$$v_{kj} = \begin{cases} 1, & \text{if } o_k \in \bar{O} \text{ is covered by } p_j \in \bar{P} \\ 0, & \text{otherwise} \end{cases}$$

//For SPEAR-IP: refer to Equations 3.1 - 3.2
Set_Cover, Objective $\leftarrow$ SPEAR-IP(U, V);
return Set_Cover, Objective

in the building are not aware of the fire locations, they are assumed to understand the generated predicate-based language.

The building layout for each trial is generated with a stochastic placement of rooms, hallways, and exits over a gridworld of fixed size. For example, a gridworld of size 40X40 (1600 states) may have parameters: rooms = 10, hallways = 40, and number of exits = 5. We utilized randomly generated predicates for validation to demonstrate the generalizability of our approach. To accommodate the full range of possible state regions to cover using predicates, we consider composite predicates (intersections of base predicates) from the power set of base predicates, providing each scenario with an upper bound of $2^n - 1$ possible predicates given n base predicates.

Once the building layout and predicates are generated, we observe a randomly placed human agent exploring a hazard-free building over a fixed number of episodes (e.g., 25 episodes for 40X40 states). The human agent's policy is then trained to always seek the shortest path to the closest exit that was discovered during these exploration episodes. Initially, SPEAR has no knowledge about

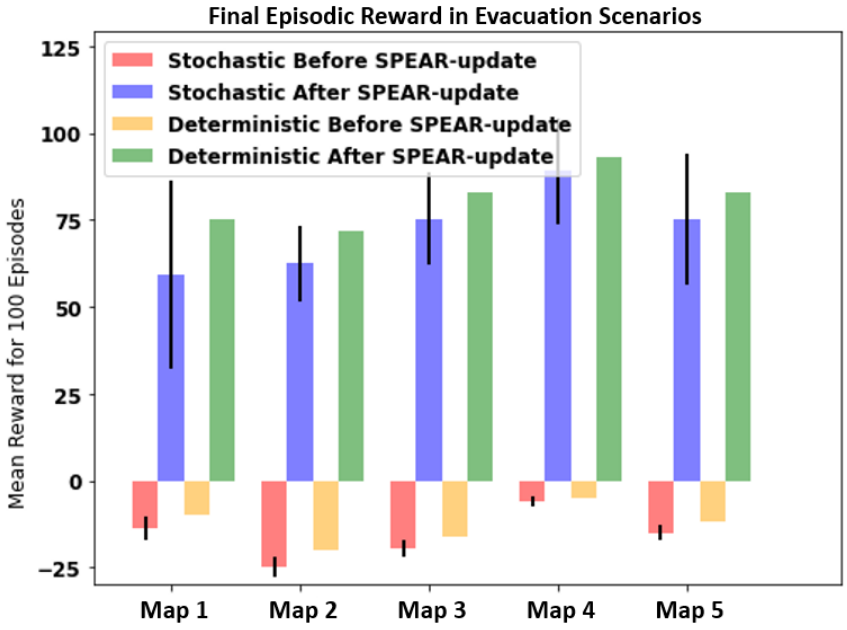


Figure 3.11: SPEAR's evaluation in stochastic and deterministic evacuation domains (25x25) shows substantial increase in episodic reward from symbolic reward updates.

which exits the human is aware of, but gradually, its belief about the human's reward function is updated from these past observations [175]. Next, we begin the evaluation by adding fire to the building randomly. Once this process completes, we start the SPEAR evaluation. Predicates from SPEAR-IP are used to update the human agent's reward function during the episode. We update the policy of the human using the repaired reward function after each update.

3.2.5.2 Algorithmic Performance

We evaluate performance based on state space size and predicate count using various building layouts. We test its adaptability to diverse state mappings with randomly generated, n-sphere shaped predicates. In structured environments like Figure 3.10a, building-grounded predicates make it easier to find hazardous predicates. However, stochastic predicate placements, which simulate non-tailored languages, complicate solutions and increase computational time, highlighting the algorithm's resilience in predicate-domain mismatches.

We assess our algorithm in an evacuation scenario for both deterministic and stochastic environments (Figure 3.11). In the stochastic domain, a stochastic transition function was applied

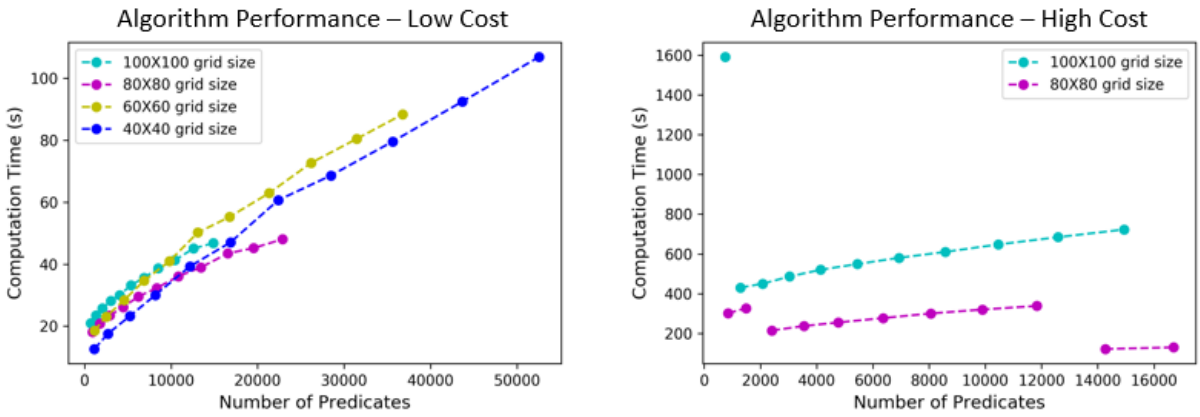


Figure 3.12: (Left) SPEAR's performance on low-cost (case 1) maps shows linear runtime growth with predicate count. (Right) High-cost map (case 3) performance scales linearly with predicate count; sharp time drops (purple) result from SPEAR more quickly finding solutions with new predicates.

describing the model's action-based state transition dynamics (agent takes prescribed action with a probability of 85%). For each map in both environments, we evaluate our algorithm for 100 episodes. In each episode the reward was computed both before and after the SPEAR update (exit: +100, fire: -100, and each step: -1). For our stochastic evaluation, we set our forward rollout count parameter, k, to 10. The results from our simulations show a substantial improvement in the episodic reward after the SPEAR update, (Figure 3.11) demonstrating utility in both deterministic and stochastic domains.

Furthermore, we analyze performance as a function of predicate count by dividing into the two success cases described in Section 3.2.4.3: 1) Maps with a low cost solution (Case 1); and 2) Maps with high cost solution (Case 3). Figure 3.12-left shows how algorithm performance changes with increasing predicate count on low cost maps. For Case 1 solutions, our algorithm exhibits linear computational time as a function of the number of predicates, with the set cover able to be computed within 50 seconds with nearly 10,000 communicable predicates. To put the significance of this in context, the prior state-of-the-art using the Quine-McCluskey(QM) algorithm [163] takes approximately 60-120 seconds for solving set covers with 10 predicates on similar hardware, with performance deteriorating exponentially as the predicate count increases. Our approach achieves an order-of-magnitude improvement over the QM-based method, operationalizing insights from past

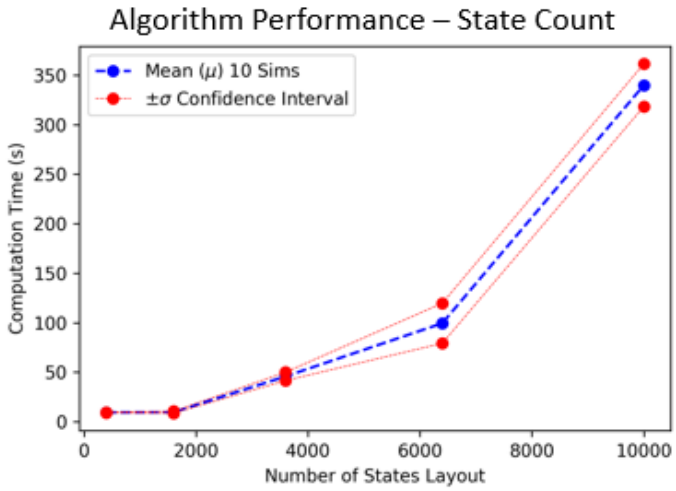


Figure 3.13: SPEAR's evaluation in stochastic and deterministic evacuation domains (25x25) and its performance as state space increases, revealing polynomial computation time with state space.

work [163] to communicate state regions — underpinning SPEAR's policy update.

We plot the performance for Case 3 solutions as predicate count increases (Figure 3.12-right). We observe that the plots again scale linearly in practice with respect to the number of predicates, but with higher computation time due to multiple SPEAR runs. Computation time is higher here because the algorithm has to explore alternative solutions for the desired policy, solving for set cover solutions multiple times (Section 3.2.4.3).

SPEAR achieves a dramatic improvement in computation time over the QM-based set cover approach by pre-computing predicates over the state space, which alleviates some of the online computation during the prime implicant step (exponential in computational time) of the QM approach [182]. Additionally, QM can only provide an exact set cover, failing when there is no exact cover, making it slower and less versatile. SPEAR's use of an approximation heuristic makes it faster and more adaptable than the QM-based approach.

We observed interesting irregularities in performance as evident by results from the 80 X 80 world (Figure 3.12-right-purple). We find that sudden dips in computation time can occur when new language enables the algorithm to find a Case 1 (single-loop) or cheaper Case 3 (multiple loops, but fewer) solution.

The final part of our performance analysis looks algorithm performance as domain size increases with fixed predicate count (100 base predicates). We generate a set of maps with similar parameters for a fixed state space size, sampling from these maps to get the mean and confidence bound for 10 simulation runs as shown in Figure 3.13. A significant take-away from this analysis is the insight that an attention mechanism is more important for abstracting and reducing the domain's state space than for limiting the number of predicates to consider, as prior work anticipated [20, 163, 177].

Result Synopsis. We have shown that **SPEAR** enables substantial improvements in agent performance through policy elicitation, through a novel method for communicating about state regions that substantially outperforms prior work. These contributions enable semantically guided policy manipulation for a much broader class of problems than was previously possible, providing a method that scales linearly with predicate count as opposed to exponentially, advancing the state-of-the-art in autonomous coaching through new algorithms and improved foundational capability.

3.2.6 Study 1: Explanation Quality Evaluation

We evaluated SPEAR explanations through an IRB-approved online user study (n = 12) from a population of graduate researchers in AI and robotics unaffiliated with this line of work in order to solicit feedback from participants with domain familiarity. The study evaluated the effectiveness of these explanations, analyzed the benefits of varying levels of abstraction in explanations, and determined their impact on user comprehension.

3.2.6.1 Experimental Design

The experiment was structured as a single remote session where participants completed five tasks. For each task, participants were shown an image of a scenario featuring a human agent with specific objectives. They were informed that some scenario details were concealed and would be later revealed by an autonomous agent through updates. Participants were tasked with choosing the optimal action from multiple-choice options, similar to [183].

Subsequently, an image illustrating the human agent's optimal behavior was presented, accompanied by three possible updates from the expert autonomous agent. Participants ranked these updates using subjective metrics from established questionnaires [107, 184], considering which update, if received earlier, would most likely lead to the desired behavior. This procedure was repeated for all five tasks. Upon completing the tasks, participants were administered a demographic survey and an open-ended post-experiment survey.

3.2.6.2 Explanation Updates

Participants were presented with three distinct types of explanation updates:

C1. No Set Cover: Here, the agent conveys good or bad states in the task by utilizing machine language in conjunction with a string template (e.g., "There are bad rewards when Obj_1 and Obj_5 are in states: {x =(6.5-7.5), y = (6.25)}."). Essentially, it provides information about crucial states with model divergence without employing any set cover to translate them into natural language updates.

C2. Exact Set Cover: This approach is similar to C1, but key states with model divergence are communicated using exact set cover conditions (e.g., "There are bad rewards when the red cereal and purple cereal are placed in the trash can."), similar to [163].

C3. Relaxed Set Cover: This approach is similar to C2, but key divergent states are conveyed with a focus on abstraction, although at the cost of specificity (e.g., "There are bad rewards when cereals are placed in the trash can."). Example explanations provided here correspond to the scenario in Fig. 3.8-left.

Both 'exact set cover' and 'relaxed set cover' explanations were generated utilizing the SPEAR-IP formulation.

3.2.6.3 Experimental Tasks

We chose three distinct domains for wider applicability and generalizability. Task 2 and Task 4 specifically test both positive and negative reward updates.

Task 1 & Task 2. Navigation Tasks: An agent traverses a gridworld domain from start to goal using the most cost-effective path. Path costs vary based on good or bad state encounters, with some grid cells colored to facilitate task abstraction and to showcase language grounding, inspired by [183].

Task 3 & Task 4. Fire Rescue Tasks: Here, an agent attempts to rescue victims from a burning building, searching a limited number of floors sequentially. The user is not aware of which floors or rooms have fire or victims, which is later communicated via updates.

Task 5. Robotic Cleaning Task: A robot removes trash from a tabletop, leaving some items believed to be trash by participants. In this scenario, the expert agent justifies its actions by providing explanations to the participants (Fig. 3.8-left), similar to [20].

3.2.6.4 Hypotheses

We hypothesized that users would rate the 'relaxed set cover' explanations higher than both the 'no set cover' and 'exact set cover' explanations **(H1)**, based on the following metrics: usefulness (H1a), conciseness (H1b), comprehension (H1c), cognitive load (H1d), decision-making (H1e), and interpretability (H1f). We also hypothesized **(H2)** that participants would prefer structured semantic explanations over numerical reward explanations (i.e., 'exact set cover' > 'no set cover') across the same subjective metrics as H1 (H2a-H2f).

3.2.6.5 Results and Analysis

We recruited 12 participants (6 males and 6 females) with ages ranging from 23 to 40 years old ($M = 27.92; SD = 4.66$). For each task, we assessed the explanation updates on ranked data provided by the participants using a nonparametric Kruskal-Wallis Test with explanation updates as a fixed effect. Post-hoc comparisons used Dunn's Test for analyzing explanation types for stochastic dominance.

Task 1 & Task 2. We found a significant effect in favor of the 'relaxed set cover' update over 'no set cover' and 'exact set cover' for usefulness, conciseness, comprehension, cognitive load, decision-

making, and interpretability, with all p-values below the 0.05 threshold for both Task 1 and Task 2. For Task 1, post-hoc analysis with Dunn's Test indicated that participants consistently preferred 'relaxed set cover' over 'no set cover' for usefulness, comprehension, cognitive load, and interpretability ($p < 0.05$). Similarly, for Task 2, Dunn's Test revealed that participants rated 'relaxed set cover' over 'exact set cover' across all measures with higher means ($p < 0.0001$), **validating H1a-H1f** for both tasks.

Interestingly, in both Task 1 and Task 2, 'no set cover' had higher means over the 'exact set cover'. For Task 1, Dunn's Test also revealed a preference for 'no set cover' over 'exact set cover' in terms of conciseness ($p < 0.01$), thus **invalidating H2b**. Similarly, for Task 2, we found the same significant results across all measures ($p < 0.05$), thus **invalidating H2a-H2f** for Task 2. We posit that the preference for 'no set cover' over 'exact set cover' may derive from the simplicity of these navigational tasks, implying that in simpler scenarios, providing full reward information in numerical form might be more effective than converting to semantic form. This partially conforms to results from [183].

Task 3 & Task 4. For both tasks, post-hoc analysis using Dunn's Test favored the 'relaxed set cover' over 'no set cover' across all measures ($p < 0.01$), **validating H1a-H1f**. There were higher mean ratings in Task 3 for 'relaxed set cover' over 'exact set cover' with respect to cognitive load ($p = 0.021$) and interpretability ($p = 0.002$). Post-hoc tests on 'exact set cover' and 'no set cover' revealed that participants preferred the former over the latter for all measures except interpretability ($p < 0.05$), **validating H2a-H2e**– the opposite of what we saw in first two tasks. Dunn's test also shows significant differences for Task 4 between mean ratings of 'relaxed set cover' and 'exact set cover' for comprehension ($p = 0.021$) and interpretability ($p = 0.003$). Similar to Task 3, post-hoc tests on 'exact set cover' and 'no set cover' show a higher mean of the former over the latter for all the subjective metrics except comprehension ($p < 0.05$), **validating H2(a,b,d,e,f)**. This demonstrates a stronger preference for structured explanation as domain complexity increases.

Task 5. Post-hoc analysis using Dunn's Test showed that participants favored the 'relaxed set cover' over the other conditions ($p < 0.001$) across all measures, **validating H1a-H1f**. This

demonstrates a preference towards structured explanations over numerical, **validating H2a-H2f**.

Post-experimental Survey. Participants were asked which update they would prefer to use if they had to complete more tasks. 'Relaxed set cover' was preferred over alternatives. Based on a one-sample test of proportions, 9/12 participants chose 'relaxed set cover'; a greater proportion than the expected random proportion of 0.33 ($z = 3.09$, $p = 0.002$). Participants justified their choice for 'relaxed set cover' citing reasons such as conciseness, ease of understanding, and reduced cognitive load, which align with our subjective findings from the experiment.

Result Synopsis: In most tasks and measures, 'relaxed set cover' consistently outperformed both 'no set cover' and 'exact set cover'. Additionally, 'no set cover' performed better than 'exact set cover' in the simpler tasks but the trend reversed with the increase in task complexity, **indicating the need for structured and simpler explanations as the task complexity increases.** The degree to which 'relaxed set cover' outperformed the other two varied among tasks and measures. For example, in the robotic cleaning task, 'relaxed set cover' significantly outperformed the other updates in all measures. This pattern suggests that the advantages of relaxed explanations become more apparent with increases in the state space and task complexity. In summary, **our findings strongly suggest that SPEAR-based reward explanations are not only more useful than numeric ones but also promote better understanding, decrease cognitive load, and improve interpretability**.

3.2.7 Study 2: Explanation Type Evaluation

We conducted a follow-up IRB-approved study (n = 38) with a user base recruited from the Prolific platform (prolific.co). The focus of this study was to assess the utility of providing state-based reward updates as opposed to traditional plan explanations [158] (i.e., step-by-step descriptions of a plan) in terms of task performance and task awareness.

3.2.7.1 Experimental Design and Protocol

The study utilized a 2 × 1 between-subjects experimental design to evaluate two types of semantic guidance: (1) plan explanation, referred to as the 'prescriptive' condition, and (2) reward explanation, known as the 'descriptive' condition.

The experiment was administered online in several batches with randomly assigned conditions using the Prolific platform. To ensure higher participant quality, we filtered for those who had both completed at least 100 approved studies on Prolific and had an approval rate of 95% or higher. This study's methodology mirrors the approach taken in Study 1, mentioned in Section 3.2.6 where participants completed the same five tasks.

For each task, participants were presented with the same scenarios as in the previous study. They were informed that some details of the scenario were concealed and that an autonomous agent would assist them in solving the task. We initially tested their baseline knowledge and policy preferences by asking them to select the optimal action from multiple-choice options. Following this initial assessment, an update was provided based on the experimental condition (more details in the next subsection). Participants made their updated choices based on the new information. Upon completing all tasks, participants were given a demographic survey, a post-experimental questionnaire, and an open-ended survey.

3.2.7.2 Explanation Updates

We presented participants with two distinct types of explanation updates depending on their experimental condition: 1) **Plan Explanation:** Here, the robotic agent communicates a step-by-step plan to a human teammate. For example, *"Begin at your current position. Move one cell to your right...."* These explanations are generated by the GPT-4 model [185], prompted by the free-form text provided by participants in Study 1, where they described the robot's actions step-by-step. These final explanations were checked for correctness by experts.

2) **Reward Explanation:** These explanations are similar to those presented in the 'relaxed set

cover' condition of Study 1 (e.g., *"There are bad rewards in the purple and orange cells."*).

3.2.8 Experimental Tasks

We utilized the same three distinct domains and five tasks from Study 1, comprising two navigational tasks (Tasks 1 and 2), two fire rescue tasks (Tasks 3 and 4), and one robotic cleaning task (Task 5).

3.2.8.1 Hypotheses

Through a human subjects study, we evaluate the following four hypotheses, partitioned into two categories:

H3: Subjective Hypotheses

H3.a: Participants will find the reward explanation more trustworthy than the plan explanation, as the transparency of the recommendation leads to increased trust [20, 186, 187].

H3.b: Participants will perceive the plan explanation condition as less stressful and demanding compared to the reward explanation condition, due to the presence of clear recommendations.

H4: Objective Hypotheses

H4.a: Participants will have a better understanding of the task when provided with a reward explanation compared to a plan explanation, as it offers insight into the task through agent updates.

H4.b: Participants will perform similarly in both conditions (i.e., they will make the correct policy choice after an update is provided).

3.2.8.2 Measurement

We recruited 39 participants via the Prolific platform but excluded one due to failing attention checks, leaving 38 participants (21 male, 16 female, 1 unspecified), with 19 participants per condition, aged 19 to 70 years ($M = 38.16; SD = 13.58$). Of these, 16% reported working in STEM fields, and 68% reported having a bachelor's degree or higher.

We evaluated our hypotheses using both subjective and objective measures. We administered

a post-experimental survey consisting of 7-point Likert scales, similar to the one from Study 1 [188, 189, 190]. Based on these questionnaires, we identified two key concepts to validate our hypothesis: **Trust and Workload**.

The **Trust** scale consists of 4 items: confidence, reliability, trust, and dependability (Cronbach's $\alpha = 0.98$). **Mental Load** scale consists of 3 items: demandingness, hurriedness, and effort (Cronbach's $\alpha = 0.82$).

For objective metrics, we recorded the following items: **Policy Accuracy (PA)**, the total number of tasks for which a participant chose the correct policy; and **Knowledge Accuracy (KA)**, the total number of tasks for which a participant had accurate final knowledge.

3.2.8.3 Results and Analysis

Subjective Analysis. To test our subjective hypotheses, we analyzed post-experiment 7-point Likert scales using independent samples t-tests for between-condition comparisons.

The trust scale revealed a significant effect favoring the 'descriptive' condition over the 'prescriptive' condition [$t(36) = -2.74, p = .009$], with higher trust scores ($M = 5.83$) versus ($M = 4.66$), **validating H3.a**.

The mental load scale results showed no significant difference [$t(36) = -1.83, p = .076$], with mean scores of ($M = 3.56$ for 'descriptive') and ($M = 2.73$ for 'prescriptive'), respectively. Due to the lack of statistical significance, **H3.b is inconclusive**, and more data is required to definitively address this hypothesis.

However, participants provided insights that pointed toward two interesting trends. First, some participants who were unsure and unconfident, interpreted the robot's direct guidance in the 'prescriptive' condition as a sign of confidence, leading them to stop thinking critically.

- "**I followed the directions from the agent closely and ignored what I first selected.**"

Second, some participants expressed confusion and were seeking more information and un-

derstanding when following prescriptive explanations.

- "**They [explanations] were useful but didn't explain why the decision was the best...**"
- "**...explanations were useful, but the system assumed I understood the context of the situation without explaining anything.**"

Whereas for the 'descriptive' condition, many participants not only found the explanations useful but also felt that they encouraged active thinking patterns, leading to a positive impact on trust, as evidenced by the quantitative results:

- "**The information made me make more calculated decisions.**"

These results align with previous findings in the xAI literature, suggesting that some individuals may **over-trust the guidance if they lack confidence in themselves, while others may under-trust and become frustrated if they lack sufficient rationale for the robot's guidance** [35, 186, 191, 192]. Furthermore, insights into the robot's decision-making process can encourage people to engage in more active thinking patterns [175, 193, 194].

Objective Analysis. To assess our objective hypotheses, we analyzed task metrics using a nonparametric Mann-Whitney test to evaluate differences between conditions.

First, we conducted a Mann-Whitney test to assess if there were differences in knowledge accuracy (KA) between conditions. The analysis revealed statistical significance in favor of the 'descriptive' condition with $M = 4.11$ compared to the 'prescriptive' condition with $M = 1.82$ for KA scores, with $z = -4.394$ and $p < .0001$, suggesting that participants in the descriptive condition demonstrated higher knowledge accuracy per task scores than those in the prescriptive condition. These findings serve to **validate H4.a**. On the other hand, no statistical differences were observed in policy accuracy (PA) scores between the 'descriptive' with $M = 4.31$ and 'prescriptive' with $M = 4.74$ conditions, with $z = 1.42$ and $p = 0.157$. These results **support H4.b**, indicating that participants in both conditions achieved higher PA scores and predominantly made the correct

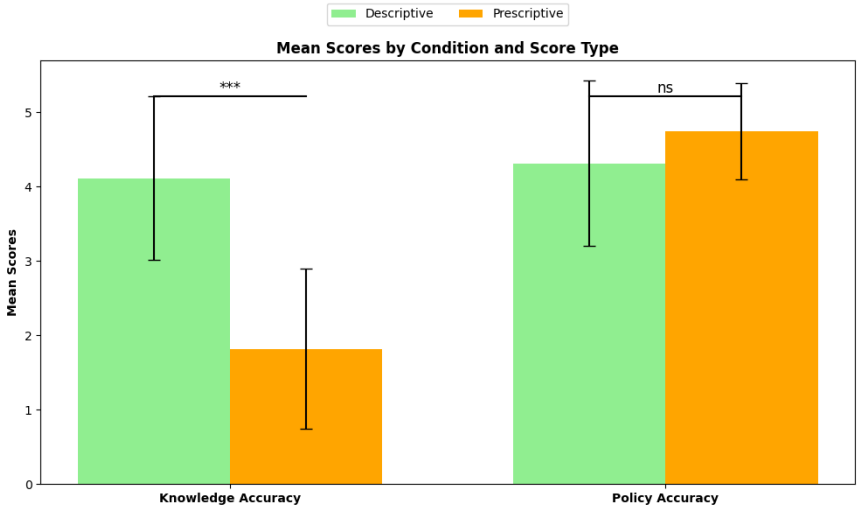


Figure 3.14: Comparison of mean scores for knowledge and policy accuracy between descriptive and prescriptive conditions. Results from the Mann-Whitney test highlight that the descriptive condition (reward-based explanations) not only facilitated correct policy elicitation but also enhanced task awareness, outperforming the prescriptive condition (plan explanations).

choice after an update was provided (refer to Figure 3.14). Therefore, these findings suggest that while participants in both conditions were able to identify the correct policy, those in the 'descriptive' condition not only identified the correct policy but also exhibited better task awareness.

Result Synopsis: In our study, the reward-based explanations consistently outperformed the plan explanations in fostering a higher perception of trust and enhancing users' task understanding, thus improving their policy decisions. Due to the opaqueness of plan explanations, this approach failed to update users' task awareness and led some people to overtrust the guidance. These results suggest that the utility of reward explanations extends beyond mere decision support, offering insights into the robot's decision-making process and fostering a better understanding of tasks, thus promoting active thinking patterns in users.

3.2.9 Robotic Application: Multiagent Cleaning

In this section, we illustrate the application of the policy elicitation process in a scenario involving two heterogeneous robotic agents, specifically robots operating with different state representations. This effectively demonstrates the utility of SPEAR's symbolic reward update in

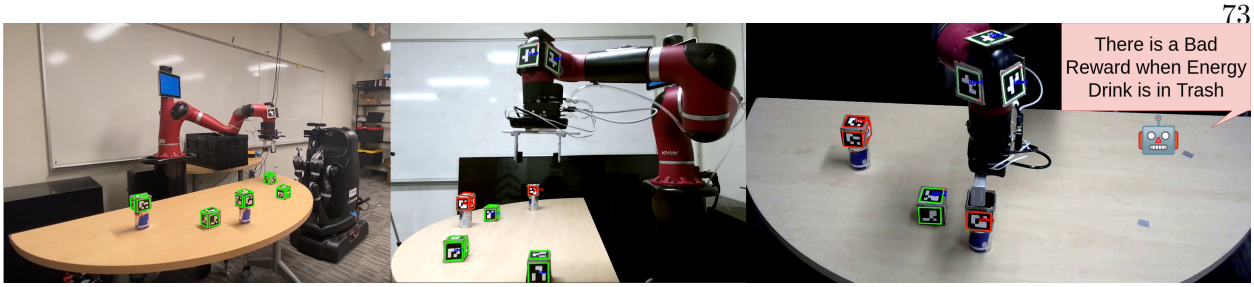


Figure 3.15: (Left) Sawyer (red robot) views a misinformed policy for a tabletop cleaning task as Movo (black robot) moves into position to assist. (Center) Movo, from its perspective views the scene with the correct policy and watches Sawyer remove the correct (green) objects from the table. Movo then observes Sawyer reach for one of the incorrect (red) objects and delivers a verbal reward update. (Right) Sawyer computes an update to its planner thus aligning with the desired policy.

facilitating agent-to-agent manipulation.

Our core insight of policy elicitation relies on modifying an agent's behavior to a desired policy by updating their reward function with targeted feedback in the form of semantic explanations (symbolic updates) during task execution (refer Section 3.2.3 for more details). This allows our method to operate at a level of abstraction that does not depend on the recipient's underlying state space representation, instead only requiring a similar vocabulary of planning predicates.

In many applications, multi-agent robotic systems have no guarantee of operating with identical state representations. Similarly, private entities, such as those in autonomous driving, may not want to share their proprietary reward functions with competitors. Therefore, we leverage common grounding (e.g., language) to communicate updates for efficient behavior modification during the task for repairing another agent's policy.

Assumptions. We implicitly assume a shared predicate grounding between multiple agents (i.e., the same predicate, even if they have different state mappings, generally means the same thing to each agent) even if they operate using different state representations, such that a common symbolic policy update is feasible. While the predicate grounding need not be exactly the same, as different state spaces are unlikely to have one-to-one mappings, we assume the possibility of having a shared natural language via a set of predicates even if it is not exhaustive or comprehensive.

3.2.9.1 Robot-to-Robot Policy Elicitation

Here, one agent needs to correct the policy of a second robotic agent to prevent it from removing specific items during a pick and place task (Figure 3.15). Importantly, these robots do not operate in the same state space, and therefore cannot directly communicate reward function updates to each other. In this scenario, the Rethink Robotics Sawyer has been tasked with clearing *trash* from a table, and is operating with the malformed belief that all objects on the table's surface are *trash*. The Kinova Movo is observing this scene and has accurate knowledge about the environment (i.e., it knows that certain objects in the scene aren't trash). As Sawyer's reach expresses intent to remove one of the items that isn't trash, Movo detects that Sawyer's policy is incorrect. Movo corrects Sawyer's behavior by providing an update for its reward function, allowing Sawyer to compute a better policy. We accomplish this reward update step by generating semantic expressions about the reward function in parts of the state space (i.e., predicate-grounded natural language communicating about a region of negative reward) for Sawyer using SPEAR.

Within this task, Sawyer used a series of ArUco markers [195] to track the 6-DOF poses of various objects (states) on the tabletop. Using these poses, Sawyer systematically transferred all items classified as *trash* to the waste bin using an interruptable pick and place action. As Movo observes Sawyer, it detects Sawyer's end effector reaching for one of the energy drinks on the table, thus indicating the intent to remove that object, and signalling to Movo that Sawyer's policy was malformed. Using SPEAR, Movo is able infer Sawyer's erroneous belief from this observed action (that a non-negative reward is associated with putting the energy drink in the trash). Movo performs a forward rollout with the inferred policy of Sawyer to predict which state transitions with **negative reward** Sawyer will reach. Next, Movo computes an updated policy for Sawyer by determining which states should be assigned negative reward and which of the available predicates are needed for communicating it with Algorithm 3. A DNF formula of predicates that covers the desired states is then computed and communicated by Movo through natural language: "**There is a Bad Reward when energy drink is in the trash**". Finally, Sawyer uses shared predicates

to map the communication into its own state space, update its reward function accordingly, and reconverge a repaired policy, showing successful application of the SPEAR framework.

3.2.10 Discussion and Conclusion

Limitations and Opportunities. Our policy elicitation formulation relies on belief estimation. If the inference of the human's reward function R_h is poor, it will also lead to the degradation of SPEAR's feedback quality for policy elicitation. To accommodate this, we formulated our design to be modular, allowing each component to be easily replaced by any state-of-the-art method [17, 48, 78, 196], provided it takes similar input and provides the required output.

Additionally, we assume users can interpret natural language descriptions of predicates, which we validated through our user study. However, predicate interpretability can vary, often depending on the creator and individual scenario, potentially necessitating the hand-engineering of these predicates for each domain. A possible avenue for future research could explore using large language models (LLMs) to make this predicate generation more robust and generalizable. Similarly, in this work, we use string templated responses for policy elicitation, which is not ideal for a conversational agent. Policy explanations from autonomous agents are expected to be conversational and dialogue-based [30, 31]. A direct extension of this work would look into integrating an LLM framework with predicate-based grounding for human reward coaching, leveraging language grounding from predicates alongside the broad contextual generalizability of LLMs.

Conclusion. In this subchapter we define the problem of **policy elicitation**, the manipulation of a human's behavior through the use of semantically (natural language) grounded reward updates, and present an optimization-based approach for solving it at scale. We introduce a novel integer programming-based algorithm that renders policy explanation [163] and policy manipulation [20] techniques feasible for use in applications substantially larger than previously possible. Our method leverages relaxed explanations, using overstatement or understatement, to deliver concise and useful feedback even with limited shared language. We demonstrate the utility of these policy explanations for both expert and novice users through a series of human subject studies. Our

results indicate that these relaxed reward-based explanations not only enhance individuals' policies but also decrease cognitive load and improve decision-making, all while preserving interpretability. Additionally, we show that these explanations provide insights into the robot recommender's decision-making process, foster a better understanding of tasks, and thus promote active thinking patterns in users, while also facilitating the desired correction of policies.

Chapter 4

Natural Language Communication for Robot Skill Learning and Repair

"Being-in-the-world means being with others."

— Martin Heidegger, *Being and Time*

This chapter introduces a novel human-in-the-loop algorithm that facilitates constraint annotation by novice users using natural language for motion planning problems through a hierarchical semantic process for robot skill learning and repair. A major motivation for this work is the significant interest in training methods that enable collaborative agents to safely and successfully execute tasks alongside human teammates. While effective, many existing methods are brittle to changes in the environment and do not account for the preferences of human collaborators. This ineffectiveness is typically due to the complexity of deployment environments and the unique personal preferences of human teammates. These complications can lead to behavior that causes task failure or user discomfort.

Our intuition is that combining the ease of using natural language with constraint motion planning can enable novice users without much expertise in robotics to perform online robotic skill corrections and personalization, thereby making working and collaborating with robots more accessible and safe. Therefore, in this chapter, we introduce **Plan Augmentation and Repair through SEmantic Constraints** (PARSEC): a novel algorithm that utilizes a semantic hierarchy to enable novice users to quickly and effectively select constraints using natural language to correct faulty behavior or adapt skills to their preferences. We demonstrate through a case study that our algorithm efficiently finds corrective constraints that match the user's intent, providing a path for

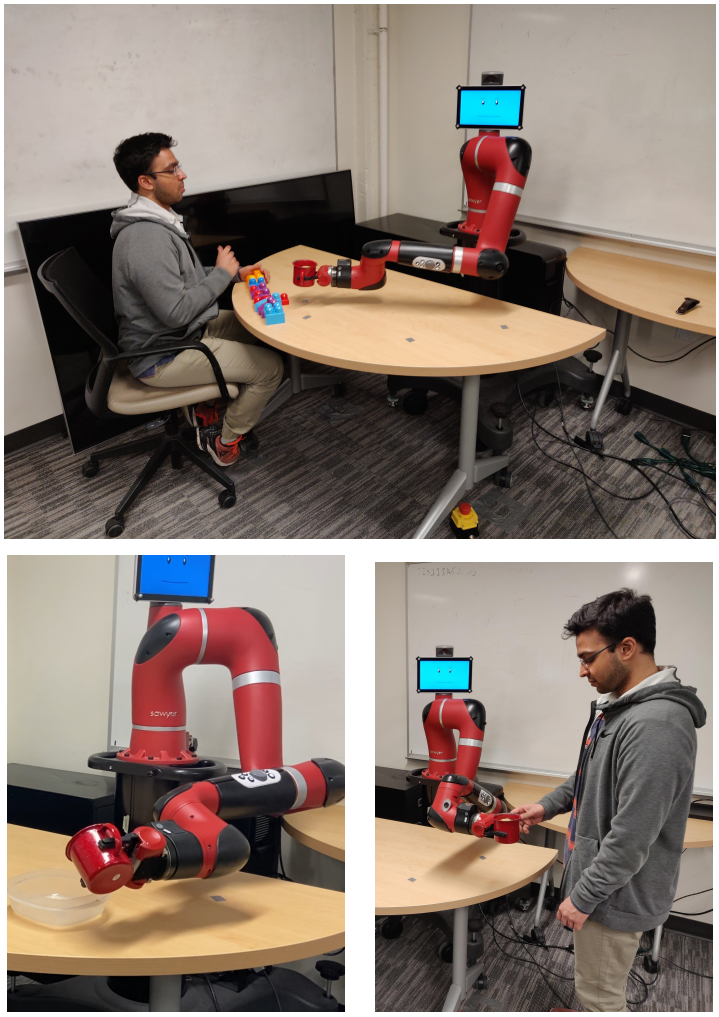


Figure 4.1: User interacting with a Sawyer robot. Three tasks are shown (top to bottom, left to right): 1) A cleaning task where Sawyer attempts to move the cup from one side of the table to the other in front of the user; 2) A pouring task where Sawyer attempts to pour the contents of the cup into another container; 3) A handover task where Sawyer attempts to hand the cup to the user without spilling the contents.

novice users to exploit the advantages of constrained motion planning combined with human-in-the-loop skill training.

4.1 Introduction

The increased availability and prevalence of collaborative robotics has led to growth in our expectations for human-robot teaming and accordingly to the roles and responsibilities assigned to autonomous systems. Robots that collaborate or work in close proximity with humans have safety-

critical requirements imposed on their autonomy, conditioned on task-specific and collaborator-specific parameters. As these deployments become increasingly widespread, their complexity and impact of failure will grow in kind. Consequently, a desirable and pertinent trait for such collaborative agents is the ability to accommodate human users' preferences [197] and requirements[198]. For example, an assistive robot designed to work in elder care environments should take into consideration the different comfort levels of individuals with whom the robot works (e.g. a desired minimum distance). Without such considerations, the robot might potentially cause physical or emotional harm should it behave in a manner that violates expectations [199].

Furthermore, it is highly unlikely that these types of interactions will only occur in the exact environments in which such robots were trained, increasing the likelihood of unexpected or dangerous behavior. This generalization is a central tenet of intelligent automation: being able to utilize a model trained in one environment within a different one [200]. The cost functions being used by these systems to plan or otherwise compute their behavior may not account for crucial factors such as novel environmental artifacts and user requirements or preferences. In this work, we introduce an interactive algorithm to help those who use these systems to add constraints into a robot's planner to create safer, more robust skills that better accommodate user specifications.

It is clear that robots must be able to adapt their behaviors to changes in their environment, as well as to the personal preferences of humans they encounter, to be successful without also levying a burden on those around them. Thus despite the many challenges it poses, in-situ learning will be essential as even modern robots require experts to reprogram them or guide them in the retraining of a skill [198]. Even with state-of-the-art learning from demonstration techniques, retraining skills to achieve reliable performance and predictable behavior takes considerable time and effort [149] or expertise [201]. In order for robots to be able to adapt their skills to novel environments and shifting user preferences, we posit that new techniques enabling non-experts to leverage the power of constrained motion planning are required.

In response to this technical challenge, we present **Plan Augmentation and Repair through SEmantic Constraints** (PARSEC): a novel algorithm that utilizes a semantic hier-

archy to enable novice users to use natural language to quickly select and parameterize constraints that can be applied within a constrained motion planner to correct faulty behavior or adapt skills to accommodate preferences. Core to this methodology is the intuition that novice users must be able to interact in a natural way with the robot and that constraint discovery is greatly accelerated by organizing constraints as leaves in a semantic tree of parameterizations. Our method uses plain language explanations given by a user to bootstrap a brief iterative query process that leads to the specification of an allowable constraint set that matches their intent. The intuitiveness of this process enables skill correction by those without robotics or motion planning experience, making it suitable for a wide audience. The two primary contributions of our work are:

- PARSEC, a human-in-the-loop algorithm that facilitates constraint annotation for motion planning problems via a novel hierarchical semantic process
- An experimental validation and evaluation of PARSEC, assessing its performance in three different robotic case studies using human feedback and demonstrating a statistically significant time reduction for skill correction compared to baseline.

4.2 Background and Related Work

Learning from Human-in-the-loop. Much work has been done analyzing the ability of human feedback to improve robot skill performance. St. Clair and Matarić showed the effectiveness of robot verbal feedback in human-robot task collaborations [202]. Additionally, Sadigh et al. presented an approach for robot production of social communication during human-robot task collaboration to improve in situ decision-making and team performance [203] and Meriçli et al. contributed a method which utilizes corrective human demonstration as a complement to an existing hand-coded algorithm for improving task performance [204].

Similar works look into cognitive inspired architectures that help infer task constraints from natural language and demonstrate through user studies that natural language is the preferred instructions method for modifying robot skills [205, 206]. Our proposed method infers the most

likely correction of the problem, and then initiates communication with the user to resolve the ambiguity before the skill is augmented.

Learning from Demonstration. Researchers have also worked on learning from failed demonstrations. In a paper by Grollman et al., humans are assumed to be sub-optimal and incapable of performing a task correctly. Their failed demonstrations are then used as negative constraints on the robot's exploration [207]. The same group of researchers in other work speculated that in higher dimensions, additional information from the user will most likely be necessary to enable efficient failure-based learning [208]. Our proposed system applies this type of information from the user to improve interaction efficiency during failure correction.

Other researchers have focused on learning robot objective functions from human guidance through physical corrections provided by the person while the robot is acting [149]. A key limitation of this technique is that it requires users with a technical background to perform the skill correction which keeps novice users from being able to benefit from it [209]. Instead of physical demonstrations, humans typically use speech to provide high-level goals or teleoperation commands for autonomy [210]. Kramer et al. [211] compared four natural language understanding models, evaluating their performance to understand domestic service robot commands by recognizing the actions and any complementary information in them. These models learn possible correspondences between parsed instructions and candidate groundings that include objects, regions and motion constraints. In the realm of learning from demonstration (LfD) there has been much focus on repairing faulty skills or even training new skills with only a single demonstration and then providing fine tuned skill adjustment through a user interface [201, 212].

Learning through user preference and querying. Researchers have also worked on learning user preferences over trajectories taken by robotic manipulators. Abdo et al. used a collaborative filtering model to learn user preferences about how best to organize objects in their environment [213]. However, Bobu et al. showed how assuming that a human's desired objective lies within the robot's hypothesis space can lead to irrelevant task corrections [214]. These works demonstrate the importance of having a feedback loop between the user and the robot so that

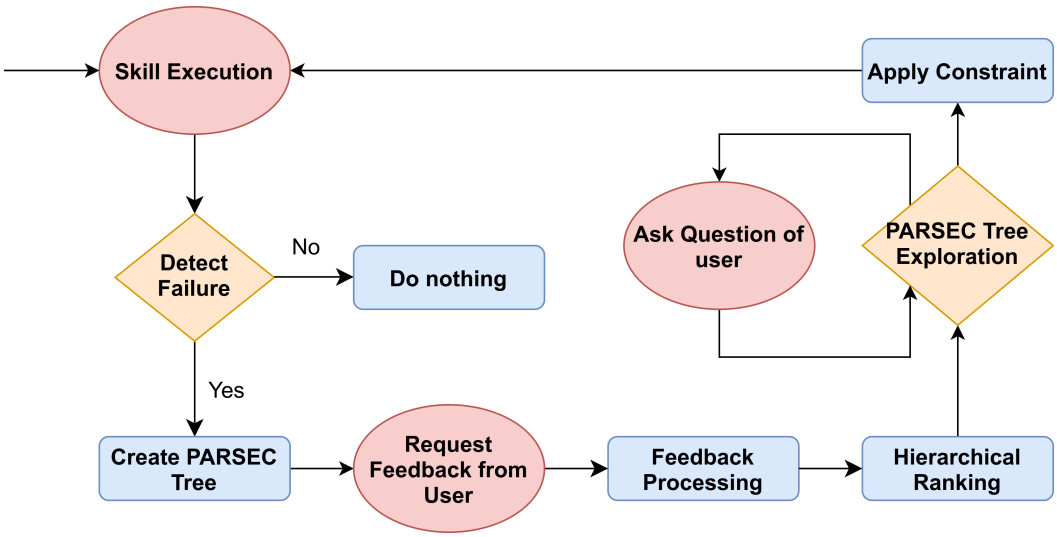


Figure 4.2: Execution loop of PARSEC, beginning with skill execution and proceeding through constraint specification.

correction can occur without confusion.

Querying users for improving performance and learning has been an active field of research as well [215]. Cakmak et al. categorized types of queries users preferred based on the informativeness and ease of answering [216]. Another approach has been for enabling a robot to recover from failures by generating targeted assistance requests[57]. Similarly, Biyik et al. showed another approach of learning through queries focused on generating easy questions through greedy maximization of information gain [217]. In Volosyak et al., the system actively queries the human for task goals or execution assistance, and through speech the user provides a high-level (e.g. "pour a drink") and low level (e.g. "gripper up") instruction [218]. To the best of our knowledge, we believe this is the first work that combines learning from human feedback, constrained motion planning, and in-situ iterative querying of a human user (using natural language) to augment and repair robot skills.

4.3 Methods

In this section, we introduce **PARSEC** (Plan Augmentation and Repair through SEmantic Constraints), an interactive method whereby a robot iteratively queries a human collaborator to determine how to apply constraints to its motion planner for improving its skill performance or

robustness. Our approach enables non-expert users to correct faulty robot skills through natural language feedback, which our algorithm processes and maps to parameterized constraints (e.g., 'stay at least 15cm away from the human').

While constrained motion planning has been shown to be a powerful tool in improving robot task and skill performance, the selection and parameterization of which constraints to apply remains an open, time consuming problem[212, 219].

Our insight is that if we can structure the available constraints and their parameterizations to **maximize the information utility of each question** and algorithmically reduce the problem space **based on the user's feedback**, we can substantially reduce the level of effort required to incorporate constrained motion planning into learning from demonstration and human-in-the-loop skill repair.

Preliminaries. PARSEC is a post-hoc method applicable once the learning agent has been trained to execute a specific task within a training environment, meaning the agent already has the foundational elements of the planning problem defined (i.e., motion planner, goal states, and cost function). The outcome of PARSEC is a list of parameterized constraints to apply to the motion planning problem, with the intent that the application of these constraints will prevent failure modes not initially captured by the planner's cost function. One intuitive use case is that constraints can be used to fill in for 'common sense' (or user preferences) that the cost function may not properly encode, such as applying the constraint that a cup in the gripper must always be upright, since the planner's cost function may not encode avoiding spilling the cup's contents.

We define a **constraint** to be a Boolean function mapping a state of the world to **true** if that state is not in violation of the constraint represented within the function and **false** otherwise (akin to STRIPS predicates [178]). For example, min_distance(object_1, object_2, distance_in_cm): $State \rightarrow \{True, False\}$ could be a constraint that evaluates to **true** only when object_1 is at least distance_in_cm from object_2 in the provided state vector (e.g., **min_distance(cup, laptop, 10)** would return **true** if the cup and laptop are at least 10 cm apart, **false** otherwise). Within PARSEC, we characterize the parameters for constraints as either belonging to a discrete finite

Algorithm 4: Plan Augmentation and Repair through Semantic Constraints (PARSEC)

Input: Motion Planner *Planner*, Start state s_0, Goal state set $\bar{G}$, Known Constraints K, List of constraint functions $\bar{C}$, List of parameter types $\bar{S}$, Dictionary mapping parameter types to lists of valid assignments P

Output: List of parameterized constraints for skill segment $s_0 \rightarrow \bar{G}$ or $False$ on failure

```
constraints ← K;
if Planner.plan(s_0, Ḡ, constraints) ≠ ∅ then return;
tree ← CreatePARSECTree(C̄, S̄, P);
while do
    // No successful plan from s_0 to a state in Ḡ
    response ← RequestExplanation();
    rankedTree ← ScoreTree(tree, response, S̄, P);
    found ← False; new_constraints ← ∅;
    for node ∈ rankedTree do
        new_constraint ← AskQuestion(node);
        if new_constraint != ∅ then
            found←True; break;
    if found is False then return False;
    constraints.append(new_constraint);
    if Planner.plan(s_0, Ḡ, constraints) ≠ ∅ then break;
return constraints;
```

set (e.g., "objects": ['parts bin', 'table', 'block'] - a list of object names in the environment) or representing a continuous or innumerable set (e.g. "distance": a real-valued quantity expressed in centimeters). These sets represent the domain knowledge of the learning agent that it can use for specifying and communicating about constraints.

We use this domain knowledge to create an informed structural framework for our query mechanism to efficiently solicit user feedback, resulting in a more rapid constraint specification process that requires less human effort. In this section, we detail our method of intelligently querying the human collaborator for determining beneficial motion planning constraints.

4.3.1 PARSEC Algorithm

Being a post-hoc method, PARSEC is intended to be applied after initial skill specification or learning. The execution flow of PARSEC can be seen in Fig. 4.2. We model skill execution as solving a motion planning problem from some initial state (s_0) to any one of a set of goal states ($s_g \in \bar{G}$). More specifically, our work is intended to be applied within domains modelled as Markov

Decision Processes defined by (S,A,T) where **S** is the set of states, **A** is a set of actions the agent can choose from, and $\mathbf{T} : S \times A \times S \rightarrow \mathbb{R}$ is the transition function that provides the likelihood of transitioning between two states given an action.

PARSEC requires a Boolean signal to indicate whether the robot has successfully completed its task or if it has failed (Fig 4.2: Detect Failure step). In collaborative Human-Robot Interaction scenarios, failure may be indicated by adverse human behaviors (e.g., human retreating from the robot/workspace, showing annoyance, etc,.) resulting from execution of the skill (as opposed to not being able to plan from start to goal state). The entry point for our approach is Algorithm 4, which interactively produces a list of parameterized motion planning constraints when given a motion planning problem specification, planner, and set of possible constraint functions.

PARSEC Walkthrough. In lines 1-2, we attempt to create a successful plan for the problem as-specified using the initial state s_0, set of goal states $\bar{G}$, and known constraints K. If a viable plan is generated, PARSEC returns K as no additional constraints are necessary for completion. Otherwise, the algorithm begins the iterative, interactive process of constraint discovery. In line 3, PARSEC creates a tree (Fig 4.3) of available constraint functions and their potential parameter assignments (described later in Algorithm 6). This tree forms the basis for our query mechanism and leverages its inherent structural benefits for effective search. In line 6, $RequestExplanation()$ solicits the user for open-ended semantic feedback to describe how the skill is failing.

Line 7 refines the PARSEC Tree based on this feedback, scoring each node's estimated relevance to optimize the order in which they are used to form queries. To build intuition for how the PARSEC Tree is refined, consider the example in Figure 4.3, where a robot may be executing a pouring skill in the vicinity of a human by picking up a cup, moving it over a target area, then pouring out its contents. Given the feedback "The cup was too close", the [cup, distance] parameterization node in the middle of the tree would become the first node queried in line 10, skipping other nodes that might normally precede it ([objects], [cup, john], etc.). In lines 9-12, the robot asks if the node is relevant to the constraint they wish to add to the planner, iterating through the ordered list of the root's children until a relevant node is found (eventually returning with no

Algorithm 5: AskQuestion

Input: PARSEC Tree Node *node*
Output: Fully Parameterized Constraint Node

```
1  // node consists of a constraint function from C̄ and has parameters of types contained in
     S̄ from Alg. 1
2  // Ask user if the parameters indicated by this node are correct
3  response ← AskUser(node.parameters);
4  if response is "No" then return False; //Wrong node;
5  if node.children is ∅ then return node;
6  // node is relevant but not a leaf node: search deeper
7  for child in node.children do
8  |   ans ← AskQuestion(child);
9  |   if ans is "No" then continue;
10 |   else return ans;
```

solution if no node is identified by the human). The *AskQuestion*(*node*) call in line 10 initiates a recursive exploration down the tree until a leaf node is confirmed (indicating a fully parameterized constraint to add), described in Algorithm 5. Finally, lines 14-15 are responsible for adding the newly identified constraint and testing the skill to see if the new formulation is successful.

Inspired by the success of Bajcsy et al.'s method of incremental skill repair [149], our method is architected to focus on providing one new constraint with each iteration of the main while loop until a successful skill is produced.

4.3.2 PARSEC Tree Creation

The process for creating the PARSEC Tree (Fig. 4.3), used as the basis for query production, is described in Algorithm 6. The PARSEC Tree is a data structure with fully parameterized constraints at its leaves, meant to efficiently guide a user through the process of selecting a constraint function and values for its parameters. To interactively select a parameterized constraint, one could perform a depth-first search, asking if the contents of the current node are relevant: descending one level if 'yes', and moving laterally if 'no'. In PARSEC, we utilize semantic feedback from a human to reduce the number of nodes and levels of the tree, further reducing the level of effort required to select parameterized constraints.

The nodes of the PARSEC tree contain either parameter types (e.g., 'objects', 'humans',

Algorithm 6: CreatePARSECTree

Input: Set of potential constraint functions $\bar{C}$, Set of constraint function parameter types $\bar{S}$, Dictionary mapping parameter types to lists of valid assignments P

Output: Tree of parameter types, values, and parameterized constraints (Fig. 4.3)

```
max_args ← max(num_function_args(c) for c in C̄);
min_args ← min(num_function_args(c) for c in C̄);
tree ← Graph(); root ← Node('root');
tree.add_vertex(root);
for c in C̄ do
    if num_function_args(c) ≠ 0 then continue;
    tree.add_vertex(Node(c));
    tree.add_edge(root, Node(c));
prev_level ← [];
for s in S̄ do
    tree.add_vertex(Node(s));
    tree.add_edge(Node(root),Node(s));
    prev_level.append(Node(s))
for i in range(1,max_args+1) do
    cur_level ← [];
    param_combinations = List of all parameterizations (using P) of i-length elements from
      power set of arg lists in C̄;
    for p_combo in param_combinations do
        tree.add_vertex(Node(p_combo));
        cur_level.append(Node(p_combo));
        for p_node in prev_level do
            if p_node ⊂ p_combo or p_combo in P[p_node.name] then tree.add_edge(p_node,
              cur_level[-1]) ;
        for c in C̄ do
            if num_function_args(c) ≠ i then continue;
            // Add function c parameterized by p_combo
            tree.add_vertex(Node(c(p_combo)));
            tree.add_edge(Node(p_combo),Node(c(p_combo)))
    prev_level ← cur_level;
return tree;
```

'robots', 'distance', etc.), combinations of parameter values (e.g., 'cup', 'table', ['cup','table'] etc.), or parameterized constraint functions (e.g., above('cup','table')). In Algorithm 6, three inputs are required: a set of constraint functions ($\bar{C}$), a set of parameter types ($\bar{S}$), and a parameter value dictionary P mapping types (elements of $\bar{S}$) to a list of valid values for each.

The set $\bar{C}$ consists of all constraint function signatures available to the robot (e.g., ($\bar{C}$ = { above(object,object), below(object,object), min_distance(object,human,distance)}). $\bar{S}$ consists of

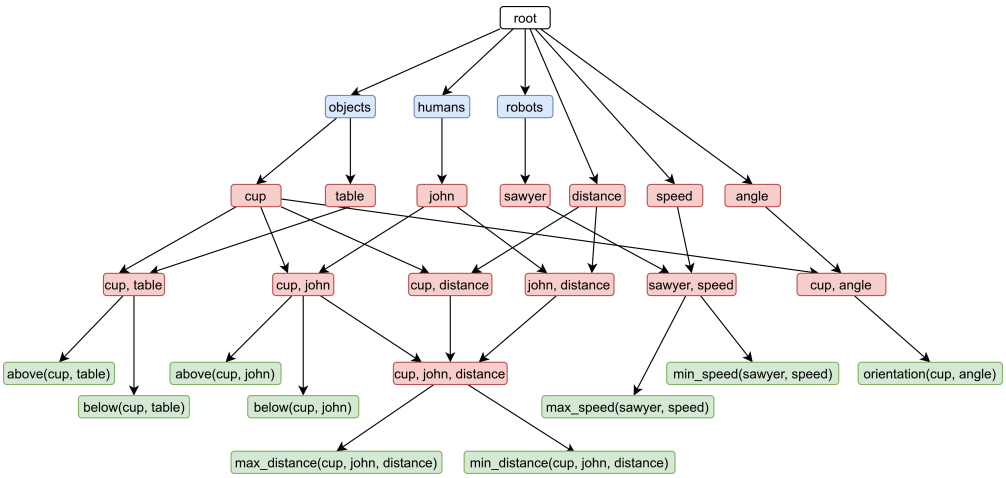


Figure 4.3: Example of a partial PARSEC tree. Blue nodes represent parameter types, red nodes represent both grounded (e.g., 'cup') and lifted (e.g., 'angle') parameters and combinations of parameters, and green nodes represent fully parameterized constraints. Lifted parameters are resolved to grounded values after they are assigned to a parameterized constraint (4.3.2).

all discrete parameter **types**, and is used to help logically cluster the parameter values within the tree. Finally, P provides all of the possible parameter values that the human could choose from, and is used to form the majority of internal tree nodes. To accommodate continuous valued constraint parameters within P, we add them as if they were each a single discrete parameter value (e.g., 'distance') and lazily ground them to specific values at the end of the constraint selection process.

Following Algorithm 6, nodes are organized such that each parent node is a subset of its children, where additional detail is added at each level of the tree until a fully parameterized constraint is reached at the leaves. We organize the tree to resolve parameter values down to constraint functions under the intuition that there will generally be many more possible constraint functions (which are robot-centric) than valid parameter types (which are environment-centric) within a given scenario.

While the PARSEC Tree is helpful in guiding users to specify valid constraints, the simple '20 Questions'-style depth-first search procedure described above is tedious and can be improved by utilizing the human for more than a simple Boolean signal.

4.3.3 Feedback Processing

During PARSEC, the robot asks the user to explain how the skill should be corrected. This open-ended explanation is parsed by a natural language processing (NLP) engine that scores each tree node in relation to the feedback given. In a practical sense, if the user's explanation refers to specific objects or attributes in the environment they will receive a higher score than objects that were not. For example, if the robot's domain knowledge consists of the words "computer" and "person" and the user gives the explanation "Don't move near the computer" the scoring process will give a high value to "computer" and a low value to "person". For our implementation, we used the Python Natural Language Toolkit (NLTK) [220] with the WordNet [221] lexical database.

Since $\bar{S}$ is the set of all parameter types and P contains all discrete parameter values and continuous parameter identifiers, then the total working dictionary for the robot is D where $D = \bar{S} \cup P$. Each element $d_i \in D$ is assigned a set of exact match words E_i and a similarity match words N_i. Processing a user explanation works by iterating over each word w_j in the explanation and assigning a value v_i to the each element $d_i \in D$ according to the formula:

$$v_i = \begin{cases} 1 & w_j \in E_i \\ sim(w_j, N_i) & w_j \notin E_i \end{cases}$$

Where $sim(w_j, N_i)$ is a function that returns the highest similarity score of the word w_j when compared to each word in N_i.

4.3.4 Node Relevance Scoring

The PARSEC algorithm can then utilize the scores given to each element of its working dictionary to score each node within the PARSEC tree. This is done using the scoring function:

$$score = \sum_{i=1}^{|\bar{V}|} v_i + \frac{\prod_{i=1}^{|\bar{V}|} v_i}{|\bar{V}|}$$

Where $|\bar{V}|$ is the number of parameters that the node encapsulates and $v_i \in \bar{V}$ is the value

given to each parameter by the feedback processing step. The intuition behind this function is that the summation has more value if a node consists of parameters with more positive scores. The product component will also be larger when the node's parameters contain non-zero scores, but it is discounted by the number of parameters to keep the tree iteration from diving too deep into the tree without high confidence in the parameters of the node. This scoring function is used by Algorithm 4 (Line 7) as **ScoreTree** to rank each node in the tree.

Additionally, if the above scoring function returns a value of 0 and $|\bar{V}| = 1$ (the node has only a single parameter), the score is set to a small positive value ϵ. This ensures that nodes near the top of the tree will be prioritized if no information is known (all other nodes have $score = 0$).

4.4 Evaluation and Results

We evaluated PARSEC using human feedback to provide corrective constraint annotations for a Rethink Robotics Sawyer robot (Figure 4.1) as it executed a collection of three representative manipulation planning tasks. As our algorithm is meant to expedite the constraint annotation process, the primary objective metric of this evaluation is the number of questions required to identify a constraint (and its parameterization) that allows the skill's motion planner to successfully plan and execute the desired behavior. This metric was chosen as a proxy for measuring the amount of time and effort expended by the user to correct faulty behavior from the robotic agent.

The three evaluation tasks are:

- **Handoff Task:** A handoff task where the robot attempts to give a cup to a user but spills the contents in the process of moving the cup towards the user. This scenario is an example of faulty training where the skill needs correction to repair a poor training. A constraint that keeps the cup upright will repair this skill.

- **Pouring Task:** A pouring task where the robot is tasked with pouring the contents of a cup into a receptacle sitting on a table. Though the skill has been properly trained, in this scenario the receptacle has been moved from its position during training and so the robot

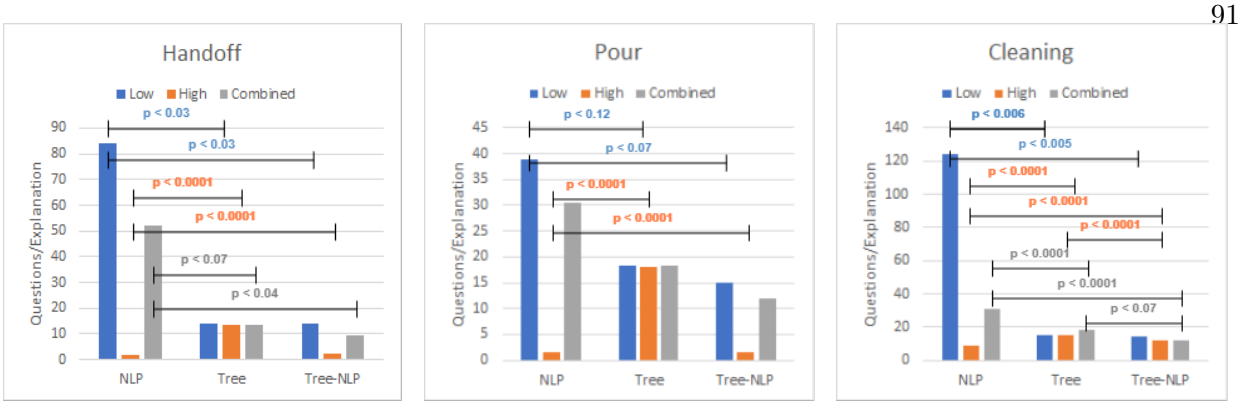


Figure 4.4: Results for all tasks, with user-provided explanations binned into low-quality, high-quality, and combined categories.

pours the contents in the wrong position over the table. This demonstrates a situation where the skill has been correctly trained but is not robust to changes in the environment. A constraint that requires the cup to be above the receptacle during the pouring motion will correctly augment this skill.

- **Cleaning Task:** A cleaning task where the robot moves a cup across the surface of a table in front of a user who is performing another task sitting at the table. The robot moves too closely to the user causing discomfort to the person. This scenario is an example of training that doesn't correctly represent the user's preferences for how the robot should act. A constraint that keeps the robot at a comfortable distance specified by the user will correct this skill.

4.4.1 Case Study Setup

We trained a Rethink Robotics Sawyer robot arm to perform the three three tasks described in 4.4. For each task, we trained the robot using Concept-Constrained Learning from Demonstration [201] both to successfully reach each skill's goal state as well as to fail in some aspect of the execution as described above (by removing an important learned constraint). We then recorded videos of the robot succeeding and failing in the skill execution so that we had examples of the faulty skill and correct skill in each example.

We then created a survey that requested the participant to view the videos of each skill and describe in a single sentence how the skill ought to be corrected. The participants could view the videos as often as they wished and had access to the videos of both the faulty skill execution as well as the correct skill. This way they could directly compare the faulty execution to the corrected behavior and so judge what specifically needed to be changed. The survey asked them to describe the correction as if they were talking to the robot.

We received 19 total responses to the survey with one response having to be discarded due to the participant not following the directions properly. Participants had varying levels of experience with robots, ranging from novice to expert. These 18 responses were used as the user feedback input for each algorithm.

We tested three approaches for the skill correction that utilized the explanations provided by users in the survey to correct the three faulty skills:

(1) **NLP Method**: Rank the available parameterized constraints (leaves of PARSEC tree) for the planner based on the scores returned by the scoring function then querying the user for the correct constraints by iterating through the sorted list.

(2) **Tree Method**: Run the PARSEC algorithm (Alg. 4) while omitting the tree scoring from line 7, setting $rankedTree$ equal to $tree$ from line 3. This explores the PARSEC Tree from the root node using Algorithm 5 but with no prioritization of the traversal ordering by the user's semantic feedback.

(3) **Tree-NLP Method**: Use the full Plan Augmentation and Repair through SEmantic Constraint (PARSEC) algorithm (Algorithm 4), leveraging the user's semantic feedback to accelerate traversal through the tree.

Our experiment set out to investigate the following hypotheses: **H1:** The number of queries in the PARSEC condition will be lower compared to the Semantic baseline (NLP method) and Naïve Exploration of PARSEC Tree method (Tree method), and **H2:** Naïve Exploration of PARSEC Tree

and PARSEC will perform better in comparison to NLP method (due to the structural benefit from the PARSEC Tree).

4.4.2 Results

We analyze each algorithm's performance on each task with the user feedback from our survey, calculating the number of questions asked before the correct constraint was discovered. Due to the large variety of potential PARSEC Tree constructions and the effect that node ordering would have on the results, each algorithm was run for 100 trials with the ordering of child nodes shuffled (before the ranking step) to account for any 'lucky' ordering effect of nodes with equal scores. This meant that that for each task we compiled a 100×18 table of data for each of the three tasks (18 from survey responses), where rows represent algorithm effectiveness per explanation and the columns represent the number of algorithm runs given that explanation. These tables can be averaged across the rows or columns to analyze different aspects of the results:

- Averaging across rows gives information about how each algorithm performs on the skill as a whole given the user feedback data. We call this type of averaging: **average by skill**.
- Averaging across columns gives information about how each algorithm performs on each explanation given by the users. This is informational in diagnosing high quality versus low quality explanations. We call this type of averaging: **average by explanation**.

For **average by explanation**, we did not observe any multimodalities in the distributed data but from the response we noticed that some of the users were descriptive about the recommendation (high quality explanation) and some users were vague about the failure and how should robot correct it (low quality explanation). This led to higher variance in the number of queries for repair (especially in the NLP condition). Therefore, to get the better insight we segmented our user provided explanations into three bins: 1) high quality explanations, 2) low quality explanations, and 3) combined (all the explanation). We conducted an ANOVA to test effects across our three algorithmic approaches for the average number of queries for each task based on the quality of

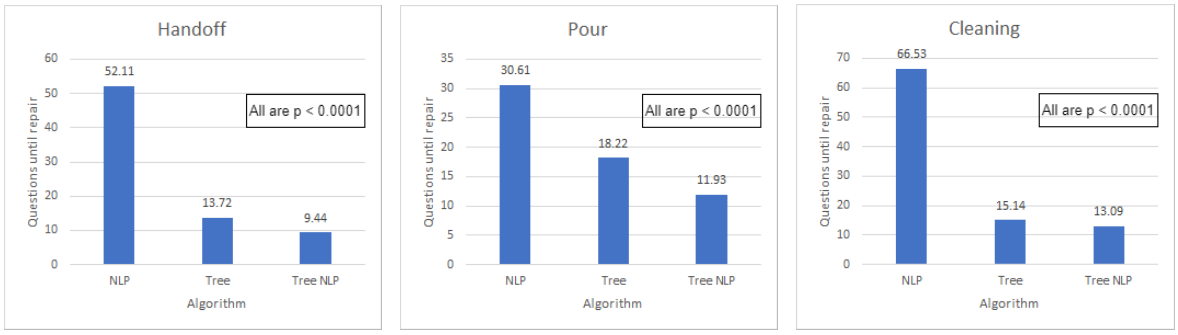


Figure 4.5: Results for all three evaluation domains, showing average performance by each method. The full PARSEC algorithm (Tree-NLP) provides a consistently more efficient experience in each task.

explanation. For the handoff task, we found a significant effect from the PARSEC algorithm on number of queries for combined explanations ($F(2, 51) = 3.91, p < 0.03$), **confirming H1**. Post-hoc comparisons using Tukey's HSD test (Figure 4.4) revealed that using PARSEC tree resulted in a significant different level of queries, **confirming H2**. Similarly, we can see that both high quality explanations ($F(2, 18) = 145.83, p < 0.0001$) and low quality explanation ($F(2, 30) = 5.39, p < 0.01$) **validate H1 and H2**.

The significant effect was observed for the cleaning task using the combined explanations ($F(2, 51) = 7.3, p < 0.002$) and post-hoc comparison show similar results as the handoff task (i.e., using PARSEC tree resulted significant different level). Similarly, observation can be seen effectiveness of PARSEC tree from low quality explanations ($F(2, 24) = 15.8, p < 0.0001$). The high quality explanations show statistical significance ($F(2, 24) = 152.8, p < 0.0001$) for number of queries and Tukey's HSD test gives significant different level for each of the approaches. No significant effects were found with respect to combined explanation for the pouring task measure of number of queries (p = 0.1) but we found the significant effect of the our algorithm for high quality explanations ($F(2, 9) = 10379, p < 0.0001$) and Post-hoc analysis reveal statistical significant different levels because of the PARSEC tree. Results for low quality explanation ($F(2, 39) = 3.12, p < 0.06$) are inconclusive (Figure 4.4) but merit further investigation to confirm an effect.

Likewise, for **average by skill** results, we conducted an ANOVA to investigate differences between our three algorithmic approaches for the handoff task, cleaning task, and pouring task. For

the handoff task, significant effects were found from the usage of PARSEC algorithm on number of queries ($F(2, 297) = 3362.48, p < 0.0001$) and Tukey's HSD test (Figure 4.5) reveals a significantly different level of queries for each approach. Outcomes were similar for the cleaning ($F(2, 297) = 2135.47$, $p < 0.0001$) and pouring tasks ($F(2, 297) = 454.24$, $p < 0.0001$) further **validating H1 and H2**.

4.5 Conclusions

In this chapter, we present **Plan Augmentation and Repair through SEmantic Constraints** (PARSEC), a new algorithm that enables novice robot users to quickly correct faulty behavior or apply personal preferences to a robot skill through a process informed by a NLP accelerated semantic hierarchy of queries. Our results show that PARSEC reduces the number of queries the user is required to answer before the skill is corrected as compared to a baseline algorithm only applying semantic rankings to constraints and a baseline algorithm that utilizes hierarchical structure to direct queries for resolving desired parameterized constraints. In demonstrating the benefit of combining the PARSEC Tree's hierarchical structure alongside a semantic analysis of the user's feedback, we contribute a novel method for human-in-the-loop skill learning that merges human-robot interaction and constrained motion planning.

Our primary result shows that PARSEC reduces time spent by users across three representative manipulation tasks, each demonstrating a different application domain: correcting a skill with faulty or incomplete training (handover task), augmenting a skill to perform in a novel environment (pouring task), and adapting a skill to user preferences (cleaning task). These results show the ability for PARSEC to be applied to various types of robotic skills while at the same time providing an efficient way for novice users to correct and adapt the robot's behavior to their own preferences.

Chapter 5

Multimodal Decision Support via Mental Model Alignment and Justification

"The limits of my language mean the limits of my world."

— Ludwig Wittgenstein, *Tractatus Logico-Philosophicus*

This chapter and the next focus on enabling multimodal decision support by leveraging visual and natural language explanations in human-machine teaming through mental model alignment and justification. The focus of this chapter is on exploring the role of visual explanations in multi-agent reinforcement learning (MARL) under uncertainty to achieve shared situational awareness, improve teaming and transparency, and influence human teammates' behavior. Semantic explanations presented in Chapter 3 are not well-suited for certain scenarios, especially those involving high uncertainty, which require the portrayal of multiple competing hypotheses as plans change based on new observed information (i.e., partially observable domains). For these continually evolving domains, visual information representation is ideal [222], motivating our subsequent work on AR-based visual guidance called MARS (Min-entropy Algorithm for Robot-supplied Suggestions) [193].

In this chapter, we first introduce characterizations of and generative algorithms for two complementary modalities of visual guidance: prescriptive guidance (visualizing recommended actions), and descriptive guidance (visualizing state space information to aid in decision-making). Robots can communicate this guidance to human teammates via augmented reality (AR) interfaces, facilitating synchronization of notions of environmental uncertainty and offering more collaborative and interpretable recommendations. We also introduce a min-entropy multi-agent collaborative planning algorithm for uncertain environments, informing the generation of these proactive visual

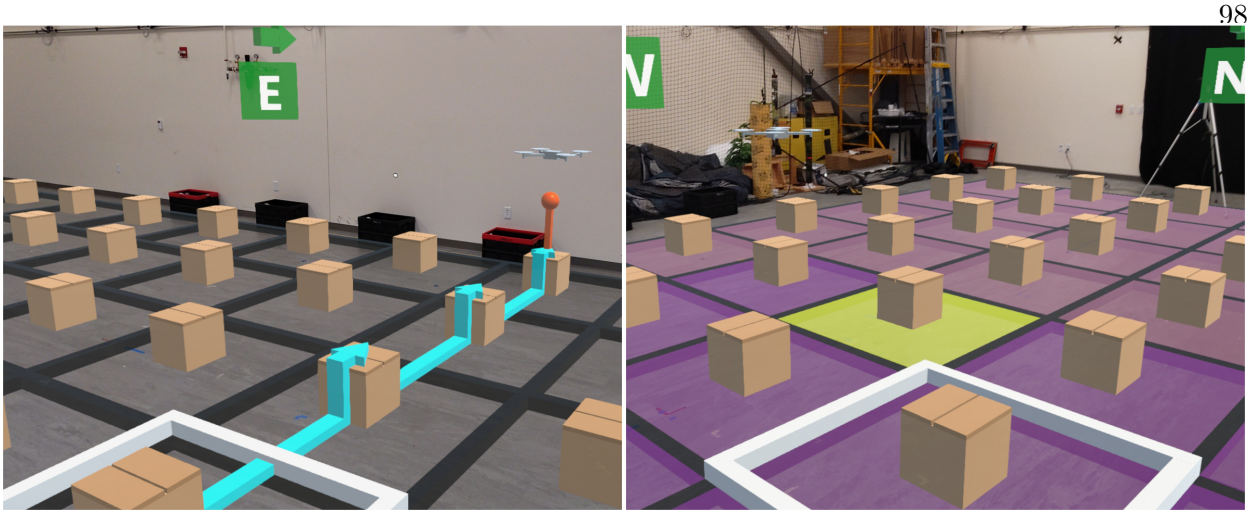


Figure 5.1: AR-based interfaces for prescriptive (Left) and descriptive guidance (Right) in the Minesweeper domain. In the prescriptive condition, suggested moves are shown as cyan arrows between grid squares, with suggested defuse actions indicated by the orange pin (underneath the virtual drone teammate). In the descriptive condition, grid squares are colored as a heatmap, representing the probability for each square containing a hidden mine as judged by the drone, from dark purple (low) to bright yellow (high).

recommendations for more informed human decision-making. We illustrate the effectiveness of our algorithm and compare these different modalities of AR-based guidance in a human subjects study involving a collaborative, partially observable search task. Finally, we synthesize our findings into actionable insights informing the use of prescriptive and descriptive visual guidance.

5.1 Introduction and Motivation

When a team is tasked with solving a problem in an uncertain environment, it is vitally important to keep notions of that uncertainty, as well as the problem-solving strategy, synchronized between teammates as this information changes over time, in order for each teammate to act in a coordinated fashion. In this work, we explore this challenge as it relates to human-robot teaming. Autonomous agents are well-equipped to plan over probabilistic state spaces, updating their probability models in response to new observations, and choosing optimal actions in response to this new information. We hypothesize that visually communicating this knowledge to human teammates efficiently improves team performance.

Consider a search and rescue task with human and robot teammates coordinating to locate a

victim: this is an inherently stochastic environment, where the likelihood of finding a victim varies location to location, characterized by a probability mass function (PMF). As the human and robot teammates cover more ground with their search, that PMF continually updates in response to the agents' observations. Since the robot agents are maintaining an up-to-date PMF to plan over, they can also communicate it to their human counterpart to keep them in the loop, a modality we call **descriptive guidance** (synchronizing state space information to aid in human decision making). Additionally, the robots can use that PMF combined with a model of their human counterpart's action space to directly recommend next actions to the human, a modality we call **prescriptive guidance**.

In this work we use a 3D collaborative analogue of the PC game Minesweeper, played using an augmented reality (AR) headset, to serve as an experimental domain reminiscent of real-world spatial navigation and search tasks. For this game, we tasked a human-drone team with locating and defusing a number of mines hidden throughout a grid of cardboard boxes projected onto the floor of an experiment space (Fig. 5.1). The drone can navigate the environment, taking measurements with a noisy sensor to attempt to determine whether a box contains a hidden mine. The human must also physically navigate the environment, taking time to search boxes and defuse mines whenever they think they've located one.

To assist the human in their task, we developed an algorithmic framework for multi-agent collaboration under uncertainty, capable of generating prescriptive and descriptive visual guidance for the human teammate as the drone explores the environment. We also developed AR interfaces for each type of drone-provided guidance, with arrows and pins indicating suggested moves under the prescriptive modality, and a heatmap overlaid onto the environment representing the PMF under the descriptive modality (Fig. 5.1).

We conducted a human subjects study using this collaborative Minesweeper task, varying which modality of guidance participants saw between conditions as they attempted to locate and defuse all hidden mines as quickly as possible: prescriptive guidance (the 'arrow' condition), descriptive guidance (the 'heatmap' condition), and a combination of both (the 'combined' condition).

This study served to validate our algorithm in a live human-robot teaming setting with environmental uncertainty, helping to assess the benefits and drawbacks of each type of visual guidance through a variety of objective and subjective measures.

We characterize the core contributions of this work as follows:

- A **characterization of** and **method for** generating AR-based prescriptive and descriptive visual guidance, communicating environmental uncertainty and providing actionable recommendations to human teammates in joint human-robot tasks.
- An empirical validation and analysis of the effectiveness of prescriptive and descriptive visual guidance through a human subjects study involving a collaborative search task with an autonomous robot.

5.2 Background and Related Work

Visual Guidance & Augmented Reality Interfaces. Visualization is frequently used in human-robot teaming for tasks such as environmental navigation, search and inspection, and fault recovery [223, 224, 225]. The visualization of task and environment data enables human teammates to develop new insights into the problem being solved and heightens their situational awareness, aiding in decision-making [226]. Gale et al. demonstrated the effectiveness of playbook-based visual interfaces to allocate roles and responsibilities between human-automation systems in an unmanned aircraft system (UAS) swarm support task [227]. Ahmed et al. successfully utilized a visual sketching interface to fuse the data of multiple noisy 'human sensors' in cooperative search missions with autonomous vehicles, further demonstrating the utility of visual information transfer in human-robot teaming [228].

Visualization is particularly useful for communicating uncertainty. Bhatt et al. explored methods for assessing and displaying uncertainty in models, communicating it to stakeholders to assist in trust-building and decision making. [229]. Furthermore, Colley et al. showed that visualizing the internal information of autonomous vehicles improves trust and situational awareness

[230]. As these works focus on the communication of internal model-based uncertainty in human-robot teaming, we apply the same concept to external environment-based uncertainty associated with unexplored terrain.

Recent work on augmented reality-based interfaces has shown that providing in-situ visualizations with an AR headset can greatly improve the efficiency of human-robot teaming [231, 232]. Fraune et al. investigated the use of mixed reality interfaces for humans monitoring and commanding drone teams for search and rescue [233]. Kunze et al. show the effectiveness of AR to visually communicate uncertainty during automated driving [234].

Explainable AI & Shared Mental Models. Recent research in model reconciliation and knowledge sharing in human-robot teams has shown the importance of explainability and mental model synchronization to improve trust, transparency, and team performance [12, 22]. Furthermore, explainable AI (xAI) can help complex models become more understandable by human teammates, allowing for faster debugging when unexpected behaviors or failures occur [7, 92]. Visualization is a common modality for presenting explanations through xAI [235]. Visual information presentation is ideally suited to explanations that are complex, long, re-referenced, and which involve uncertainty or noise [222]. Therefore, visualization is often used to aid in the interpretation of complex models, showing how model parameters affect final classification decisions (e.g., in local approximation methods such as SHAP [236], model-agnostic methods such as LIME [7], and saliency map methods such as Grad-CAM [29]).

Other recent studies have utilized case-based explanations as visualizations to expose overconfidence in models and visualize class boundaries [168]. A related technique is visual counterfactuals [237, 238] (showing how an input must change to change the classification of the output). These techniques are typically utilized post-hoc by AI experts to debug models [30, 31]. Our visual guidance methodology on the other hand assumes very little domain knowledge, leverages an AR-based interface for more user friendly visualization, and is usable in live human-robot teaming scenarios.

5.3 Algorithmic Approach

In this section, we introduce a novel algorithm for multi-agent collaboration under uncertainty using min-entropy online reinforcement learning called MARS (Min-entropy Algorithm for Robot-supplied Suggestions).

Our algorithm assumes the presence of two classes of agents: exploration agents (agents who can move through the environment and take observations) and active agents (agents who can directly affect environment state through taking actions). This divide between agents with differing goals and action spaces is typical in human-robot teaming domains. For example, a common search and rescue practice involves an initial search phase conducted by an aerial vehicle, with ground rescue or airlift units deployed to extract targets once their locations are determined. In this work, we explore the case where the active agent is human and the exploration agents are autonomous.

5.3.1 Multi-Agent Entropy Minimization

The core insight behind this algorithm is that environmental uncertainty over task-relevant variables can be succinctly characterized by probability density distributions, a common practice in search and rescue operations [239, 240, 241]. We use the multivariate probability mass function (PMF), a discrete version of this concept, to model environmental uncertainty as it changes over time. This PMF serves as a shared utility function between all agents in our formulation for min-entropy collaborative planning, allowing for solving a single Markov Decision Process (MDP) with the PMF as its utility function. Furthermore, this PMF can be communicated to human teammates in order to provide insight into the autonomous agents' policy which we detail in Section 5.4.

The collaborative task can be formulated as a single MDP M_R, over which one or multiple exploration agents maximize their expected reward. M_R is defined by the 4-tuple: (S, A, T, R):

- S is the finite set of discrete world states consisting of traditional "world features" W (e.g., agent positions) along with "distance features" D that encode pairwise distances between all agents in the collaborative task (including the human teammate), using an appropriate

distance metric for the task being solved. A finite set of distance features is given by $D = \{d_{12}, d_{13}, \cdots, d_{(N-1)N}\}$, such that d_{12} represents the distance between agent 1 and 2, and so on. $|D| = \binom{N}{2}$, where N is the total number of agents in the collaborative task.

$$S = \begin{bmatrix} W \\ \\ D \end{bmatrix}, W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \end{bmatrix} D = \begin{bmatrix} d_{12} \\ d_{13} \\ \vdots \\ d_{(N-1)N} \end{bmatrix}$$

- A is the set containing all N-tuples representing the product of all possible exploration agent joint actions.
- $T : S \times A \to \Pi(S)$ is the state-transition function describing the model's state transition dynamics.
- $R : S \times A \times S \to \mathbb{R}$ defines the expected immediate reward gained by the agent for taking an action $a \in A$ in a state $s \in S$ and transitioning into the next state $s' \in S$.

We solve this single MDP M_R via online reinforcement learning to get an optimal policy π_R^* for each autonomous agent using a joint PMF as a reward function given by:

$$R(s, a, s') = \alpha(0.5 - |0.5 - pmf(s')|) + \beta \sum_{n \in N} d_n - 1 \tag{5.1}$$

In Equation 5.1, α and β are tunable hyper-parameters, and $pmf(s')$ is the value of the probability mass function at state s', representing the probability that s' contains a desired goal or target. The first term of Equation 5.1 encourages the exploration of states with higher uncertainty (PMF values close to 0.5), minimizing entropy over time as those states are observed. The second term maximizes distance from other agents, maximizing coverage over the state space for faster learning. Each agent's reward function is affected by the current PMF, which is updated every time agents observe a new state in the environment according to Bayes' rule. Therefore, the MDP should be re-solved whenever the PMF updates, in order to minimize the entropy of the distribution over task-relevant latent state information.

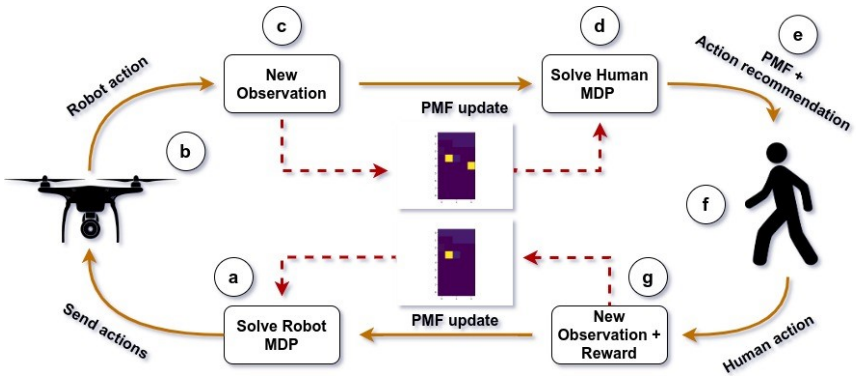


Figure 5.2: Algorithmic flow: a) the robot's MDP is solved, parametrized by the PMF, and actions are sent to all agents, b) the robot takes an action and c) observes a new potential mine, updating the PMF (the new mine is visible as the righmost yellow square), d) the updated PMF is used to solve the human recommendation MDP, e) the resulting PMF and action recommendations are sent to the human, who f) views the guidance via an AR interface, and takes an action, defusing the mine, g) the new observation and reward update the PMF again (the new mine has been defused, removing the yellow from the heatmap)

5.3.2 Generating Assistive Guidance

Here we present our approach for generating assistive guidance for human teammates in uncertain environments. Similarly to section 5.3.1, we can model a human agent's behavior using an MDP with the PMF as its utility function. The MDP M_H is likewise defined by a 4-tuple (S, A, T, R), where:

- S is the finite set of world states consisting of traditional "world features" W, along with the expected number of goals left ("**goals_left**"), and the latent boolean variable **is_goal** $\in \{0, 1\}$ with **is_goal** $= 1$ indicating a goal is present.

$$S = \begin{bmatrix} W \\ goals\_left \in \mathbb{N} \\ is\_goal \in \mathbb{B} \end{bmatrix}$$

- A is the set of possible task-relevant human actions.
- T and R are similarly defined as seen in Section 5.3.1.

Our reward function distinguishes between two classes of actions: exploration and goal-centric

actions. Exploration actions are geared towards navigating between states to minimize uncertainty or reach a state containing a goal. In comparison, goal-centric actions are conducted within a state and contribute towards task completion (e.g., signaling for pickup in SAR domains).

The reward function for a human agent exploration action is given by:

$$R(s, a, s') = pmf(s') - \beta * is\_goal_s - penalty \tag{5.2}$$

where,

$$penalty = 1 - \alpha * goals\_left$$

The first term of Equation 5.2 provides the immediate reward from the next state s', the second term encodes a negative reward for ignoring a goal in the current state s, and the penalty term provides long term incentive to achieve the desired task objectives as quickly as possible. α and β are tunable hyper-parameters. We can expand Equation 5.2 to get the expected immediate reward as follows:

$$\begin{aligned} \mathbb{E}(R) = (1 - pmf(s)) * (pmf(s') - penalty) \ + \\ pmf(s) * (pmf(s') - penalty - \beta) \end{aligned} \tag{5.3}$$

The reward function for a human agent to take goal-centric actions is as follows:

$$R(s, a, s') = \beta * is\_goal_s - penalty \tag{5.4}$$

The first term of equation 5.4 provides the immediate reward if a goal is present in the current state s, and the rest of the terms are defined the same as in Equation 5.2. Expanding Equation 5.4, the expected immediate reward is:

$$\begin{aligned} \mathbb{E}(R) = pmf(s) * (\beta - penalty) \ - \\ (1 - pmf(s)) * penalty \end{aligned} \tag{5.5}$$

Solutions to this MDP M_H can be used to obtain policy recommendations for a human agent.

Algorithm 7: Min-entropy Algorithm for Robot-supplied Suggestions (MARS)

Input: Robots' MDP M_R (S, A, T, R), Human's MDP M_H (S, A, T, R), R_h, Current Robots State $\overline{S_R} = \{s_1, s_2, \cdots, s_{n-1}\}$, Current Human State s_h, Num. rollout k, Prior P

1 $pmf \leftarrow P$; // Initialize pmf with prior
2 **while** s_h **is not a terminal state do**
3 $\quad \pi^*_R \leftarrow solve\_policy(M_R, pmf)$;
4 $\quad \overline{A_R} \leftarrow \pi^*_R(\overline{S_R})$; // Get optimal actions for each robot
5 $\quad \overline{S_R} \leftarrow send\_actions(\overline{A_R})$; // Send optimal actions
6 $\quad pmf \leftarrow update\_pmf(\overline{S_R})$; // Get observations
7 $\quad \pi^*_H \leftarrow solve\_policy\_human(M_H, pmf)$;
8 $\quad \overline{A_H} \leftarrow rollout(\pi^*_H, s_h)[:k]$; // Get actions for human
9 $\quad recommend\_action(\overline{A_H}, pmf)$
10 $\quad s_h, r_h \leftarrow observe\_human\_action()$
11 $\quad pmf \leftarrow update\_pmf(r_h)$

5.3.3 Algorithm

In this section, we outline the details of MARS, as presented in Algorithm 7. We ground the algorithm with an example task inspired by Minesweeper, involving a single human agent and a single robotic drone. The goal of the task is to locate and defuse a number of mines hidden throughout a grid-based environment without unintentionally detonating them. Although only the human teammate is capable of defusing mines, the drone has a noisy sensor capable of determining whether the grid square it is currently flying over contains a hidden mine, parameterized by a false positive and false negative rate. If the human teammate leaves a square containing a mine without defusing it, it detonates, providing a substantially negative (non-terminal) reward for the episode.

Before the task begins, the PMF is initialized with a prior to provide an initial heuristic (Line 1). If there is no information with which to seed a prior, a uniform PMF can be used at this step. An optimal policy can then be computed using the prior PMF and the robots' MDP M_R. Based on the learned policy π^*_R, optimal actions are sent to all robots (Lines 3-5). Once the robots execute these actions, they obtain new observations from the environment and update the PMF using Bayes' Rule (Line 6). In the Minesweeper example shown in Figure 5.2, step c shows the resultant PMF after the robot takes an action and obtains a new observation.

Given this updated PMF, the human agent's policy π^*_H is computed and a k-step rollout is

used to provide action suggestions for the human (Line 7-8). The number of steps k determines how many actions into the future will be recommended to the human teammate, which can be chosen depending on the nature of the task. For the Minesweeper example, we provided suggested actions up to and including the first recommended "defuse" action (step e in Figure 5.2). These actions $\overline{A_H}$ and the updated PMF are provided to the human agent as guidance (Line 9), the visualization of which is discussed in Section 5.4. Next, the human action is observed, the reward r_h is recovered from the environment, and the PMF is updated again in response (Lines 10-11).

5.4 AR-based Visual Guidance Design

The PMF and action recommendations meant to be communicated to the human agent are particularly well-suited for visual presentation in the Minesweeper domain, but this will vary by task. For the Minesweeper domain, we developed a set of AR visualizations geared toward environment navigation and search tasks. An AR headset-based interface was chosen due to its hands-free nature and its ability to present information in-situ, as holograms projected in environmental context aid in the efficiency of information uptake.

We generalize the proposed AR-based visual guidance into two categories, corresponding to the two data products of Algorithm 7. First is prescriptive guidance, in which sequences of actions are directly suggested to the human based on the algorithm's current recommendations. Second is descriptive guidance, where state space information is presented to the human in the form of the current PMF to support decision making.

5.4.1 Prescriptive Guidance

The essence of prescriptive guidance is directly suggesting to a human teammate what they should do next. In tasks involving physically navigating through space, like search and rescue or the Minesweeper experimental domain, movement suggestions can be represented as holographic arrows projected onto the ground, extending from the human's current location to their next suggested waypoint (Fig. 5.1 Left), an AR visualization technique which has shown effectiveness

for navigation tasks [242].

This arrow-based guidance is straightforward to understand and requires little mental effort to follow. However, since the recommendations are presented without rationale, they require a degree of trust from the human teammate if they are to be followed, which may or may not be warranted depending on the performance of the autonomous agents under environmental uncertainty. This uncertainty may also lead to frequent changes in the path recommendations, deflecting the arrows and causing confusion on the part of the human teammate as the old guidance is discarded.

5.4.2 Descriptive Guidance

In contrast to explicit action recommendations, descriptive guidance involves providing state space information with which human teammates can make their own decisions. For spatial navigation tasks like the Minesweeper domain, the current PMF can be projected onto the environment itself, dividing the space into discrete regions and coloring those regions as a heatmap (Fig. 5.1 Right). In the Minesweeper domain, dark purple is used to represent a low chance of a region containing a goal while bright yellow is used to represent a high chance, with intermediate probabilities colored on a gradient between purple and yellow. Since decision-making in the Minesweeper domain relies more on discrimination between PMF probabilities close to 0 than probabilities close to 1, the heatmap is generated using a logarithmic color scale, a technique used to visually bring out finer distinctions towards the low end of a scale with an uneven distribution [243].

This descriptive guidance acts as a decision support tool, providing the human with information which they can use however they see fit. In contrast to the prescriptive arrows, this type of guidance is highly transparent. On the other hand, it is more cognitively demanding, requiring the human to actively plan ahead, thereby reducing its effectiveness in domains with large and complicated state spaces or domains with time pressure.

5.5 Experimental Validation

We evaluate the utility of the AR-based visual guidance modalities presented in Section 5.4 within a partially observable environment involving live human-robot teaming, utilizing the proposed multi-agent entropy minimization algorithm. These results were obtained through a human subjects study using our collaborative Minesweeper-inspired domain.

5.5.1 Experimental Design

We use a 3×1 within-subjects experiment to evaluate three different varieties of AR-based visual guidance: 1) prescriptive guidance, or the 'arrow' condition, 2) descriptive guidance, or 'heatmap', and 3) a combination of prescriptive and descriptive guidance, or 'combined' (Figure 5.3). A within-subjects design was chosen to obtain direct, grounded comparisons between visualization types from participants. The guidance was visualized through a Microsoft HoloLens 2, overlaid onto a rectangular grid of cardboard boxes on the floor of the experiment space.

The orderings of the 'arrow' and 'heatmap' conditions were randomized and fully counterbalanced between participants. Since the 'combined' condition relied on the prior introduction of both modalities independently, it was ordered last. As participants played three rounds of the game with differing conditions, three environment maps were created, each with the same number of hidden mines, located on different squares. We blocked participants to match experimental conditions to environment maps using a balanced Latin square design to achieve partial counterbalancing and minimize ordering and learning effects [244, 245]. The Latin square resulted in blocks of size six differing in the ordering of the 'arrow' and 'heatmap' conditions, and in the matching of environment map to condition. Participants were randomly assigned to one of these six permutations.

5.5.2 Hypotheses

Through a human subjects study, we evaluate five visual guidance hypotheses partitioned into three categories:

H1: Subjective Hypotheses

H1.a: Participants will find the combined guidance to be more trustworthy than descriptive or prescriptive guidance, as transparency of recommendation leads to more trust [20, 187].

H1.b: Participants will find the combined guidance to be more interpretable, informative, and helpful for decision-making compared with the other conditions.

H1.c: Participants will find the combined and prescriptive guidance conditions to be less stressful and demanding compared with descriptive guidance, due to the presence of clear recommendations.

H2: Performance Hypothesis

H2: Participants will take less time to solve the task when given combined or prescriptive guidance compared with descriptive guidance, since they can reduce thinking time by leveraging direct algorithmic guidance.

H3: Independence Hypothesis

H3: Participants will act with more independence and deviate more frequently from the prescribed path in the combined condition compared with solely receiving prescriptive guidance, as they can utilize the added descriptive information to take their own initiative when they perceive suboptimality in robot suggestions.

5.5.3 Rules of the Game

Each round, participants attempted to solve the Minesweeper puzzle by successfully locating and defusing all four mines hidden throughout the 9×5 grid of cardboard boxes as viewed through the HoloLens headset. Each turn, participants had four options for movement actions: "Go North", "Go South", "Go East", and "Go West", each of which moves a single square in the respective direction. If the participant suspected a square contained a hidden mine, they could take a fifth action: "Defuse", which opened the box on the square they were currently standing on, revealing whether it was empty or contained a mine, which they had now successfully defused (Fig. 5.3). If they moved from a square containing a mine without defusing, the mine would be uninten-

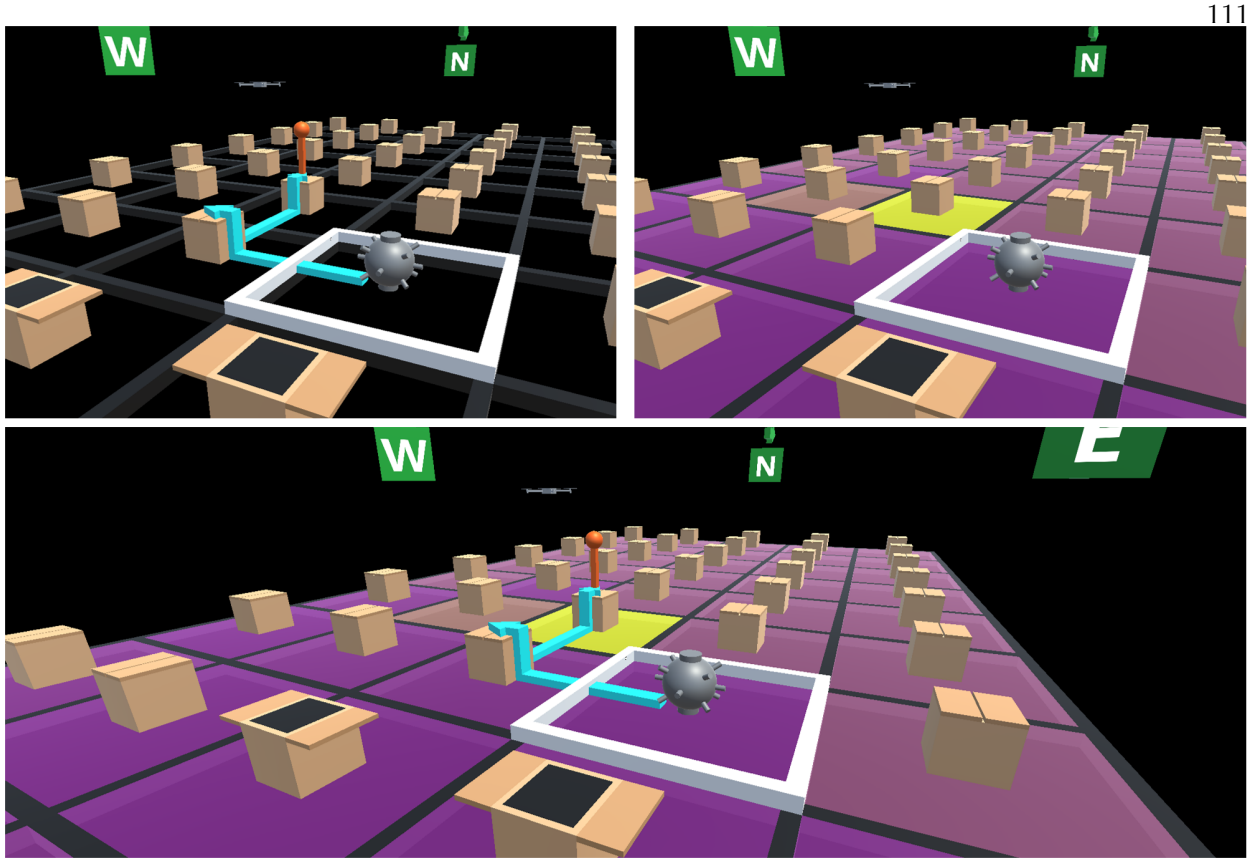


Figure 5.3: The three experimental conditions. A white square marks the user's current location where they have defused a mine. Top-left: 'arrow' condition, Top-right: 'heatmap' condition, Bottom: 'combined' condition.

tionally detonated. Unlike Minesweeper, this did not end the game; participants were simply told beforehand that this would contribute to a low score.

As the participants moved through the grid, a virtual drone teammate concurrently explored the grid autonomously, providing assistive guidance in a format dictated by the experimental condition. After the participant took a turn, they waited briefly for the drone to take theirs. The drone could move faster than the human teammate, moving three squares for every human action and using its noisy mine-detection sensor on every square it flew over. However, the drone was incapable of defusing or otherwise interacting with the mines; only the participant could do that. The human and drone teammates alternated turns until all four mines had been successfully defused or unintentionally detonated.

5.5.4 Study Protocol

Upon providing informed consent, participants were educated on the overall rules of the game through alternating phases of reading an illustrated instruction manual and reviewing it with an experimenter to reinforce the ideas. To minimize potential learning effects, participants were given a brief practice round (without visual guidance) using the HoloLens to ensure that they acclimated to the AR interface and became comfortable exploring the environment and issuing commands, trying every action at least once. Participants were told about their drone teammate, including information about the drone's capabilities and limitations, namely its uncertain sensor. This served to ensure participants would not be overly confused if they saw the drone's guidance change during the experiment round.

Participants began their first experimental round with randomized condition and environment map. They were first shown a page in the instruction manual describing the form of guidance they would be receiving that round. They then donned the HoloLens and played the round, taking actions and navigating the experiment space until all four mines had been defused or unintentionally detonated. After finishing the round, participants removed the HoloLens and returned to the staging area to complete a post-round survey. These steps were repeated twice more for the other experimental conditions. Following the third post-round survey, participants completed a final post-experiment survey and an exit interview.

5.5.5 Implementation Details

Three environment maps with different locations for the four hidden mines were selected to be of similar difficulty and similar optimal solving time. Each round, the virtual drone's actions were controlled by our algorithm running on a laptop (Intel(R) Core i7-10870H CPU @ 2.20GHz) and broadcasted turn-by-turn via a ROS publisher to the HoloLens. The drone's guidance each round was similarly computed by our algorithm and broadcast to the HoloLens using ROS. Each turn, the drone took three steps to mimic the relative speed of aerial robot navigation over human

navigation. The drone observed every square it flew over, even observing some squares more than once, using a simulated noisy sensor with a 10% false-positive rate and a 1% false-negative rate to determine whether a hidden mine is present on that square, adding uncertainty into the drone's recommendations. We chose to use a single drone for our experiment since our domain was small and adding more autonomous agents would lead to quicker convergence towards optimal guidance, causing a more deterministic interaction with participants. The robot's MDP and the human recommendation MDP were solved online each turn using policy iteration.

In the prescriptive 'arrow' condition, our algorithm sent action suggestions every turn up to and including the next suggested "Defuse" action to the AR interface. In the descriptive 'heatmap' condition, our algorithm sent the updated PMF every turn, shown as a heatmap from dark purple for low values to bright yellow for high values, interpolating logarithmically for intermediate values. Each turn, participants selected their action via their choice of voice control (comprising 69.3% of all 1597 recorded moves), or menu-based hand control (30.7% of recorded moves).

In all three environmental maps, there was the possibility for certain scenarios we dub "switchbacks" where participants will turn around and double back on their previous state if they follow the drone's updated prescriptive arrow. These scenarios are an emergent behavior when the participant is located immediately between two potential mine locations, whether they are actual mines or false positives. The drone simply updates its path based on new information and reward maximization, but its behavior is often perceived as suboptimal from the perspective of the human teammate. We observed how participants responded to these switchbacks, especially as they differed based on guidance condition.

5.5.6 Measurement

We had 19 participants (12 males, 7 females) in our IRB-approved study, ranging in age from 18 to 37 ($M = 25.42; SD = 4.76$). We used a number of subjective and objective measures to evaluate our algorithm and the AR-based visual guidance.

For subjective metrics, we administered post-round questionnaires to participants for each

condition to get immediate impressions. These surveys consisted of 7-point Likert-scale items derived from questions from established questionnaires in the robotics and explainable AI community, geared at trust and reliability [107, 246], interpretability and decision-making [106, 107], and stress and workload (NASA-TLX) [190]. From these items, we were able to identify three concepts: **Trust, Interpretability, and Mental Load**.

The **Trust** scale consists of 4 items: confidence, reliability, trust, and intelligence (Cronbach's $\alpha = 0.90$). **Interpretability** consists of 4 items: decision-making power, adaptability, informativeness, and sufficiency (Cronbach's $\alpha = 0.89$). **Mental Load** consists of 2 items: stress and cumbersomeness (Cronbach's $\alpha = 0.84$).

Following the last round of the experiment, participants compared each of the three guidance types they received. Participants ranked each guidance type relative to one another in terms of **trust**, **usefulness**, **helpfulness for decision making**, and **confidence**.

For objective metrics, we recorded the following items for each experiment round: **Total Moves** (the total number of moves needed to solve the puzzle), **Total Time** (the total time needed to solve the puzzle, in seconds), **Time per Move** (the average time per move, in seconds), and **Compliance Rate** (the percentage of moves taken matching the recommendation provided by the system, only applicable for the 'arrow' and 'combined' conditions).

5.6 Results and Discussion

5.6.1 Analysis

5.6.1.1 Subjective Analysis

We analyzed both the post-round survey scales and post-experiment comparison results to test our subjective hypotheses. The post-round Likert scale data suffered from a significant ceiling effect, where many participants rated all guidance types highly, using primarily 6s and 7s out of a maximum score of 7. For this reason, we transformed the raw Likert scores into rankings, giving for each survey item the participant's preference ordering between the three guidance types, with

any ties receiving equal ranks. We analyzed both this ranked scale data and the ranks from the post-experiment survey's comparison questions using a nonparametric Kruskal-Wallis Test with experimental condition as a fixed effect. Post-hoc comparisons used Dunn's Test for analyzing guidance type sample pairs for stochastic dominance.

We found a significant effect in favor of the 'combined' condition over 'arrow' for the **Trust** scale ($H(2) = 8.26, p = 0.016$). Post-hoc analysis with Dunn's Test found that participants consistently preferred 'combined' ($M = 2.68), p = 0.017$ over 'arrow' ($M = 2.03$). We also found significant effects in the related post-experiment comparison measures of **trust** ($H(2) = 21.56, p < 0.0001$), and **confidence** ($H(2) = 20.63, p < 0.0001$). Post-hoc analysis for the **trust** comparison found that 'combined' ($M = 2.52), p < 0.0001$ and 'heatmap' ($M = 2.16), p = 0.0051$ were both ranked significantly higher than 'arrow' ($M = 1.32$). Likewise, post-hoc analysis for the **confidence** comparison also found that 'combined' ($M = 2.58), p < 0.0001$ and 'heatmap' ($M = 2.05), p = 0.032$ were both ranked significantly higher than 'arrow' ($M = 1.37$). These results all serve to **validate H1.a**.

Many participants shared similar insights in the post-experiment survey, reporting trust in the 'combined' condition over 'arrow' because they could reason about the rationale of the suggestions:

- *"The combination of a "safe" path and heatmap information helped me trust the system because I could compare the assessed path with the sensor information and make my own decision"*

We also found a significant effect in favor of the 'combined' condition over 'arrow' for the **Interpretability** scale ($H(2) = 8.26, p = 0.039$). Post-hoc analysis with Dunn's Test found that participants consistently preferred 'combined' ($M = 2.70), p = 0.040$ over 'arrow' ($M = 2.14$). There was an additional significant effect in the related post-experiment comparison measure of **helpfulness for decision-making** ($H(2) = 19.24, p < 0.0001$). Post-hoc analysis found that 'combined' ($M = 2.53), p < 0.0001$ and 'heatmap' ($M = 2.11), p = 0.0018$ were both ranked significantly higher than 'arrow' ($M = 1.37$). These results serve to **validate H1.b**.

Participants also emphasized how simply following the arrow-based guidance was easy, while noting that they were taking a leap of faith by following the suggestions, a feeling which was alleviated through the addition of the heatmap and its associated transparency.

- *"The arrows were certaintly "easier" to use...The heatmap [guidance] required more thought, but it made me more confident."*
- *"...with the heatmap you could see how confident the system was in its choices... The arrows alone were bad because you couldn't see why the system was changing its mind. "*

Though we found overall significance for the *Mental Load* scale ($H(2) = 6.68, p = 0.036$), there was not enough statistical power to make definitive post-hoc conclusions. Analysis with Dunn's Test found nearly significant effects for 'arrow' ($M = 2.63$) being rated as higher load than both 'heatmap' ($M = 2.24), p = 0.062$ and 'combined' ($M = 2.32), p = 0.099$. Interestingly, this effect appears to be indicating the opposite of hypothesis H1.c, showing that conditions containing prescriptive guidance are rated as more taxing. However, due to the lack of significance, **H1.c is inconclusive**, and will require more data to definitively address.

Some insight into this effect is visible though in participant reactions to path changes in the 'arrow' condition. Participants felt they needed to follow the guidance given to them since they had no other information, but felt stressed and irritated when they encountered sudden path changes, especially switchbacks.

- *"Arrow advice was frustrating when it kept changing the suggestions. I was not sure why it was happening."*
- *"I would like to be involved in the decision making, rather than being restricted by the guidance system. The arrow system essentially tells the player to trust its decision with no alternative consideration."*

The post-experiment comparison measure of **usefulness** also had significant effect. ($H(2) = 15.98, p = 0.0003$). Post-hoc analysis revealed significant effects for 'combined' ($M = 2.58$) being

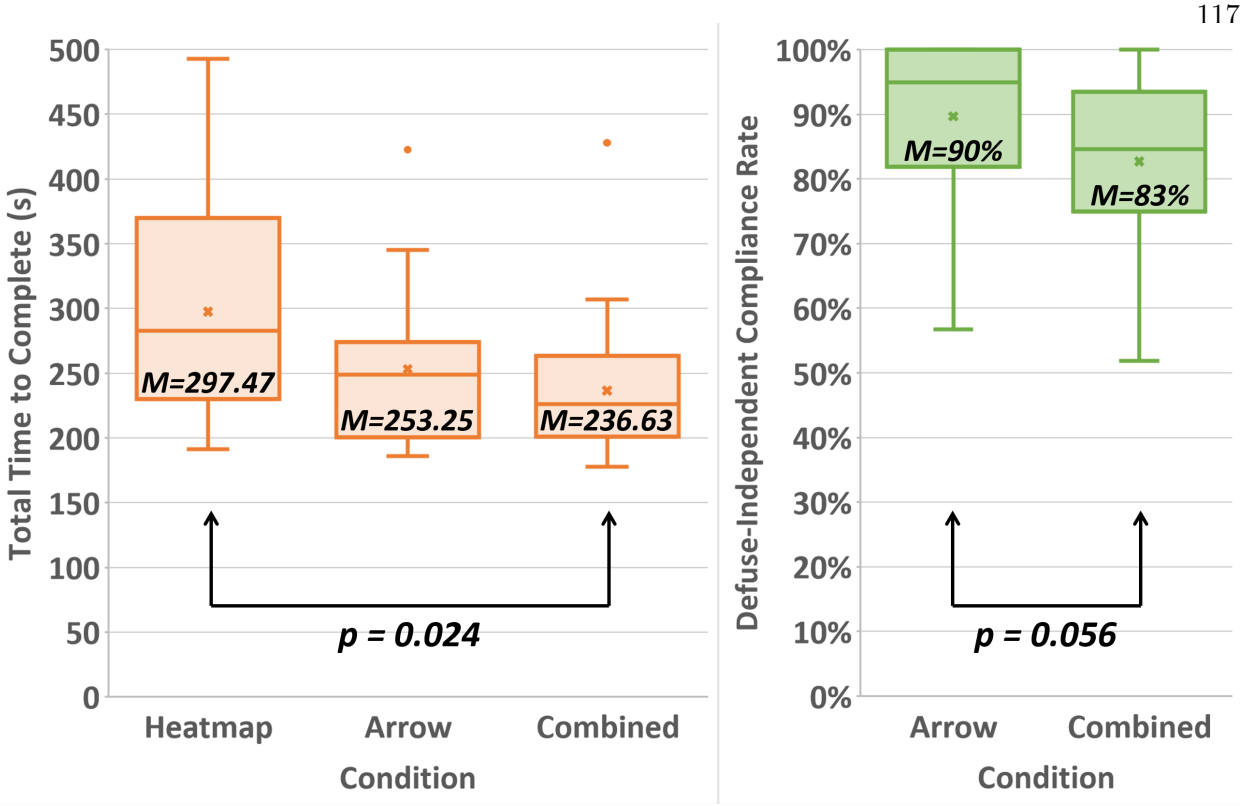


Figure 5.4: 'Combined' visualization achieves the Total Time performance benefits of 'arrow' while allowing for reduced rigidity in suggested action compliance.

rated as more useful than both 'arrow' ($M = 1.89), p = 0.0003$ and 'combined' ($M = 1.53), p = 0.032$. Lastly, in asking which guidance participants would prefer to use in a hypothetical round 4, the significant favorite was also 'combined' based on a one-sample test of proportions (11/19 participants chose 'combined'; a greater proportion than the expected random proportion of 0.33, $p = 0.024$).

5.6.1.2 Objective Analysis

For measuring the performance of a round, we investigated two measures: **Total Time** and **Time per Move**. The domain was small enough that most participants solved it within a few moves of the optimal solution length. For all objective data analysis, we removed a single round out of the 57 conducted where the experiment was interrupted and the participant removed their HoloLens for an extended period of time, invalidating the data. We analyzed these performance

metrics using a one-way analysis of variance (ANOVA) with experimental condition as a fixed effect. Post-hoc tests used Tukey's HSD to control for Type I errors in comparing performance across each guidance type.

The ANOVA revealed significant effects for both total time ($F(2, 53) = 3.91, p = 0.026$), and time per move ($F(2, 53) = 3.78, p = 0.029$). Post-hoc analysis for total time with Tukey's HSD shows that participants spent significantly less time solving the puzzle in the 'combined' condition ($M = 236.63s), p = 0.024$ compared to the 'heatmap' condition ($M = 297.47s$). The 'arrow' condition ($M = 253.25s$) fell in the middle, with no significant effects. Post-hoc analysis for time per move discovered that participants spent significantly less time per move in the 'arrow' condition ($M = 8.59s), p = 0.045$ compared to 'heatmap' ($M = 10.41s$), with 'combined' ($M = 8.74s), p = 0.066$ nearly achieving significantly lower time per move compared to 'heatmap'. The effects surrounding time and time per move serve to **validate H2**.

We were also interested in observing how differing compliance rates affected total moves in rounds using the 'arrow' and 'combined' conditions (conditions which contained prescriptive guidance), to see whether straying from the prescribed path led to changes in performance. Using Pearson's correlation coefficient, in the 'arrow' condition, there is a significant negative correlation between compliance rate and total moves (i.e., the more participants follow the guidance, the quicker they solve the puzzle) ($r(18) = -0.49, p = 0.039$). However, there is no such statistically significant correlation between compliance rate and total moves in the 'combined' condition ($r(19) = -0.11, p = 0.64$). This suggests that deviation from the path is a bad strategy when it is not informed, as in the case of 'arrow', but when there is extra information to work with such as the addition of PMF data in 'combined', it may be acceptable to deviate in certain cases.

Interviews from participants who deviated from the system's suggestions paint a similar picture: providing PMF data empowers people to act more independent of the guidance.

- *"It gives specific recommendations which are really just easy to use and follow. But it also gives you the broader understanding of the map to make deviations when they make sense."*

To determine the extent that this strategy was employed by participants, we compare the compliance rates of 'arrow' and 'combined'. Using a one-tailed t test, we measure whether participants strayed from the path more frequently in the presence of the added PMF data. Running this test, no significance was found between 'arrow' ($M = 0.83$) and 'combined' ($M = 0.78$); ($t(35) = -0.84, p = 0.20$). However, a high proportion of noncompliant moves were overly conservative defuse actions, especially early in rounds. By measuring the **defuse-independent** compliance rate between the two conditions, representing the frequency with which participants stayed on the same recommended path, we find a near-significant effect between 'arrow' ($M = 0.90$) and 'combined' ($M = 0.83$); ($t(35) = -1.63, p = 0.056$). This compliance data suggests that the addition of PMF data in 'combined' allows for more independence and injection of beneficial human decision compared to the monolithic 'arrow', and that participants are willing to take advantage of this. These findings **support and nearly validate H3**.

However, from the survey responses, it is evident that many participants altered their search strategy in the 'combined' condition: instead of entirely relying on the system's suggestions, participants started mixing the provided guidance with their own intuition.

- *"With just the arrow guidance, I was forced to follow it always since there was no other way to gather information. With the heatmap and combined (since it includes the heatmap) I was able to incorporate my own decisions as well."*

5.7 Algorithmic Limitations and Making MARS Hierarchical

MARS is a promising framework for integrating human teammates into complex multi-agent robot planners for multi-objective navigation and search tasks. However, it suffers from scalability issues when confronted with large numbers of agents and high state counts, limiting its applicability in certain real-world robotics domains. We address these limitations by making MARS hierarchical through the introduction of a spatial hierarchy technique for visual explanation generation. This approach allows the MARS framework to be tuned to tasks with arbitrary environment size and

spatial resolution requirements [247].

By exploiting the inherently hierarchical nature of search tasks, we can transition between levels of state and action abstraction depending on the phase of the search, allowing for planning at varying levels of detail (similar to how humans naturally think about search) [248, 249]. This methodology enables the MARS framework to be applied to a much broader class of real-world search scenarios.

5.7.1 Approach

Here, we describe our modified hierarchical multi-agent reinforcement learning planner. The delta between this algorithm and the previous MARS algorithm in [193] includes the following improvements: (1) it introduces a hierarchical structure capable of reasoning over arbitrary environments, making it scalable to real-world applications, and (2) it enhances the interpretability of guidance [248, 249].

5.7.2 Hierarchical MARS Algorithm

At a high level, the hierarchical algorithm functions similarly to MARS as described in [193]. We refer to this version as H-MARS (Hierarchical Min-entropy Algorithm for Robot-supplied Suggestions). Human and robot Markov Decision Processes (MDPs), encoding the heterogeneous goals and capabilities of each agent class, are solved via online reinforcement learning to generate actions for robot agents and action suggestions for human agents, using a shared, dynamically updating state-wise probability mass function (PMF) to synchronize a notion of likely goal locations between all agents. The algorithm differs, however, in the addition of the ability to group together low-level states into a smaller number of larger regions. H-MARS is capable of dynamically switching between levels of state space abstraction for providing its actions and guidance: considering the entire environment with regions as states, or considering a single region with low-level discretized states (e.g., grid squares). The concept is inherently recursive, and can be extended beyond two levels of spatial resolution: for example, an environment could be divided into regions, which are

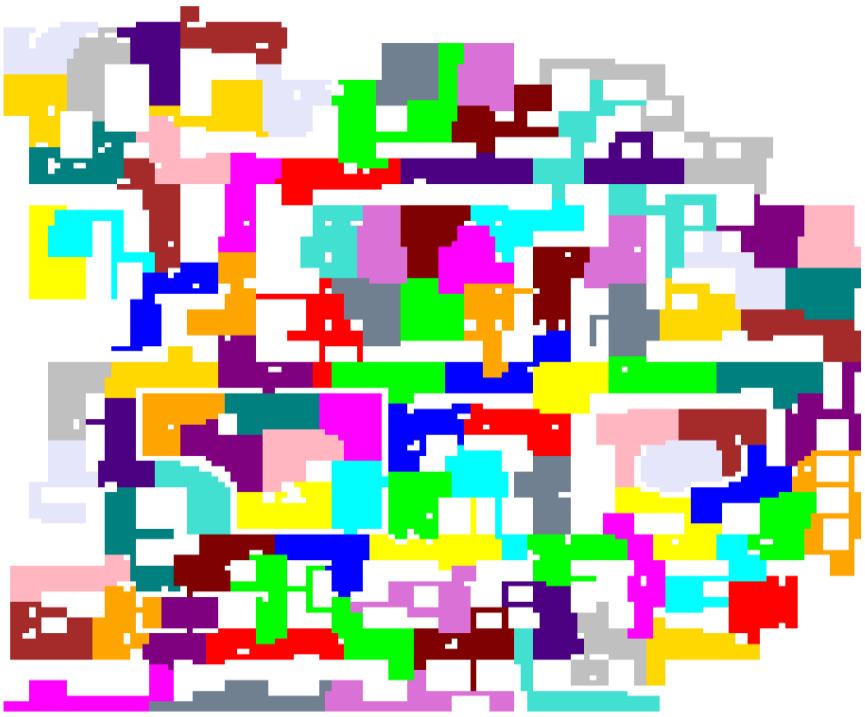


Figure 5.5: Results of graph partition on 2-dimensional projection of an experimental environment. In this example, the environment is divided into approximately 10,000 grid squares (3m x 3m each), grouped into 100 regions, with impassible obstacles rendered in white.

themselves divided into sub-regions, which are divided into individual states.

To obtain these regions, we discretize our environment into a grid of a desired spatial resolution, and form a graph with grid squares as nodes and edges connecting adjacent, traversable nodes. We then run the METIS graph partition algorithm [250] over this graph, producing contiguous regions of reachable states. To optimize for computational efficiency when running the algorithm in real-time, the number of regions produced should roughly equal the nth root of the total number of states in the environment (for a desired n-level hierarchy). By considering an equal number of states in each phase, the complexity of the combined computation is minimized, reaching a state of Pareto-optimality [251]. An example of this can be seen in Fig. 5.5, where an environment of 10,000 discrete states is programmatically divided into 100 regions. Assuming a two-level hierarchy, the algorithm progresses through three phases for each time-step, corresponding to swapping the state space, action space and reward function input to MARS between levels of abstraction:

Phase 1 (Local Window Search): The algorithm first considers individual states within

a limited distance of each agent. This is to avoid edge cases that would arise by starting Phase 2, involving potentially high-reward actions taking agents to physically nearby states that happen lie across a region boundary. By considering these actions first, we avoid the situation where they receive an outsized reward penalty, normally given to represent the time taken to travel to a separate region.

Phase 2 (Inter-Regional Search): If the tuned reward threshold within Phase 1 is not passed, the algorithm moves on to considering entire regions as single states, with the state-wise PMF used to calculate the expected number of targets to be found per region. The algorithm decides whether it is preferable to stay and search within the current region, or travel to a new, more target-rich region, considering the added movement penalty for taking the time to travel to a separate region, proportional to that region's distance from the current region. If the algorithm decides an agent should move regions, it commands actions or provides action recommendations that path the agent to the nearest edge of the new target region. If the algorithm decides to stay within the current region, it progresses to Phase 3.

Phase 3 (Intra-Regional Search): The hierarchical MARS algorithm now moves to consider the states within an individual region for calculating optimal agent actions, utilizing the PMF value of states in reward calculations, identical to the state space, action space, and reward function of MARS. The phases are repeated every time the global PMF updates in response to the accumulation of agent observations.

5.7.3 Algorithmic Evaluation & Results

We validate the utility of our Hierarchical MARS (H-MARS) framework's ability to handle large state spaces by conducting simulation episodes of a multi-objective collaborative search task, with simulated human agents following the system's guidance. Our evaluation includes a comparison of (1) Hierarchical MARS (H-MARS), (2) Ablated H-MARS without phase 1, (3) MARS, as described in Chapter 5, and (4) Non-RL limited horizon multi-objective A* [252], in environments of varying sizes.

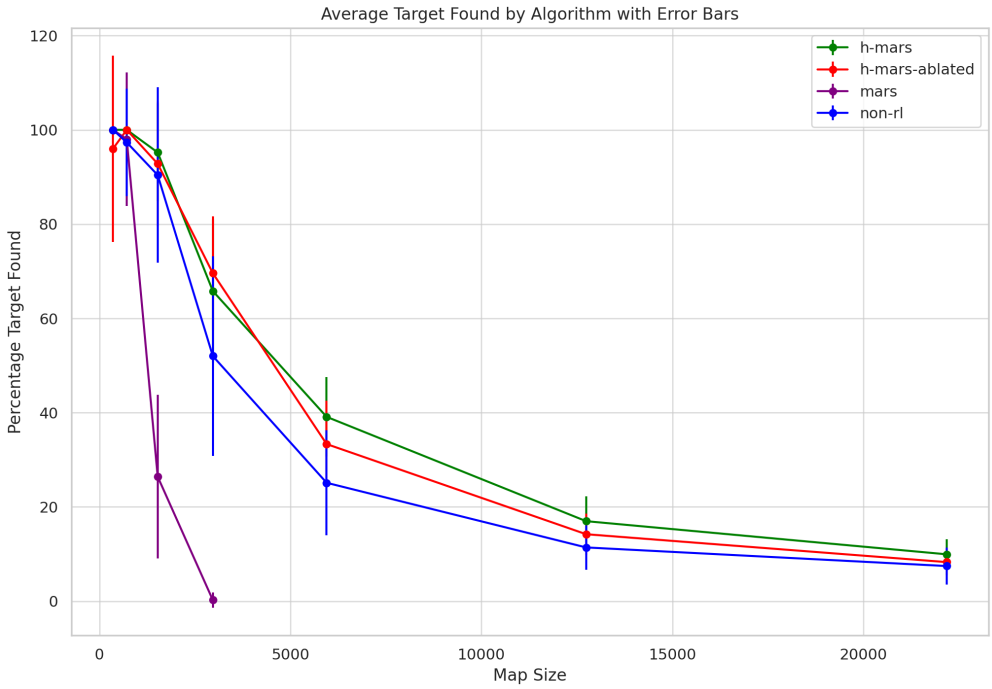


Figure 5.6: Simulation results of H-MARS compared to other baselines within a multi-objective collaborative search task. The X-axis represents the number of traversable states, and the Y-axis shows the percentage of targets found in a simulation run with a fixed step count. H-MARS performs consistently better than the rest of the baselines, while MARS performs worse compared to the other methods.

The results are summarized in the plot (Fig. 5.6). On the x-axis, we have the number of state counts, and the y-axis indicates the percentage of targets found. We ran 50 simulation runs for similar maps for each of the algorithms mentioned above. As the state space count increases (traversable states), MARS performs worse compared to other methods—it cannot solve for the agent's policy within 300 seconds for states greater than 3,000. In contrast, H-MARS performs consistently better than the rest of the baselines, even when traversable state counts increase to 30,000, showcasing the utility of making MARS hierarchical and enabling its application in more real-world scenarios.

5.8 Discussion and Key Takeaways

We summarize key takeaways to inform the design of visual guidance systems for human-robot teaming, aligning with findings in the xAI literature where people consider robots to be more helpful and trustworthy when they justify their actions [20, 186].

T1: Prescriptive guidance, in the form of arrow or waypoint based suggestions, can be inherently restrictive. This guidance is easy to follow but puts human teammates in an 'automatic' pattern of thought (also known as system 1 thinking) [253]. In contrast, descriptive guidance forces the user to take more conscious actions (system 2 thinking). By combining both types of guidance, human teammates can leverage the explicit prescriptive guidance to help them reduce their workload, while still maintaining environmental awareness and acting with greater independence.

T2: In the 'arrow' condition, participants initially had a highly variable degree of trust in the system's suggestions. Some people over-trusted the guidance, taking its suggestions to be inherently correct, and some under-trusted the guidance, ignoring the arrow to defuse more conservatively. By providing descriptive data alongside prescriptive suggestions, people's behavior often tended towards a degree of trust somewhere in the middle of the two extremes, as they could see for themselves where a drone was more or less confident. This echoes findings on the ability of interpretable systems to mitigate over- and under-trust [35, 192].

T3: Some participants found it difficult to notice changes in the PMF when the change was not in their field of view. They suggested adding a feature notifying the user when a new high confidence target was found so they could be made aware of it. Additionally, some participants expressed desire to receive an explanation when a highly confident square suddenly becomes less confident.

T4: Participants did not like sudden path changes, viewing the behavior as unconfident. Participants expressed a preference for direct paths, desiring an explanation when a change was necessary.

5.8.1 Conclusion.

In this chapter, we introduced two complementary modalities of AR-based visual guidance: prescriptive guidance (visualizing recommended actions) and descriptive guidance (visualizing state space information to aid in decision-making). We also presented an algorithm to generate and utilize these modalities in partially-observable multi-agent collaborative tasks to communicate environmental uncertainty and provide interpretable recommendations for efficient and fluent human-robot teaming. We compared these modalities in a human subjects study, demonstrating the value of providing visual insights into environmental uncertainty alongside robot-generated suggestions, which improved trust, interpretability, performance, and human teammate independence. Additionally, we presented a hierarchical version of the same algorithm, which scales favorably with state space count. Finally, we consolidated our findings into actionable recommendations for applying them in human-machine teaming scenarios.

Chapter 6

The Utility of Justifications in Human-Machine Teams: When and What to Explain

"The activity of thinking is invisible to others and, to a large extent, also to ourselves, until it crystallizes into words or deeds."

— Hannah Arendt, *The Life of the Mind*

This chapter continues the thread from the previous chapter on multimodal decision support in human-machine teaming for mental model alignment and justification. Specifically, it focuses on leveraging multimodal explanations as **justifications**, timed appropriately to instances of expectation mismatch, with the intent of convincing or influencing a human agent.

In the MARS study (Chapter 5), some participants were frustrated when the system's recommendations exhibited unexpected behavior, such as sudden path changes. These inexplicable recommendations resulted from policy optimization within an uncertain environment. Participants viewed this emergent behavior as confusing and unconfident, expressing a desire for explanations, echoing previous findings [21, 254]. Similarly, we noticed that some participants over-trusted the guidance (taking its suggestions to be inherently correct), while others under-trusted it (frequently ignoring good advice). Exit interviews indicated that participants did not have an appropriate way of judging the quality of recommendations, leading to variable perceived system reliability.

Justification is an important facet of policy explanation, a process for describing the behavior of an autonomous system. In human-robot collaboration, an autonomous agent can attempt to justify important decisions by offering explanations as to why those decisions are right or reasonable,

leveraging a snapshot of its internal reasoning to do so. Without sufficient insight into a robot's decision-making process, it becomes challenging for users to trust or comply with those important decisions, especially when they are viewed as confusing or contrary to the user's expectations (e.g., when decisions change as new information is introduced to the agent's decision-making process), as evident in our previous studies [24, 193].

In this chapter, we characterize the benefits of justification within the context of decision support during human-machine teaming (i.e., agents giving recommendations to human teammates). We introduce a formal framework using value of information theory to strategically time justifications during periods of misaligned expectations for greater effect. We also characterize four different types of counterfactual justification derived from established explainable AI literature and evaluate them against each other in a human-subjects study involving a collaborative, partially observable search task. Based on our findings, we present takeaways on the effective use of different types of justifications in human-robot teaming scenarios to improve user compliance and decision-making by strategically influencing human teammate thinking patterns. Finally, we present an augmented reality system incorporating these findings into a real-world decision-support system for human-robot teaming.

6.1 Introduction and Motivation

Many works in the explainable AI (xAI) literature have illustrated the benefits of illuminating the black box of AI decision-making for end users interacting with autonomous and robotic agents [2, 8, 255]. Various xAI techniques facilitate better transparency into collaborative robots' choices, improving trust, interpretability, and user acceptance [21, 24, 256, 257]. However, if explanations are given at inopportune times with poor context, they can produce the opposite effect [258]. Furthermore, different explanation content can have differing effects on a human collaborator's mental model, which can impact their behavior [158, 259]. In this work, we hypothesize that since human collaborators have limited cognitive bandwidth to process explanations, it is best to time them strategically for maximum impact on improving understanding and behavior. We also

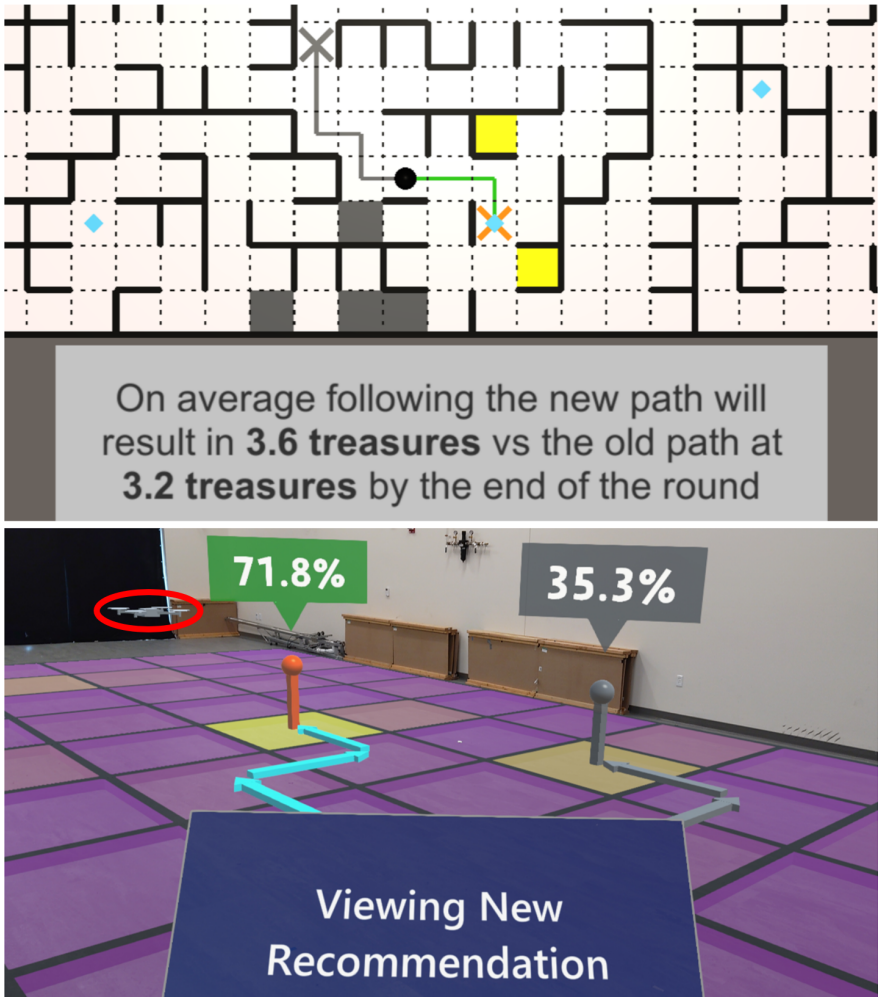


Figure 6.1: **Top:** a counterfactual policy-based justification provided by drones (blue diamonds) to the human in a collaborative 2D treasure hunting game. **Bottom:** a counterfactual environment-based justification showing the relative percentages of finding a target, provided by a drone (circled in red) in an augmented reality navigation interface. Both justifications are attempting to explain to a user why they should take a new (colored) recommended path, rather than the old (gray) path.

propose that the content and manner in which the explanations are given should be tailored to a collaborative context to encourage the desired effect on a human teammate.

In collaborative human-robot interaction tasks, accounting for a human in a multi-agent planner is challenging due to the innate unpredictability and opacity of the human's decision-making [14, 260]. Therefore, having a robotic teammate also act as a decision support system for the human, suggesting actions for the human to perform while itself working towards a shared task, is helpful for alleviating this unpredictability [20, 21, 193, 261]. With this type of interaction, it is crucial that autonomous agents **justify** their behavior or suggestions when they deviate substantially from

the human teammate's expectations.

We define justifications in this context as explanations timed appropriately to instances of expectation mismatch, with the intent of convincing or influencing a human agent. For example, in a human-robot collaboration scenario where a robotic agent is providing navigation recommendations, a sudden change in the recommended direction may appear confusing and strange to the human teammate, and is likely to be disregarded [193]. A justification (see Fig. 6.1 for examples) provided in this context serves to convince the human teammate of the utility of the previously difficult to interpret recommendation. Our work addresses two research questions: 1) When are such justifications most impactful and useful? And 2) What information should be presented in justifications to improve human teammate decision-making and behavior?

The core contributions of this work are as follows:

- A novel mathematical framework, informed by value of information theory, to decide **when** a robot collaborator should justify its recommendation to a human teammate, validated by an expert-feedback case study for determining the utility of justification timing strategies.
- A methodological characterization of four different types of justification, derived from established features in xAI literature, along with a validation and analysis of these justification types via an online human subjects study.
- A set of actionable design recommendations and implementation strategies for the use of justifications in human-robot interaction, taking into account differing levels of human and robot decision-making competence, along with an augmented reality interface showcasing these design principles for practical applications.

6.2 Background & Related Work

Explainable AI and Human-Robot Interaction: Recent research on shared mental models within human-robot collaboration has shown the importance of explainability for enhancing interaction efficiency, fluency, and safety [8, 12, 262]. This is particularly relevant in the context

of model reconciliation, where mismatches in expectations can lead to catastrophic failures [21, 158]. Explainable AI can help bridge the gap between human and robotic agents by making complex models more understandable, allowing for faster debugging and failure recovery, ultimately improving joint performance [7, 8, 92].

As such, it is important for robotic agents to be able to effectively communicate and explain their decision-making rationale to human collaborators, with awareness of how these explanations influence and affect team dynamics. Moreover, research has also shown that people trust autonomous agents more when they convey their decision-making process [34, 263]. Robots with this explanation-providing capability are generally perceived to be more helpful and transparent [20]. Conlon et al. [264] show that when a robot provides a self-assessing explanation, operator trust more appropriately aligns with robot ability, leading to increased performance and trust.

Explanation Strategies: Research in two areas of explainable AI are particularly relevant to explanation generation: methods that explain how a learned model functions (explainable ML) and methods that produce explainable agent behavior during human-in-the-loop interaction [265]. Explainable ML methods are often aimed at helping developers interpret complex classifiers by illustrating how individual parameters impact model output. Popular techniques include local approximations like SHAP [236], model-agnostic methods like LIME [7], and visualizations like Grad-CAM [29].

Explainable behavior methods attempt to make the intentions of robotic agents clearer to humans by improving metrics like explicability [105], predictability [266], or legibility [76]. Research has demonstrated that people dislike inexplicable behavior from robots, rating it as frustrating, and leading to mistrust of the robot [8, 267]. Robot behavior that attempts to align itself with human expectations often must sacrifice optimality to achieve high explicability. In Tabrez et al. [193], participants in a collaborative search scenario expressed a preference for explanations from an autonomous agent when its behavior was unexpected or confusing. These explanations, provided they are contextualized properly to mismatches in human and robot expectation, can serve as a bridge between explicability and optimality: alleviating the negative effects of inexplicable but

optimal robot behavior, and building trust in the system over time.

Explanations as Justification: This work focuses on the strategic use of explanations as justification in human-robot teaming. This involves timing explanations to an instance of expectation mismatch between humans and robotic agents, with the goal of influencing a human teammate. Correia et al. [254] found that using justification as a recovery strategy for robot failures can mitigate the negative perception of those failures. Prior work has focused on using justification to explain why a decision is good or bad, without necessarily aiming to give an explanation of the decision-making process [20, 186]. In this work, we introduce and analyze different types of justifications aimed at addressing both of those goals.

6.3 Definition of Application Domain

To ground and evaluate our contributions, we utilize a multi-target search and retrieval problem as a representative human-robot teaming application. This multi-goal, multi-agent planning domain includes agents with heterogeneous capabilities operating under partial observability.

We utilize an experimental paradigm previously established by Tabrez et al. [193], which assumes two distinct classes of heterogeneous agents working toward a multi-objective task (e.g., search and recovery): autonomous agents (information-gathering agents that move through the environment and take sensor observations) and human agents (interactive agents that can directly affect the environment state with their actions and complete objectives, such as collecting a sample) in a partially observable domain. In this paradigm, humans serve as interactive agents that receive action recommendations from autonomous information-gathering agents that typically have access to features the interactive agents cannot directly perceive. The decision-making process for each class of agent is codified by a separate Markov Decision Process (MDP):

- Autonomous agent MDP, M_r, is defined by the 4-tuple: (S_r, A_r, T_r, R_r), where S_r is the set of states in the MDP, A_r is the set available actions, T_r is a stochastic transition function describing the model's action-based state transition dynamics, and R_r is the reward function

$R_r : S_r \times A_r \times S_r \to \mathbb{R}$.

- Recommendations for human agents are generated using an MDP model of the human M_h defined by a 4-tuple (S_h, A_h, T_h, R_h).

Environmental uncertainty over task-relevant variables (e.g., whether a location contains a buried sample) is characterized by a dynamically-updating probability mass function (PMF). This PMF serves as a shared utility function common to all agents (both human and autonomous), and can be communicated to human teammates as it changes in response to autonomous agent observations to provide insight into the agent's policy (additional detail provided in Section 6.5.1).This relationship can be seen in Fig. 6.2.

In the multi-target search task, the PMF is in essence a heatmap representing the probability at each location for finding a target. The autonomous agent MDP M_r generates optimal moves for these information-gathering agents to attempt to collapse the uncertainty of that PMF by locating targets via sensor observations. Meanwhile, the human MDP M_h generates recommendations for the human agent to follow to achieve the task goals, constantly updating based on the most recent PMF.

The novel justification framework evaluated by our experiment was situated within the context of a human-drone collaborative search task, an established evaluation domain for decision support [193]. Fig. 6.2 shows the interaction flow of the task. In this section, we will use the circled letters in the diagram to walk through its implementation.

To start, drones solve for their next actions (a) using the MDP M_r; in our domain each drone is assigned its own segment of the environment to cover to ensure uniform search coverage. As the drones take their actions (b), they observe noisy sensor readings over the cells they fly over to attempt to detect targets (c). Using these readings, the shared PMF undergoes a Bayesian update. Next, the system calculates a recommendation for the human using M_h (d). The system determines whether a justification is needed, and if so, generates one (e); the justification framework (the primary contribution of this work) is described in detail in Sections 6.4 and 6.5. The human's

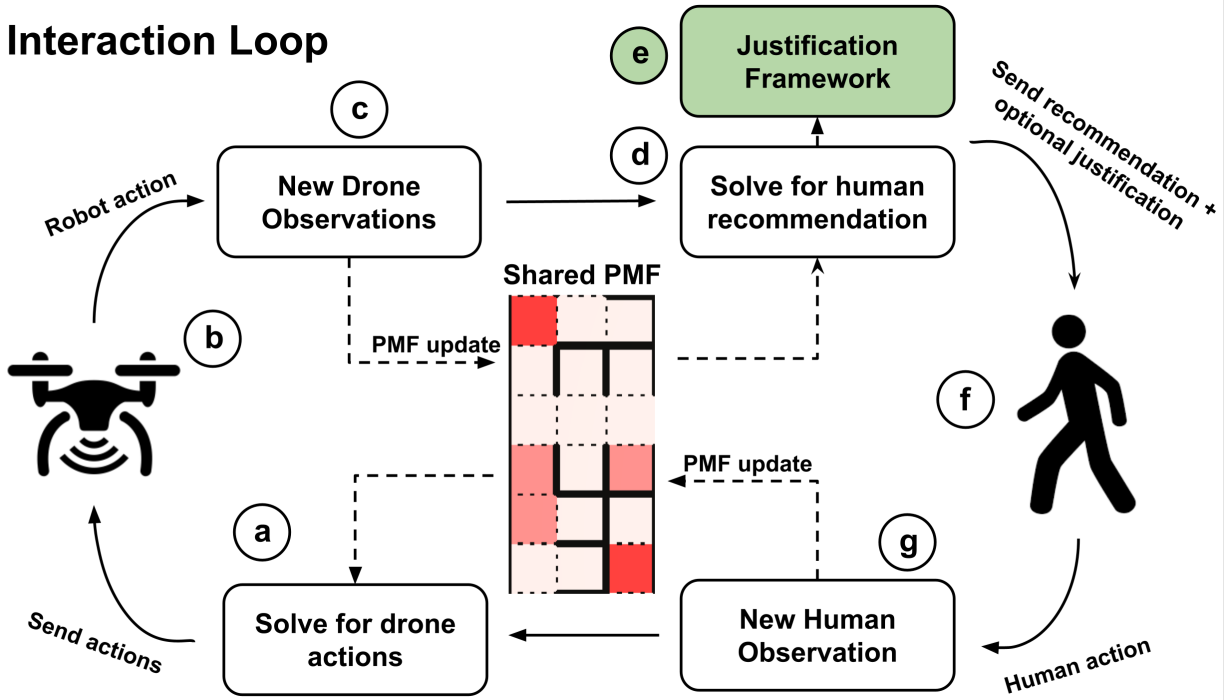


Figure 6.2: The loop describing the human-drone interaction with shared PMF in our domain. The Justification Framework, the primary contribution of this work, is highlighted in green.

next recommendation and optional justification are sent to the human, who then takes their next action (f). Based on the system's observation of the human action, the PMF and state is updated again (g), and the cycle returns to (a).

6.4 Justification Framework: Timing

In this section, we address the question of "when" justification should be provided within human-robot teaming scenarios, and present a novel framework for the timing of justifications based on value of information theory. Throughout this section, we focus on the use case where the collaborating agent is acting as a decision support system, providing recommendations to a human agent who can either comply with or reject them.

6.4.1 Spectrum of Justification Timing Strategies

Prior work has shown that in collaborative human-robot interaction, humans are highly influenced by the timing and frequency of those interactions [258]. To examine the question of when and how frequently justifications should be presented, we start by anchoring the range of possible actions at the two extremes: never justifying or always justifying.

There are two general criteria that would render a justification unnecessary within a human-robot collaboration. 1) there are no actionable consequences stemming from the recommendation to be justified, or 2) the robot's recommendations are generally accepted and trusted without scrutiny [3]. In most adaptive autonomy use cases, the second criterion is rarely met, especially in high uncertainty environments [193, 268]. Prior research has found that whenever there is a misalignment of expectations between human and autonomous teammates, explanations are expected to be provided [158, 193]. These expectation mismatches can stem from a variety of causes, including sudden changes in recommendation or a recommendation based on environment data that is unknown to the human [21]. Trust and reliance in these systems deteriorate when they lack the capability to justify their recommendations in the presence of such mismatches [254]. In these scenarios, never justifying is undesirable.

On the other hand, always justifying is ill-suited for human-agent collaboration. Prior research has shown that administering too many queries increases frustration and irritation in users [216]. Justifying too frequently can lead to habituation, as repeated explanations reduce user responsiveness to them [269, 270, 271, 272]. Thus, always justifying is also undesirable.

6.4.2 Strategically Timing Justifications: Value of Information

Even though justifications have benefits, agents should provide them strategically to take advantage of them efficiently. As there is a direct cost of increased workload and habituation inherent to providing an explanation to users, justification should only be made when the value exceeds the cost. We utilize value of information (VOI) theory [273] to decide how much value a

specific justification may add.

Value of Information. VOI is typically used in autonomous systems contexts to maximize the information that a system can gather or observe by using a "pull" communication pattern, where a requesting agent (usually an autonomous system) formally weighs the cost to query a responding agent (usually a human) to provide additional information [173].

However, as we are operating within the context of conveying an explanation to a human agent autonomously, we adopt VOI in a "push" communication pattern, where an information-providing agent (robot teammate) formally weighs the cost to a receiving agent (a human) in parsing that information, along with the cognitive burden of interrupting their current task [216].

Justification Framework. Using the human MDP M_h described in Section 6.3, our framework constructs an optimal policy for the human π_h^*. However, this optimal recommended policy is not necessarily agreed upon by the human and the autonomous agents since they may have differing reward functions. Therefore it is necessary for the system to model the human and estimate what their π_h should be.

- $\widehat{\pi_h^*}$ is a human's optimal policy as derived from the human's own internal reward function $\widehat{R_h}$ and operating using their world model $\widehat{M_h}$. The notation '$\widehat{--}$' denotes that the variable in question is derived from the human's internal model of the world, which is latent to the system and must be estimated.

- π_h^* is the system's optimal policy for the human derived from R_h, the system's model of the human's reward function and its model of the human MDP M_h. The policy recommendation can change based on receiving new information (e.g., new sensor readings).

When there is perfect synergy between the human and the system (a shared mental model), these two policies will be the same ($\widehat{\pi_h^*} = \pi_h^*$). However, the human's and the system's understanding of the optimal policy will drift as the system receives new information and makes updates to π_h^* while the human makes potentially different choices while using out-of-date information, leading to a mismatch in the mental model.

The human and the autonomous agent will have two separate understandings of the expected reward for following a given policy starting from a state s:

- $\mathbb{E}_{\pi_h^*,s}(R_h)$ is the expected reward the **system** expects the human to receive by following the recommended policy.
- $\mathbb{E}_{\widehat{\pi_h^*},s}(\widehat{R_h})$ is the expected reward the **human** expects to receive by following their own policy.

Justification is needed when the autonomous agent's recommendation appears unintuitive or confusing to a user. We hypothesize that the two primary reasons for this confusion are 1) an explicit mismatch in the expected reward, or 2) a mismatch in the sequence of states that are expected to be visited even in the case of identical expected reward.

The first contributor is the mismatch in expected reward and is formalized as:

$$\mathcal{D} = |\mathbb{E}_{\pi_h^*,s}(R_h) - \mathbb{E}_{\widehat{\pi_h^*},s}(\widehat{R_h})| \tag{6.1}$$

Where $\mathcal{D}$ is a scalar representing the difference in the robot's expected reward and the human's expected reward from following their respective policies for the human agent. To formalize the second contributor, it is useful to define two possible trajectories for the human.

- ψ_h denotes the sequence of states the **system** thinks the human should traverse, obtained from a rollout of π_h^* starting from current state s.
- $\widehat{\psi_h}$ denotes the sequence of states the **human** thinks the human should traverse, obtained from a rollout of $\widehat{\pi_h^*}$ starting from current state s.

The expected mismatch in path is defined as a distance function between the two paths:

$$\mathcal{T} = dist(\widehat{\psi_h} - \psi_h) \tag{6.2}$$

Here, $\mathcal{T}$ is a scalar representative of the difference between the robot's recommended path and the human's expected path. We define the value of a justification, V($\mathcal{J}$), as a piecewise linear filter with three components:

$$V(\mathcal{J}) = max \begin{cases} \alpha * \mathcal{D} \\ \beta * \mathcal{T} \\ \gamma * \mathcal{D} + \kappa * \mathcal{T} \end{cases} \tag{6.3}$$

α, β, γ, and κ are tunable hyper-parameters. The first component of Eq. 6.3 captures the mismatch in the expected reward, the second captures the mismatch in the expected path, and the third provides a more comprehensive filtering criteria based on a linear combination of the two. The three filters combine to create an expressive notion of the value of a potential justification.

This justification to a user comes at a cost $C(\mathcal{J})$, which is highly dependent on the particular task and mode of communication, and should tuned separately per domain. A justification should only be triggered if the expected benefit to the user is higher than the justification cost.

$$V(\mathcal{J}) - C(\mathcal{J}) > 0 \tag{6.4}$$

In human-robot teaming scenarios, as the mismatch between the robot's recommendation and human mental model increases, the usefulness of the robot's recommendations decrease. VOI can be used to determine the trade-off between providing justification to bridge the gap and making the recommendations more useful.

Additional Implementation Details. Here, we present additional details about how we applied this framework to our domain. The value of a potential justification relies on the human's internal policy $\widehat{\pi_h^*}$ and the system's recommended policy for the human π_h^*. Since the human's internal policy is latent from the perspective of the system, we infer the human's most likely reward function $\widehat{R_h}$ based on the information they can observe, and derive their policy $\widehat{\pi_h^*}$ assuming that humans optimize expected reward given their current reward knowledge: a common practice

within inverse reinforcement learning and preference learning literature [20, 73]. Since the only reward information humans receive is communicated via the robots, we update the human's reward function $\widehat{R_h}$ and resultant policy $\widehat{\pi_h^*}$ whenever the robot provides a communicative update, using the reward content of that update as an approximation of the human reward knowledge (i.e., using π_h^* from the last recommendation received by the human, at a previous timestep). The human's desired path $\widehat{\psi_h}$ is estimated using π_h^* from that previous timestep.

The specific implementation for our domain of the distance function in Eq. 6.2 to find $\mathcal{T}$ uses an XOR of states in the human's expected path $\widehat{\psi_h}$ and the states in the new recommended path ψ_h. Simply put, the difference function takes into account states that are visited by one of the compared trajectories, but not both. Prior research has shown that people are more concerned by actions that are nearer to them [193, 274]. With that in mind, we weight differences higher the closer they are to the human's current location.

$$\mathcal{T} = \sum_{s' \in \psi_h \oplus \widehat{\psi_h}} \gamma^{d(s', s_h)} \tag{6.5}$$

The distance function is the sum of a tuned discount factor γ raised to the Euclidean distance $d(s', s_h)$ between a state s' and the human's current state s_h ($d(s', s_h)$) for all states s' in the XOR set $\psi_h \oplus \widehat{\psi_h}$.

We combine the scalar state difference $\mathcal{T}$ with the scalar reward difference $\mathcal{D}$, as described in Eq. 6.1, and tune the relevant hyperparameters in Eq. 6.3 to create an appropriate function for the value of justification $V(\mathcal{J})$, justifying whenever it exceeds the cost $C(\mathcal{J})$, tuned for our domain.

6.4.3 Justification Timing Case Study

We validate our VOI-based timing mechanism for offering justifications through a within-subjects expert-feedback case study (n=10) where participants (graduate students in the fields of robotics and human-computer interaction) watched video of three playthroughs of a treasure hunt game (shown in Fig. 6.1-top) with differing justification timing strategies. In this partially

observable maze-like domain, players must uncover as many hidden treasures as possible in a limited number of turns, aided by autonomous drone teammates who explore the maze and provide continually updating recommendations based on their noisy 'treasure detector' sensor readings.

The video paused periodically during trials at moments where a justification (Fig. 6.1-top) could be offered. The experts were asked at each pause how useful the addition of a justification at that point in the game would be, on a scale from 1 (not useful at all) to 5 (very useful), similar to [184].

Each 21-turn long playthrough utilized one of three timing strategies, presented in a random order: justifying once every turn (21 justifications), justifying at regular intervals of once every four turns (5 justifications), or justifying based on the proposed VOI-based mechanism (5 justifications). We hypothesized that users would find strategically timed VOI justifications to be more useful than constant or timed-interval justifications.

As shown in Table 6.1, we found that strategic justification led to the highest average perceived usefulness rating, showing that it is not only preferable to justify less frequently, but also that the specific timing of justifications to periods of high mismatch in expectations is preferable to a similarly infrequent justification strategy.

	Always	**Interval**	**VOI-strategic**
Usefulness Mean	2.34	2.74	4.16
Usefulness SD	1.47	1.31	0.74

Table 6.1: Means and standard deviations of rated usefulness of justification timing (on scale of 1-5) per timing strategy.

6.5 Justification Framework: Content

In this section, we investigate the **content** of effective justification. Drawing from previous works in explainable AI [7, 183, 275], we introduce four broad categorizations of justifications using a 2x2 cross of *environment-centric* vs. *policy-centric* and *local* vs. *global*.

The first axis of the 2x2 cross, *environment-centric* vs. *policy-centric*, determines whether

the justification is grounded in features from the environment that influence the policy, or features of the resultant policy itself. As an example, an algorithm recommending a location for a new wind turbine might provide the average wind speed at various prospective locations as an *environment-centric* justification for those locations. Alternatively, it could provide the expected power produced in a year if a recommended location was chosen, contrasted with the expected power produced if alternative locations were chosen as a *policy-centric* justification.

The second axis, *local* vs. *global*, determines whether the explanation is grounded in a localized, short-horizon context, or a global, long-horizon context. While a *local* justification may focus on the sub-goals and immediate rewards of a given task, a *global* justification would give a broader overview of the end goal of a domain.

All justifications in our framework are structured counterfactually, comparing the recommendation expected by the human, derived from a model of their own policy $\widehat{\pi_h^*}$, to the current recommendation actually given to the human by the robot derived from π_h^*. Counterfactual explanations are broadly defined as answers to contrastive questions of the form "Why did outcome P happen rather than outcome Q? [276]" These explanations can be conveyed via natural language or visually. Counterfactuals have shown usefulness for model debugging and failure recovery, as these types of explanations provide contextual information about a model's internal reasoning [277, 278, 279].

The following four proposed types of features used in a justification vary along a spectrum of interpretability and comprehension for its users [280].

C1. Environmental Features: These types of features provide a sense of interpretability for users, as they get quick insight into the robot's decision-making rationale.

C2. Policy Features: These features lack in interpretability, since they don't provide any insight into the robot's rationale, but they are highly comprehensible, as the user can easily compare the end results of the agent's decision-making.

C3. Local Features: Humans are bounded by a limited cognitive capacity [281], and tend to prioritize short-term rewards in their own reasoning (e.g., Stanford marshmallow experiment

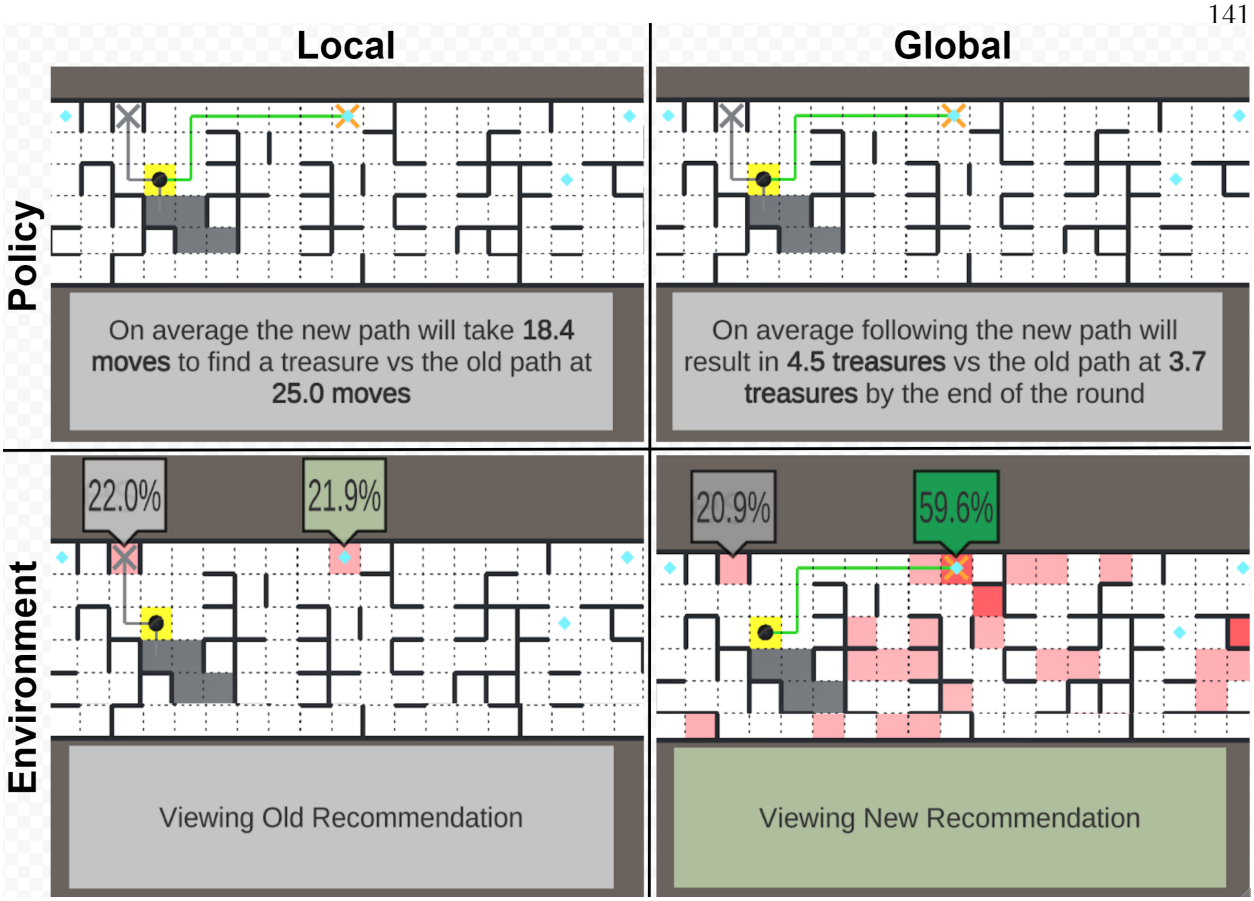


Figure 6.3: The four types of characterized justifications, given during the same gameplay scenario in the treasure hunt domain. Note that the percentages shown on the map in both environment-based justifications involve alternating visually between the old and new probabilities every 1.5 seconds. For simplicity, only the old probabilities are shown for 'environment local' and the new probabilities for 'environment global' in this figure.

[274]). Therefore, local features provide a mix of short-sighted interpretability and compliance characteristics.

C4. Global Features: Global features sacrifice precision for high comprehensibility, succinctly conveying the robot's long-term policy with human-understandable explanations tied to the success criteria of the task itself.

6.5.1 Framing Justifications for Search Tasks

We frame the four proposed justification types, built from the 2x2 cross, in the context of a multi-target search task which utilizes a dynamically updating probability mass function (PMF)

as the primary element of the feature space, a common practice in search and rescue operations [239, 240, 241]. The PMF is a discrete mapping of locations to the probability of a target being found at the location. It is, in essence, a heatmap representing the likely locations of targets across the environment. As information is gathered through environmental exploration, the PMF is updated via Bayes' Rule.

To estimate mental model divergence over time, the system estimates the human's policy $\widehat{\pi_h^*}$ by using the last recommendation given to the human by the robot π_h^*, taken from a previous timestep. This leverages the assumption that the human teammate's mental model is aligned with the most recent guidance they have received from the system, with divergence occurring in the interval between justifications. To repair this divergence, four types of justification can be used:

Environment-centric Global. This justification is conveyed visually by converting the current PMF to a heatmap, with a color gradient from white to red representing the likelihood of finding a target at a particular location. Counterfactuals are employed by cycling images between the PMF heatmap for the previous guidance (an estimation of the features that led to $\widehat{\pi_h^*}$), and the current PMF heatmap (the features that led to π_h^*) at a regular frequency. The numerical probability of finding a target for both the current recommended goal location and the previously recommended goal location is overlaid onto both the prior and current heatmap. This shows explicitly, in numeric form, how the odds have changed to prioritize the current recommendation over the previous one.

Environment-centric Local. This justification uses the same visual representation of alternating between the current and prior PMF as *environment-centric Global*, but instead of showing the entire heatmap, only the heatmap values at the specific goal locations of the current and previous recommendations are shown, alongside the numerical probabilities associated with those two locations.

Policy-centric Global. This justification is conveyed as a natural language counterfactual, focusing on long term rewards. For a multi-target, time-constrained search domain, an example of this justification is "On average, following the new path will result in X targets found overall,

compared to the old path at Y targets found." This takes an abstract concept of expected long-horizon reward and maps it to a human understandable sentence. To estimate values X and Y in our partially observable domain, we utilize a heuristic combining the computed odds over the given recommendation with the overall entropy of the PMF, which decreases over time through exploration. This strategy can be employed for any domain that uses a PMF-based goal likelihood formulation.

Policy-centric Local. This justification is also conveyed as a natural language counterfactual, but focused on short-term rewards. For example, our domain uses the form "On average, the new path will take X moves to find a target, compared to the old path at Y moves." The means of generating X and Y in this case is simpler, as the reward can be more accurately estimated over a fixed-horizon recommendation. It is simply a case of mapping abstract reward to human understandable output. Fig. 6.3 shows how these four justification types were mapped to our treasure-hunt domain.

6.5.2 Hypotheses

H1: Objective Hypotheses

H1.a (Compliance): Participants will have higher compliance with recommendations when given policy-based justifications, compared with environment-based justifications and no justification, as policy-based justification utilizes abstraction and framing effects, resulting in a higher level of persuasiveness[282].

H1.b (Performance): Participants will perform better in the game when given policy-based justifications, compared with environment-based justifications and no justification, as compliance should correlate with performance given the relatively high competence of the recommending system in our domain.

H1.c (Decision-making Time): Participants will take longer to make decisions when given environment-based justifications, compared with policy-based justifications and no justification, as environment-based justification includes more contextual information, which promotes active

thinking patterns.

H2: Subjective Hypotheses

H2.a (Mental Load): Participants will report lower mental load when given policy-based justifications, compared with environment-based justifications, since environment-based justifications have more information to process, and compared with no justification, as people tend to report higher workload when interacting with systems behaving inexplicably [193].

H2.b (Trustworthiness): Participants will rate the system as more trustworthy and reliable when given environment-based justifications, compared with policy-based justifications and no justification, as environment-based justification provides more transparency and contextual information, which will result in participants feeling like they understand the decision-making process.

H2.c (Perceived Intelligence): Participants will rate the system as more intelligent when given environment-based justifications, compared with policy-based justifications and no justification, also due to the transparency into the decision-making process provided by environment-based justifications.

H2.d (Justification Interpretability): Participants will rate environment-based justifications as more interpretable, informative, and helpful for decision-making compared to policy-based justifications, due to the extra information provided by environment-based justifications.

6.6 Experimental Evaluation

We investigate the preceding hypotheses regarding the effects of different types of justification on participants through an IRB-approved human-subjects study.

6.6.1 Experimental Design

We conducted a 5x1 between-subjects experiment using Amazon Mechanical Turk to evaluate the four types of justifications introduced above, alongside a control condition that did not include justifications, in the experimental domain described in Section 6.3 (Fig. 6.1-top). The participants'

goal was to explore a maze and find as many buried treasures as they could in a limited number of turns. Participants were assisted in their task by a team of autonomous drone teammates who simultaneously explored the maze and provided constantly-updating recommendations to the human based on their own noisy sensor readings. The VOI-based framework for strategic justification timing described in Section 6.4 determined when justifications should be provided to participants. The type of justifications were determined by experimental condition: 'global policy', 'local policy', 'global environment', 'local environment', or 'no justification' (control).

6.6.2 Rules of the Game

Participants played two rounds of the game with the goal of digging up as many of the 25 treasures hidden throughout an 18x27 maze grid as they could in a period of 60 turns. Each turn, participants could choose either to move to any available adjacent grid square, or to dig on the square they currently occupied to earn one treasure if one was located there. A team of AI-controlled drones explored the grid autonomously, moving multiple tiles in a turn and taking noisy treasure-detecting measurements of every tile flown over. These readings were used to update both their PMF and the guidance they provided to the participant. The guidance took the form of a green line with an orange 'X' at the end, indicating where the drones thought the participant should dig next (see Fig. 6.4), which participants could choose to follow or not. Whenever a justification was triggered by our framework, the prior path recommendation was shown in gray, with the rest of the justification depending on condition (see Fig. 6.3).

6.6.3 Study Protocol

The experiment was run in several batches with randomly determined condition, using Amazon Mechanical Turk to crowd-source participants. High quality participants were targeted by filtering for high numbers of previously approved tasks on Mechanical Turk, as well as approval percentage. Additionally, on top of the base compensation rate of \$3, a bonus of 5¢ per treasure found during the game was paid to further incentivize participant effort towards high performance.

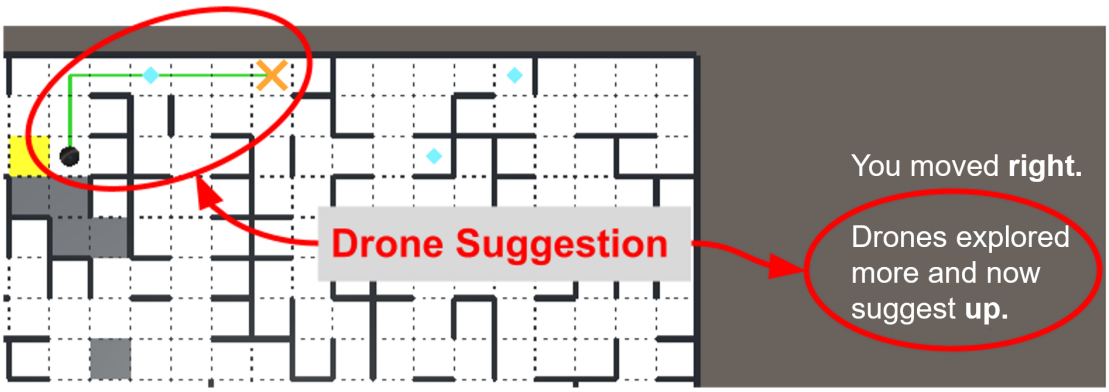


Figure 6.4: Drone guidance is shown as a path overlay and a textual representation of the next suggested move.

After providing informed consent, participants completed a short pre-experiment demographic survey. After reading the rules of the game, participants completed a short comprehension quiz and played a tutorial level to ensure they understood their objective. Next, participants played the two rounds of the game and completed a post-experiment survey which involved a combination of Likert scale and free response questions.

6.6.4 Measurement

The pre-survey collected demographic information about our participants. Out of 104 initial MTurk participants, we removed 13 from data analysis for either failing to locate a single treasure during the game or for repeatedly spending excessive time inactive without inputting a move, indicating lack of understanding of or concentration towards the game, respectively. This left 91 participants (51 males, 37 females, and 3 who did not specify gender) with ages ranging from 23 to 72 years old ($M = 40.99; SD = 11.80$). 39.6% of participants reported working in a STEM field, and 69.2% of participants reported having received a bachelor's degree or higher. 19 participants each ran the 'global environment' and 'no justification' conditions, 18 each ran the 'global policy' and 'local policy' conditions, and 17 ran the 'local environment' condition.

We collected a number of objective measures from participant gameplay, including:

- **Targets Found**: The total number of treasures discovered.

- **Compliance Rate**: The percentage of moves taken by users that matched the recommendations provided by the system.
- **Compliance Rate During Justification**: The percentage of moves taken by users that matched the recommendations provided by the system, on turns when justifications were provided. Note that in the control condition 'no justification', although justifications are never offered, we still collect this measure by applying the same VOI-timing algorithm but never acting on it.
- **Time Per Move**: The average time taken per move.
- **Time Per Move During Justification**: The average time taken to make decisions when justifications were provided.

For subjective measures, we administered a post-experiment questionnaire to participants after completing the treasure hunt task. The questionnaire was developed using well-established metrics from the fields of robotics and explainable AI, including the Trust in Automation Survey [188], the Interpretability and Decision-Making Surveys for XAI metrics [8, 107, 108], the Stress and Workload (NASA-TLX) [190], and the Perceived Intelligence (Godspeed Questionnaire) [283]. Participants were asked to rate their opinions on the guidance provided by the agent using 7-point Likert-scale items. Based on these questionnaires, we identified four key concepts to validate our hypothesis: **Trust, Justification Interpretability, Workload, and Perceived Intelligence**.

To determine these constructs, we used principal component analysis to extract latent factors from the above mentioned scales and calculated the factor loading matrix using varimax rotation. We identified items that could be combined to create concept scales with a correlation cutoff point of $r \geq 0.6$ to the factor matrix [284] which resulted in the scales presented in table 6.2.

Table 6.2: Subjective Scale Measure Items.

Trust (Cronbach's $\alpha = 0.95$)
1. I am confident in the system
2. The system is dependable
3. The system is reliable
4. I can trust the system
Justification Interpretabilty (Cronbach's $\alpha = 0.94$)
1. I found the justifications to be complete and understandable.
2. I was able to adapt better to the game due to the justifications provided.
3. I found the justifications to be sufficient for making decisions.
4. I found that the justifications were informative during the game.
5. The justifications were useful.
6. I understand why the system used specific information in its justifications.
7. I understood how the system arrives at its answer.
8. I understood the systems reasoning.
9. I could easily follow the justifications to arrive at a decision.
Workload (Cronbach's $\alpha = 0.76$)
1. How mentally demanding was the game?
2. How hurried or rushed was the pace of the game?
3. How hard did you have to work to accomplish your level of performance?
4. How insecure, discouraged, irritated, stressed, and annoyed were you during the game?
Perceived Intelligence (Cronbach's $\alpha = 0.92$)
1. System is Competent
2. System is Knowledgeable
3. System is Intelligent
4. System is Sensible
Likert items are coded as 1 (Strongly Disagree) to 7 (Strongly Agree)

6.7 Results

6.7.1 Objective Analysis

To test our objective hypotheses, we analyzed the various metrics collected during the game using a one-way analysis of variance (ANOVA) with experimental condition as a fixed effect. Post-hoc tests used Tukey's HSD to control for Type I errors in comparing results across each of the four justification types and the control condition.

Our hypotheses expected between-conditions differences to be more pronounced along the axis of policy-based vs. environment-based features, compared with global vs. local features. Hence, we conducted additional analysis using a one-way ANOVA with bucketed results, comparing policy-based justification vs. environment-based justification vs. no justification. Again, post-hoc

significance was determined using Tukey's HSD. The means per condition and per bucket are shown in Tables III and IV below.

	Global Policy	**Local Policy**	**Global Env.**	**Local Env.**	**None**
Compliance Rate*	84.67%^{A}	81.53%	70.65%^{B}	75.48%	70.53%^{B}
Compliance Rate (During Justification)*	56.46%^{A}	54.50%	40.57%^{B}	49.54%	48.52%
Targets Found*	9.28^{A}	8.47$^{A/B}$	7.00$^{B/C}$	7.78	6.32^{C}
Time per Move*	1.30s^{B}	1.40s	2.01s	2.10s^{A}	1.90s
Time per Move (During Justification)*	1.74s^{B}	1.66s^{B}	2.49s	3.39s^{A}	1.85s^{B}

Table 6.3: Means for objective measures across all conditions. Measures with ANOVA significance are indicated by *. Post-hoc significance is shown using letters. Individual means denoted by A are significantly higher than B/C or C. Likewise, A/B is significantly higher than C.

The ANOVA revealed significant effects for both overall compliance rate ($F(4,86) = 3.98$, $p = 0.0052$), and compliance rate during justification ($F(4,86) = 3.09$, $p = 0.020$). Post-hoc analysis for overall compliance rate with Tukey's HSD shows that participants complied significantly more in the 'global policy' condition compared to both the 'no justification' condition ($p = 0.019$), and the 'global environment' condition ($p = 0.020$). Post-hoc analysis of compliance rate during justification found a significantly higher compliance in 'global policy' compared to 'global environment' ($p = 0.016$).

Significance was likewise found in the ANOVA comparing the policy-based, environment-based, and no justification buckets for both overall compliance rate ($F(2,88) = 7.19$, $p = 0.0013$), and compliance rate during justification ($F(2,88) = 4.41$, $p = 0.015$). Post-hoc analysis showed that overall compliance rate was significantly higher for users with policy-based justifications than those with environment-based justifications ($p = 0.0047$), and those with no justification ($p = 0.0062$). Post-hoc analysis of the compliance rate during justification additionally showed a significant effect for policy-based over environment-based justifications ($p = 0.012$). These results serve to **validate**

H1.a (compliance).

Since our experimental domain was associated with a high degree of robot competence, performance in the game (number of targets found) highly correlated with compliance with the drones' suggestions. Using Pearson's correlation coefficient, we verified this relationship (i.e., the more participants chose to follow the guidance, the better they perform) ($r(91) = 0.77$, $p < 0.0001$). The ANOVA showed a statistically significant effect for number of targets found ($F(4,86) = 4.77$, $p = 0.0016$). Post-hoc analysis showed three significant effects. Participants in 'global policy' found more targets than those in 'no justification' ($p = 0.016$), or in 'global environment' ($p = 0.027$). Additionally, those in 'local policy' found significantly more targets on average compared to 'none' ($p = 0.047$).

	Policy Features	**Env Features**	**None**
Compliance Rate*	$83.14\%^{A}$	$73.00\%^{B}$	$70.53\%^{B}$
Compliance Rate (During Justification)*	$55.51\%^{A}$	$44.93\%^{B}$	48.52%
Targets Found*	8.89^{A}	7.38^{B}	6.32^{B}
Time per Move*	1.35s^{B}	2.06s^{A}	1.90s^{A}
Time per Move (During Justification)*	1.70s^{B}	2.93s^{A}	1.85s^{B}

Table 6.4: Means for objective measures across the three condition buckets. Measures with ANOVA significance are indicated by *. Individual means denoted by A demonstrated post-hoc significance over means denoted B.

The ANOVA per bucket also revealed significance ($F(2,88) = 8.46$, $p = 0.0004$). Post-hoc analysis found that policy-based justifications led to better user performance in the game, compared with both no justification ($p = 0.0005$), and environment-based justifications ($p = 0.018$). These results serve to **validate H1.b (performance)**.

The timing measures, related to the latent measure of participant thinking load, had significant effects both for time per move ($F(4,86) = 3.71$, $p = 0.0078$) and time per move during

justification ($F(4,86) = 3.74$, $p = 0.0075$). Post-hoc analysis for time per move showed that participants in the 'local environment' condition took significantly more time to take their moves compared to 'global policy' ($p = 0.030$), but not significantly more time compared to 'local policy' ($p = 0.089$). Additionally, while there was no significant effect for 'global environment' taking longer on average than 'global policy', further exploration may be merited in future work ($p = 0.063$). Post-hoc analysis for time per move during justification showed three significant effects, with 'local environment' taking more time than 'local policy' ($p = 0.016$), 'global policy' ($p = 0.022$), and 'no justification' ($p = 0.033$).

In the bucketed analysis of timing, the ANOVA showed significance in both time per move ($F(2,88) = 7.44$, $p = 0.0010$), and time per move during justification ($F(2,88) = 5.91$, $p = 0.0039$). Post-hoc analysis of time per move showed that, with environment-based justifications, participants took significantly longer than with policy-based justifications ($p = 0.0009$). Interestingly, no justification similarly had a significant effect, taking longer than policy-based justifications ($p = 0.047$). This shows that despite the added cost of attending to justifications, participants were able to take their moves faster on average in the policy-based justification conditions. Similarly, post-hoc analysis of time per move during justification showed that environment-based justifications took significantly higher time than both policy-based justifications ($p = 0.0049$), and no justifications ($p = 0.050$). These results serve to **validate H1.c (decision-making time)**.

6.7.2 Subjective Analysis

We conducted similar analysis to test our subjective hypotheses, running one-way ANOVAs fixed by both experimental condition, as well as bucketed by the feature class seen during justification (policy-based, environment-based, or no justification). Post-hoc significance was determined using Tukey's HSD. In the case of the scale for justification interpretability, the Likert-scale questions asked referred specifically to justifications, so was limited only to the four experimental conditions that possessed justifications, excluding the control.

Of the 91 participants with usable gameplay data, an additional five failed basic attention-

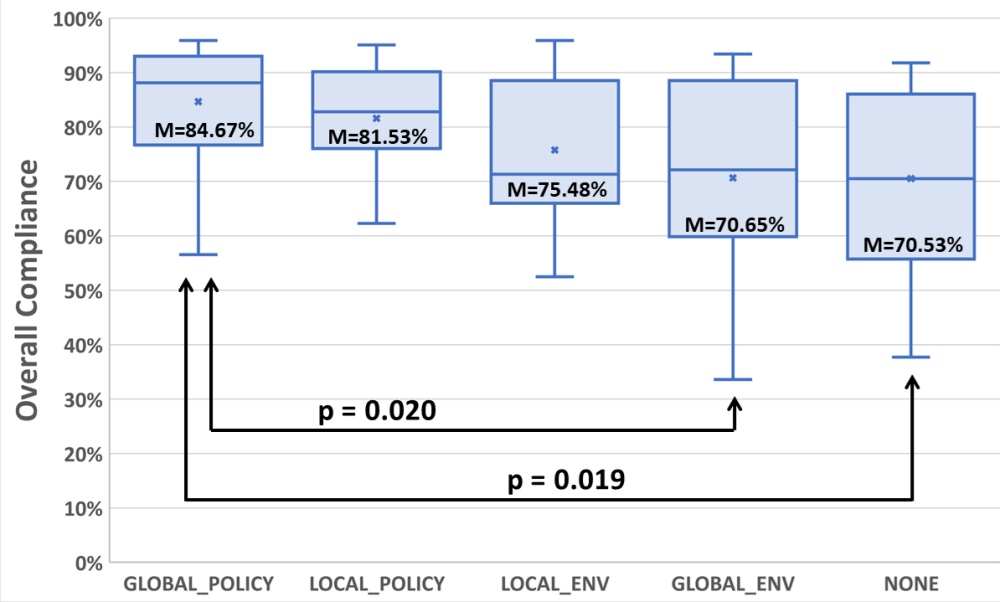


Figure 6.5: Compliance rate by condition, with means and post-hoc significance shown.

	Global Policy	**Local Policy**	**Global Env.**	**Local Env.**	**None**
Workload	3.40	3.67	4.05	3.63	4.24
Trust	4.15	3.94	5.23	4.80	4.87
Perceived Intelligence	4.59	4.88	5.73	5.16	5.27
Justification Interpretability*	4.32^{B}	4.24^{B}	5.40^{A}	4.96	N/A

Table 6.5: Means for subjective measures across all conditions. Measures with ANOVA significance are indicated by *. Individual means denoted by A demonstrated post-hoc significance over means denoted B.

check questions in the survey. Post-hoc analysis of survey responses showed six further outliers, with significantly lower internal consistency among related survey question answers than other participants, appearing more like random clicking than coherent responses. Removal of those 11 participants left us with the surveys of 80 participants for subjective analysis.

There were no statistically significant differences on the *Workload* scale, either in the ANOVA

	Policy Features	Env Features	None
Workload	3.53	3.85	4.24
Trust*	4.05^{B}	5.03^{A}	4.87
Perceived Intelligence*	4.73^{B}	5.47^{A}	5.27
Justification Interpretability*	4.28^{B}	5.20^{A}	N/A

Table 6.6: Means for subjective measures across all conditions. Measures with ANOVA significance (or Student's t-test significance, in the case of Justification Interpretability) are indicated by *. Individual means denoted by A demonstrated post-hoc significance over means denoted B.

with experimental condition as its fixed effect or between the bucketed classes of policy-based, environment-based, and no justification. Therefore, the hypothesis **H2.a (mental load) is inconclusive**.

The condition-wise ANOVA of the *Trust* scale also did not reveal a significant effect ($F(4,75) = 2.33$, $p = 0.064$), but the bucketed ANOVA for *Trust* did reveal significance ($F(2,77) = 4.29$, $p = 0.017$). Post-hoc analysis with Tukey's HSD revealed that environment-based justifications were rated as significantly more trustworthy than policy-based justifications ($p = 0.019$). However, no effect was found between environment-based justification conditions and no justification, meaning this result serves to **partially validate H2.b (trustworthiness)**.

Likewise, while the per condition ANOVA of the *Perceived Intelligence* scale was not significant ($F(4,75) = 2.23$, $p = 0.073$), the feature-class bucketed ANOVA for *Perceived Intelligence* was ($F(2,77) = 3.30$, $p = 0.042$). Post-hoc analysis showed that the drone teammates using environment-based justifications were rated as significantly more intelligent than the drone teammates using policy-based justifications ($p = 0.038$). Again, no effect was found between environment-based conditions and no justification, meaning this result serves to **partially validate H2.c (perceived intelligence)**.

Lastly among the subjective scales, the ANOVA for the *Justification Interpretability* scale

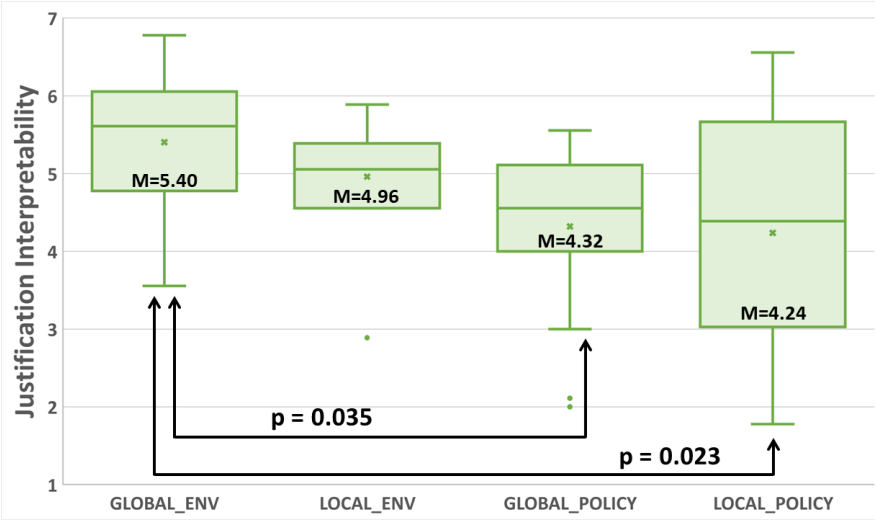


Figure 6.6: Rated interpretability of justifications by condition, with means and post-hoc significance shown.

did reveal significance when fixed by experimental condition ($F(3,59) = 3.94$, $p = 0.013$). Post-hoc analysis revealed that the justifications in the 'global environment' condition were rated as significantly more interpretable and informative when compared to the justifications from both the 'local policy' condition ($p = 0.023$), and the 'global policy' condition ($p = 0.035$).

There was an additional significant effect for the data bucketed by feature class for the *Justification Interpretability* scale. Since this scale specifically compares justifications, the 'no justification' bucket is excluded from analysis, and the data is compared using a simple one-tailed t test, where the justifications from environment-based justification conditions are rated as significantly more interpretable compared to justifications from policy-based justification conditions ($t(61) = -3.35$, $p = 0.0007$). These results serve to **validate H2.d (justification interpretability)**.

6.8 Recommendations & Potential Applications

6.8.1 Recommendations for Justification Design

In this section, we summarize the main findings and implications drawn from the results of our user study on the utility of justification in human-robot interaction.

6.8.1.1 High Robot Competence or Low Human Competence: Use Policy-based Justifications

Policy features are highly comprehensible to human teammates, as the information is packaged such that users can compare the end results of the robot's decision making. The information is highly abstract, and is framed taking the human teammate's own utility into account. There is little room to think critically about or question the accuracy of policy-based counterfactual justifications, which resulted in a high level of persuasiveness in our study (we saw that policy-based justifications led to significantly higher compliance when compared with environment-based or no justifications). In our user study with highly competent robot teammates, participants were more successful in accomplishing their task when presented with this style of low transparency, easily comprehensible justification.

It is important to note that if the robotic agent were not giving competent recommendations, participants would likely have performed significantly worse due to their over-reliance on a low-quality decision support system. Policy-based justification could result in over-reliance and dependence on the system, causing passive thinking patterns [253] where the human cedes effective control of decision-making entirely to the robot agent. In cases of low robot competence, this would lead to a large number of Type I errors where users accept low-quality advice from the system [285, 286]

Therefore, during human-robot teaming scenarios or domains where you would expect the quality of robotic guidance to be fairly high relative to a human operating by themselves, policy-based justification should be used, increasing human teammate compliance, making them a more predictable member of a multi-agent team. This would significantly improve the planning system's ability to optimize over all agents, since the innate uncertainty associated with accounting for human decision making would be greatly reduced [287, 288]. Policy-based justification can also be suitable when the human needs to make snap decisions in time-critical situations.

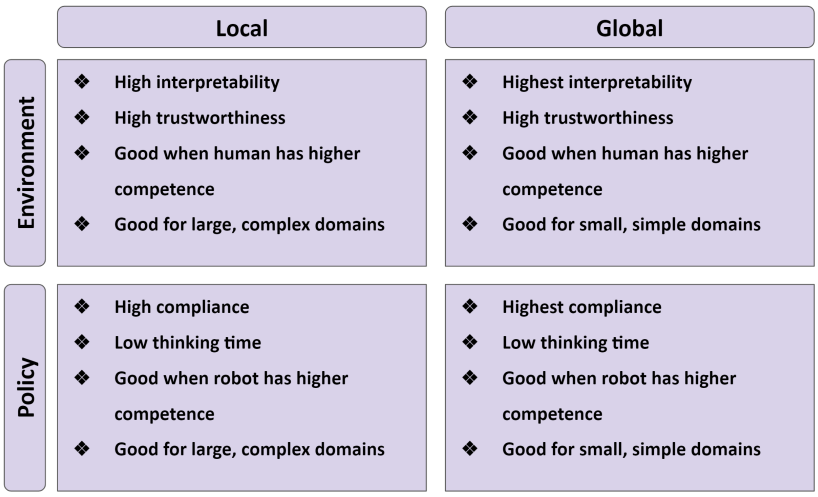


Figure 6.7: A taxonomy of the usefulness of each justification type.

6.8.1.2 Low Robot Competence or High Human Competence: Use Environment-based Justification

Environment-based features provide highly interpretable, highly contextual information, and are well-suited for representing uncertainty. They push human teammates towards a more active thinking pattern, which is more analytical, deliberate, and rational [253]. Humans tend to view this type of justification as more of a tool, compared with the more abstracted policy-based justifications. This can lead to better-informed decision making and more successful adaptation to uncertain situations. In our study, we observed that environment-based justifications during changes of recommendation were associated with significantly more thinking time than policy-based or no justifications. What's more, participants rated robotic agents using environment-based justifications as the most trustworthy, and environment-based justifications themselves as the most informative, interpretable, and helpful for their decision-making process.

This added transparency and increased information content comes at the cost of being more demanding and time-consuming to parse, leading to slower decisions. Additionally, environment features are able to be interpreted in any number of ways by different human agents, which often leads to highly variable, independent human behavior [193]. This leads to a significantly lower compliance rate when compared with policy-based justifications. If environment-based justifications

were deployed in a domain with a high relative competence of robot-provided guidance, there would be a large number of Type II errors made, whenever users reject the high-quality advice of the robot. Therefore, in scenarios where the human teammate brings expertise in their decision-making that is hard to match with the automated guidance of a collaborative robot, environment-based justifications are more appropriate.

Focusing on the other axis of our 2x2 justification characterization, in our study we generally found that the use of global features outperformed the local features on the respective measures that policy-based and environment-based justifications excelled at. For instance, 'global policy' had the highest user compliance rate and performance, and 'global environment' had the highest perceived interpretability. We posit that this is likely related to the short-term nature of the interaction in our evaluation domain. In longer lasting, more complex domains, local features may prove may beneficial, as they can help prevent the human teammate from being overwhelmed by excess information. More research is needed to confirm this. We summarize the characteristics and suitable use cases of each justification type in Fig. 6.7.

6.8.2 Potential Application: AR-based Spatial Navigation

To illustrate the application of these synthesized justification design principles, we present a concept of how they might by implemented in a real-world decision support system embedded in an augmented reality (AR) interface (similar to [193]). Since our framework and results are drawn from a partially observable, multi-goal search task, we designed this interface for domains that share these characteristics, such as search and rescue, radiological device recovery, or explosive ordnance disposal. However, since the features tested were derived from general xAI principles, it is likely that the taxonomy presented in Fig. 6.7 is more broadly applicable to a wide range of human-robot collaborative tasks, though further research is needed to confirm this.

Humans using this interface explore an environment searching for hidden targets. Meanwhile, a drone teammate conducts its own exploration of the environment, using its sensors to update its model of where it believes the hidden targets are likely to be. The drone continually provides

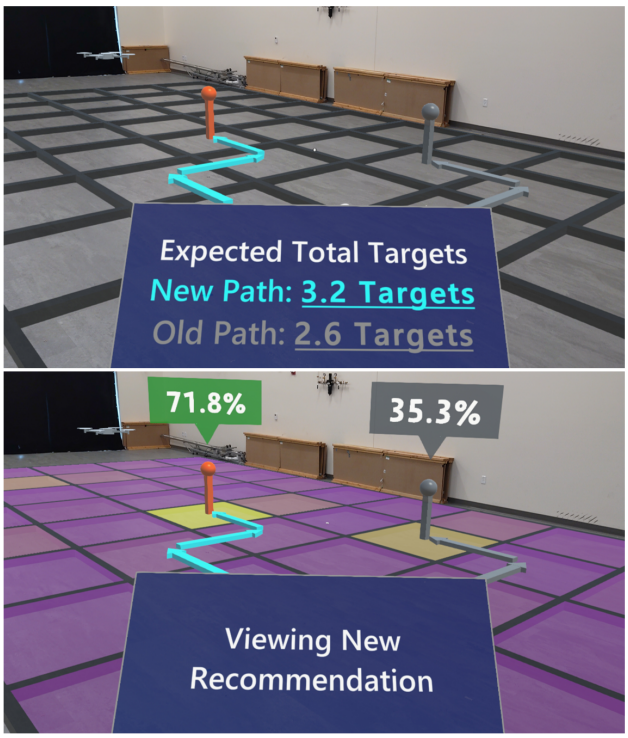


Figure 6.8: Top: AR-based policy justification. Bottom: AR-based environment justification.

navigation guidance to the human, aiding them in the task of locating as many targets as possible in a limited amount of time. Whenever justification is triggered by a significant change in guidance, one of two justification modules is chosen, depending on the drone's current confidence in the quality of that guidance.

AR-based Policy Justification. In regions of high drone confidence, a policy justification is triggered (Fig. 6.8 Top). The AR interface renders the current guidance in the form of a colored arrow and pin directly overlaid onto the environment, telling the human where the drone thinks they should go and search next. The guidance from the prior time step is rendered as a gray arrow and pin. In addition to these paths, a counterfactual natural language description is provided as justification on the user's AR-based menu, showing the difference in expected utility of taking the new path in contrast to the old path.

AR-based Environment Justification. In regions of low drone confidence, an environment justification is triggered (Fig. 6.8 Bottom). In addition to rendering the current and previous paths as seen in the policy justification, the AR interface renders the drone's current PMF as a

heatmap overlaid onto the environment, using a gradient from purple to yellow to represent low and high chances of finding a target, respectively. Two AR-based pins are rendered over the current and prior targets, showcasing the local PMF values at each location. Users are able to view the PMF and pins from the prior timestep to visualize how the environment features changed to lead to a changed recommendation, providing a justification for taking the new path as opposed to the old path.

The task in this implementation has similar dynamics to the treasure hunt game, though lifted into a 3D, real world domain. Although the interface pictured in Fig. 6.8 is shown at the scale of a large room, the same type of visualization could be spatially expanded to large outdoor environments to serve as a viable interface for real-world drone assisted target-finding tasks.

6.9 Conclusion

In this chapter, we highlighted the value of strategic timing for robot-provided explanations that serve as justifications during instances of mismatched expectations in the context of decision-support for human-robot teaming (e.g., when an agent's recommendation is unexpected or confusing). A justification provided in this context aims to convince the human teammate of the utility of the previously difficult-to-interpret recommendations. Our work contributes answers toward two fundamental questions at the intersection of explainable AI and human-robot teaming: 1) When are justifications most impactful and useful? And 2) What information should be presented in those justifications to improve human teammate decision-making and behavior?

We propose a novel value of information-based framework to determine when a decision-support system should provide justifications to a human collaborator, such that a balance is struck between informativeness, and avoiding habituation and excess cognitive load. We validated the proposed framework through an expert-feedback case study, demonstrating the usefulness of justifications when they are timed appropriately. We also present a characterization of four types of counterfactually generated justification, drawing from a taxonomy established in explainable AI literature: **global policy**, **local policy**, **global environment**, and **local environment**. The

justification types were evaluated in an online human subjects study ($n = 91$) involving a collaborative, partially observable search task alongside robot teammates.

We show that robots providing policy-based justification led to higher compliance and faster decision-making. We additionally show, in contrast, that robots providing environment-based justification led to higher subjective ratings of interpretability, intelligence, and trustworthiness of the robot teammates.

Based on our experimental findings, we offer actionable recommendations for operationalizing these results into decision-support systems that prioritize explainability and foster appropriate trust and reliability. We additionally demonstrate how these sythesized design principles can be applied to a real-world decision-support system with a concept augmented-reality interface. Justifications should be user-centric, taking into consideration the relative competence of human and robotic agents, the user's expectations of the robot, and how different types of justification can influence user thinking patterns and performance.

Chapter 7

Conclusion

"We shall not cease from exploration, and the end of all our exploring will be to arrive where we started and know the place for the first time."

— T.S. Eliot, *Four Quartets*

This dissertation presents novel contributions toward improving human-robot teaming through multimodal communication and explanations for transparency and mental model synchronization. We primarily focus on endowing autonomous agents with the capabilities to explain their decision-making rationale using multiple modalities (natural language and visual), coach and influence human teammates' behavior using explanations, and leverage justification for successfully convincing and mediating trust in human-robot interaction, informed by insights from cognitive psychology and human factors. In this conclusion chapter, I summarize the contributions of each work and present key takeaways. Finally, I conclude with the future implications of this work and future research opportunities in the areas of explainable robotic coaching, explainability for multi-agent reinforcement learning, bi-directional communication in human-machine teaming, and the role of large language models in decision support systems.

7.1 Summary of Contributions and Key Takeaways

Reward Augmentation and Repair through Explanation (RARE): One of our goals in this thesis was to transform robots into competent coaches, using explainable AI to establish shared mental models amongst teammates. Therefore, we developed a novel robot coaching

framework called Reward Augmentation and Repair through Explanation (RARE) [20]. The core functionality is as follows: 1) RARE infers the collaborator's task understanding, estimating their reward function using Hidden Markov Models, 2) it identifies missing components of the reward function via a Partially Observable Markov Decision Process, and 3) it provides natural language explanations to facilitate reward function repair, improving task comprehension.

Through a between-subjects user study, we evaluated the viability and effectiveness of RARE using a collaborative color-based sudoku game, where users teamed with an autonomous robotic arm. The experiment compared two study conditions based on the content provided during a robot interruption: control and justification conditions. In the control condition, the robot provided a simple indication that the user was about to make a mistake leading to task failure. In the justification condition, the robot included additional information explaining the reason for the potential failure.

We found statistically significant support across subjective measures to validate the hypothesis that participants found robots more helpful, useful, and intelligent when they provide justifications. Objectively, we observed more game terminations (irreversible mistakes) during the control condition than the justification condition (80% vs. 20%). Our exit survey showed that people did not trust the robot when it intervened without further explanation (e.g., the reason for game termination), indicating justification is likely necessary when a robot corrects users or recommends alternate actions.

Single-shot Policy Elicitation for Augmenting Rewards (SPEAR): RARE corrects a single instance of suboptimal human action at a time, which can be tedious and time-consuming for human collaboration. Furthermore, RARE does not consider the recipient's world model, leading to the generation of uninterpretable or cognitively demanding explanations. Consider an emergency evacuation scenario, where an agent is tasked with guiding people safely out of a building. Someone visiting for the first time may not know how to change their evacuation plan when told, "There's a fire near Conference Room 3", but may be able to adapt their plan if told "the north half of the building is on fire".

Thus, we proposed Single-shot Policy Explanation for Augmenting Rewards (SPEAR) [289], a novel optimization algorithm that uses semantic explanations derived from combinations of planning predicates to augment agents' reward functions, driving their policies to exhibit more optimal behavior. Predicates are pre-defined boolean state classifiers (as found in traditional STRIPS planning [178]) with associated string explanations, as shown in Figure 1-left. Prior work solves natural language generation as a set cover problem to find the smallest logical expression of predicates, but their solution is of exponential runtime, preventing its use in most real-world problems [92]. We solve the minimum set cover using a novel integer programming formulation and policy elicitation to improve the collaborator's task performance.

We experimentally validated our algorithm's policy elicitation capabilities in practically grounded applications. Our approach outperforms prior work [92] by orders of magnitude. We also conducted a series of human subjects studies to demonstrate the utility of these explanations for both expert and novice users. Our results indicate that these relaxed reward-based explanations not only enhance individuals' policies but also decrease cognitive load and improve decision-making, all while preserving interpretability. Additionally, we show that these explanations provide insights into the robot recommender's decision-making process, foster a better understanding of tasks, and promote active thinking patterns in users, while also facilitating the desired correction of policies.

Natural Language Communication for Robot Skill Learning and Repair: We introduce a human-in-the-loop algorithm, Plan Augmentation and Repair through SEmantic Constraints (PARSEC), that facilitates constraint annotation by novice users using natural language for motion planning problems through a novel hierarchical semantic process for robot skill learning and repair.

PARSEC combines the ease of using natural language with constraint motion planning, enabling novice users to perform online robotic skill corrections and personalization, making working and collaborating with robots more accessible and safe. By utilizing a semantic hierarchy, PARSEC allows users to quickly and effectively select constraints using natural language to correct faulty behavior or adapt skills to their preferences. Through a human subjects case study, we demonstrate that PARSEC efficiently finds corrective constraints that match the user's intent, providing a path

for novice users to leverage constrained motion planning combined with human-in-the-loop skill training.

AR-based Visual Guidance for Multi-agent Reinforcement Learning: Semantic explanations are not well suited for certain scenarios, especially those involving high uncertainty, requiring the portrayal of multiple competent hypotheses as plans change based on new observed information (i.e., partially observable domains). For these continually evolving domains, visual information representation is ideal [222], motivating our subsequent work on AR-based visual guidance called MARS (Min-entropy Algorithm for Robot-supplied Suggestions) [193].

MARS consists of a planning algorithm for uncertain environments, informing the generation of proactive visual recommendations. Environmental uncertainty is characterized as a dynamically-updating probability mass function (PMF), a common practice across various classes of search task [223, 224, 225]. The PMF serves as a shared utility function common to all agents (both human and autonomous), providing insight into the agent's policy. This PMF is utilized by two separate Markov Decision Processes (MDPs); one for autonomous agents, and another for generating assistive guidance for the human teammate. MARS solves both of these MDPs via online reinforcement learning to get optimal policies for autonomous agents and action recommendations for human teammates respectively. We also provided a characterization of two distinct AR-based visual guidance modalities: prescriptive guidance (visualizing recommended actions) and descriptive guidance (visualizing state space information to aid in decision-making).

We evaluated the utility of our visual guidance modalities and the effectiveness of the MARS algorithm through a within-subjects human study using a human-robot collaborative analogue of the PC game Minesweeper, played using a HoloLens 2 AR headset. Participants experienced three conditions based on the type of visual guidance given to the human teammate as informed by sensor readings from a virtual drone: 1) prescriptive guidance, 2) descriptive guidance, and 3) a combination of prescriptive and descriptive guidance. We found statistical significance supporting our hypothesis that combining visual insight into environmental uncertainty (descriptive guidance) with robot-provided action suggestions (prescriptive guidance) improved trust, interpretability, and

performance, and made human collaborators more independent.

Justification Timing and Characterization of Justification Types: In the MARS study, participants were frustrated by the system's unexpected behaviors, such as sudden path changes. This unpredictability stemmed from policy optimization in uncertain situations, leading to varied trust levels in the system; some participants over-trusted it while others under-trusted it. Participants perceived this emergent behavior as unconfident and expressed a desire for explanations, along with a mechanism to judge the quality of recommendations, echoing our previous findings [20]. Therefore, in this work, we aimed to leverage multimodal explanations to serve as justifications, defining a justification as an explanation of an action or suggestion, timed strategically to align with a mismatch in expectation between agents. This motivation led us in [175] to evaluate when justifications are most impactful and what information they should include to enhance human decision-making.

In this work, we developed a novel mathematical framework grounded in the value of information theory to identify the optimal timing for a robot to justify its recommendations to a human teammate. This framework was validated through an expert-feedback study, revealing that our strategic timing for justifications received the highest average rating for perceived usefulness compared to constant or timed-interval justifications.

We also introduced a methodological characterization of four distinct justification types: global policy, local policy, global environment, and local environment. These types were evaluated through an online human-subjects study. Our findings revealed that policy-based justifications promote higher compliance and quicker decision-making, while environment-based justifications enhance perceptions of a robot's interpretability, intelligence, and trustworthiness. Based on these insights, we recommended using policy-based justifications when the robot has high competence or the human has low competence. Conversely, environment-based justifications are best suited for situations with a less competent robot or a highly competent human.

7.2 Implication for Future Work

A major focus of this thesis is on improving human-machine teaming through explainable AI techniques, specifically via mental model alignment. This involves the concept of explainable coaching, multimodal communication, and leveraging insights from psychological research to enable appropriate trust and influence on human teammates. The research and insights presented in this work enable the following avenues for further exploration:

Explainable Robotics Coaching: Future work in this area involves exploring the implementation of explainable coaching or skill coaching applied to specific domains and applications. For instance, teaching a novice to operate a drone [290], applying skill coaching in rehabilitation or elderly care [291], and enabling knowledge testing and modification in educational contexts to encourage learning [292]. One crucial factor here is personalization, which includes building personal learning models for each learner and enabling intuitive queries for personalized feedback and coaching [293]. For example, identifying specific mistakes and providing targeted feedback can significantly enhance learning outcomes. Some recent research in this area leverages pedagogical theories and large language models to facilitate human-like interactions, providing live feedback and fostering a more engaging learning experience [290, 294]. One promising approach has been to use concepts such as curriculum learning or scaffolding, where knowledge is built incrementally, and learners are tested on their understanding and retention [295].

Additionally, novel interfaces such as the Meta Quest 3, which offers extensive AR and VR capabilities, can enhance multimodal learning experiences [296]. Furthermore, recent research has focused on designing innovative interfaces that improve communication and facilitate learning, which will play a crucial role. For example, Dhat et al. present a web software package that facilitates the integration of 3D mice into robot manipulation interfaces by offering configurable input signal processing schemes to enhance usability and an interactive visual representation of the device's 6DOF input for operator familiarization and visual assistance during teleoperation [297].

Explainable AI for Multi-Agent Reinforcement Learning: Another avenue for fu-

ture research is the application of explainability in multi-agent reinforcement learning (MARL). Explainability in RL is already challenging, and it becomes even more complex with multi-agent systems. For example, one challenge in MARL is measuring the contribution of individual agents, or decoupling individual agents' policies. The best approach currently is to calculate Shapley values for each agent, which grow exponentially with the number of agents [298]. However, Shapley values are not easily comprehensible to novice users, making them less useful in human-centric applications such as warehouse management or search operations with heterogeneous teams.

Therefore, there is a need for more innovative approaches to ensure that human users can effectively manage and interact with these systems while maintaining shared situational awareness and transparency. For example, this includes effectively summarizing each agent's capabilities, allowing end users to dynamically reallocate tasks or resources based on evolving situations, and enabling human teammates to introduce new strategies and preferences [294, 299, 300]. Additionally, developing novel interfaces for communication and interaction can significantly enhance the usability of MARL systems. These interfaces could provide intuitive visualizations and interactive tools that help human operators better understand and manage the behaviors and strategies of multiple agents.

Bi-direction Communication in Human-Machine Teaming: The majority of this thesis focuses on agents communicating with humans (Chapters 3, 5, and 6), while Chapter 4 explores how asking questions to end users can enable them to provide appropriate constraints for robot skill learning and repair. One active research area is enabling complete bi-directional communication. Most research in explainable AI focuses on one-way communication, where a robot explains a single instance of failure or decision-making [24, 301]. However, social science research shows that people prefer explanations similar to human explanations, which tend to be contrastive, selective, and social [30, 59]. Engaging in a more dialogical exchange allows for thorough testing of the alignment between the robot's and the human's mental model, which is crucial in human-machine teaming, especially in partially observable scenarios with mixed-initiative teams. Mixed-initiative teams are flexible groups that allow agents, such as humans or robots, to contribute

their best skills and knowledge at the most appropriate time.

Recent research into interactively explaining robot policies shows promising results for improving transparency in underlying behaviors, but these are still in the initial stages with limited applications where human users do not have full autonomy to inspect or query as they wish [294, 300, 302]. Therefore, there is a need for developing innovative approaches and scenarios that enable bi-directional communication between humans and robots. For example, a robot negotiating with humans when the user provides a suboptimal plan, or humans adapting to robot plans once they understand them [303]. Furthermore, recent techniques such as diffusion models and large language models represent a promising research direction for more user-friendly accessibility in the context of human-robot teaming, allowing people to inject preferences and knowledge at the time of inference to guide robot policies [304, 305].

LLMs and Decision Support Systems: The use of large language models (LLMs) in decision support systems is a promising area of future work, as these models facilitate natural conversations between agents and users. For example, DeepMind's RT-2 [306] presents a "vision-language-action" model that enables a human to provide natural language instructions to a robot for manipulation tasks. Similarly, the other models that combine robotics with LLMs include Apple's Large Language Model Reinforcement Learning Policy (LLaRP) [307] and SayPlan [308]. While these systems are promising, they tend to hallucinate and do so confidently, leading to overreliance and overtrust, which can be detrimental in safety-critical applications.

Addressing these issues involves introducing friction when the models might be wrong and ensuring they quantify and communicate uncertainty to end-users, preventing blind trust in these systems. Some of our work has looked into how we can provide friction and encourage critical thinking by leveraging different modalities of explanations [193]. Promising research in this direction involves developing techniques to make users more actively engaged with these decision support tools rather than being overly reliant on them. Furthermore, there needs to be research towards enabling verifiable plans and safety guarantees while working with these types of systems [309, 310].

Additionally, LLMs have shown potential in theory of mind (ToM), which is beneficial for

human-robot teaming applications [311]. A robot equipped with a ToM capacity can collaborate more effectively with its human teammate by inferring what the human knows and planning accordingly. This capability can better model the user's mental state, be effective in personalization, and generate synthetic data. It can also be used to improve value alignment between human and robot teams [312].

Bibliography

[1] David Gunning. Explainable artificial intelligence (xai). Defense Advanced Research Projects Agency (DARPA), nd Web, 2, 2017.

[2] David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang. Xai—explainable artificial intelligence. Science robotics, 4(37):eaay7120, 2019.

[3] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608, 2017.

[4] Partha Pratim Ray. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. Internet of Things and Cyber-Physical Systems, 3:121–154, 2023.

[5] Davide Castelvecchi. Can we open the black box of ai? Nature News, 538(7623):20, 2016.

[6] Matthew Arnold, Rachel KE Bellamy, Michael Hind, Stephanie Houde, Sameep Mehta, A Mojsilović, Ravi Nair, K Natesan Ramamurthy, Alexandra Olteanu, David Piorkowski, et al. Factsheets: Increasing trust in ai services through supplier's declarations of conformity. IBM Journal of Research and Development, 63(4/5):6–1, 2019.

[7] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pages 1135–1144, 2016.

[8] Sebastian Wallkötter, Silvia Tulli, Ginevra Castellano, Ana Paiva, and Mohamed Chetouani. Explainable embodied agents through social cues: a review. ACM Transactions on Human-Robot Interaction (THRI), 10(3):1–24, 2021.

[9] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". AI magazine, 38(3):50–57, 2017.

[10] Nicholas Diakopoulos. Accountability in algorithmic decision making. Communications of the ACM, 59(2):56–62, 2016.

[11] U.S. Congress. S.3572 - 117th congress (2021-2022): A bill to amend the immigration and nationality act to alter the definition of "conviction", and for other purposes., 2022.

[12] Aaquib Tabrez, Matthew B Luebbers, and Bradley Hayes. A survey of mental modeling techniques in human–robot teaming. Current Robotics Reports, pages 1–9, 2020.

[13] Manisha Natarajan, Esmaeil Seraj, Batuhan Altundas, Rohan Paleja, Sean Ye, Letian Chen, Reed Jensen, Kimberlee Chestnut Chang, and Matthew Gombolay. Human-robot teaming: grand challenges. Current Robotics Reports, 4(3):81–100, 2023.

[14] Bradley Hayes and Brian Scassellati. Challenges in shared-environment human-robot collaboration. learning, 8(9).

[15] Nancy J Cooke, Eduardo Salas, Janis A Cannon-Bowers, and Renee J Stout. Measuring team knowledge. Human factors, 42(1):151–173, 2000.

[16] Philip R Cohen, Hector J Levesque, José HT Nunes, and Sharon L Oviatt. Task-oriented dialogue as a consequence of joint activity. Proceedings of PRICAI-90, pages 203–208, 1990.

[17] Guy Hoffman, Tapomayukh Bhattacharjee, and Stefanos Nikolaidis. Inferring human intent and predicting human action in human–robot collaboration. Annual Review of Control, Robotics, and Autonomous Systems, 7, 2023.

[18] Tathagata Chakraborti, Sarath Sreedharan, Yu Zhang, and Subbarao Kambhampati. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. arXiv preprint arXiv:1701.08317, 2017.

[19] Tathagata Chakraborti, Yu Zhang, David E Smith, and Subbarao Kambhampati. Planning with resource conflicts in human-robot cohabitation. In Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, pages 1069–1077, 2016.

[20] Aaquib Tabrez, Shivendra Agrawal, and Bradley Hayes. Explanation-based reward coaching to improve human performance via reinforcement learning. In 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 249–257. IEEE, 2019.

[21] Tathagata Chakraborti, Sarath Sreedharan, and Subbarao Kambhampati. The emerging landscape of explainable automated planning & decision making. In International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence, 2021.

[22] Tathagata Chakraborti, Sarath Sreedharan, and Subbarao Kambhampati. The emerging landscape of explainable automated planning & decision making. In IJCAI, pages 4803–4811, 2020.

[23] Adriana Tapus, Maja J Mataric, and Brian Scassellati. Socially assistive robotics [grand challenges of robotics]. IEEE robotics & automation magazine, 14(1):35–42, 2007.

[24] Aaquib Tabrez and Bradley Hayes. Improving human-robot interaction through explainable reinforcement learning. In 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 751–753. IEEE, 2019.

[25] Aaquib Tabrez, Matthew B Luebbers, and Bradley Hayes. Automated failure-mode clustering and labeling for informed car-to-driver handover in autonomous vehicles. arXiv preprint arXiv:2005.04439, 2020.

[26] Daniel Leyzberg, Samuel Spaulding, and Brian Scassellati. Personalizing robot tutors to individuals' learning differences. In Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction, pages 423–430. ACM, 2014.

[27] Sonia Chernova, Elizabeth Mynatt, Agata Rozga, Reid Simmons, and Holly Yanco. Ai-caring: National ai institute for collaborative assistance and responsive interaction for networked groups. AI Magazine, 45(1):124–130, 2024.

[28] Donald Michie. Machine learning in the next five years. In Proceedings of the 3rd European Conference on European Working Session on Learning, pages 107–122. Pitman Publishing, Inc., 1988.

[29] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision, pages 618–626, 2017.

[30] Brent Mittelstadt, Chris Russell, and Sandra Wachter. Explaining explanations in ai. In Proceedings of the conference on fairness, accountability, and transparency, pages 279–288, 2019.

[31] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence, 267:1–38, 2019.

[32] Q Vera Liao and Jennifer Wortman Vaughan. Ai transparency in the age of llms: A human-centered research roadmap.

[33] Paul Henne, Laura Niemi, Ángel Pinillos, Felipe De Brigard, and Joshua Knobe. A counterfactual explanation for the action effect in causal judgment. Cognition, 190:157–164, 2019.

[34] Bing Cai Kok and Harold Soh. Trust in robots: Challenges and opportunities. Current Robotics Reports, 1:297–309, 2020.

[35] Alan R Wagner, Jason Borenstein, and Ayanna Howard. Overtrust in the robotic age. Communications of the ACM, 61(9):22–24, 2018.

[36] Amos Tversky and Daniel Kahneman. Judgment under uncertainty: Heuristics and biases. science, 185(4157):1124–1131, 1974.

[37] Valerie Chen, Q Vera Liao, Jennifer Wortman Vaughan, and Gagan Bansal. Understanding the role of human intuition on reliance in human-ai decision-making with explanations. Proceedings of the ACM on Human-computer Interaction, 7(CSCW2):1–32, 2023.

[38] John R Wilson and Andrew Rutherford. Mental models: Theory and application in human factors. Human Factors, 31(6):617–634, 1989.

[39] Kenneth James Williams Craik. The nature of explanation, volume 445. CUP Archive, 1952.

[40] John E Mathieu, Tonia S Heffner, Gerald F Goodwin, Eduardo Salas, and Janis A Cannon-Bowers. The influence of shared mental models on team process and performance. Journal of applied psychology, 85(2):273, 2000.

[41] Michelle A Marks, Stephen J Zaccaro, and John E Mathieu. Performance implications of leader briefings and team-interaction training for team adaptation to novel environments. Journal of applied psychology, 85(6):971, 2000.

[42] David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? Behavioral and brain sciences, 1(4):515–526, 1978.

[43] Alison Gopnik, David M Sobel, Laura E Schulz, and Clark Glymour. Causal learning mechanisms in very young children: Two-, three-, and four-year-olds infer causal relations from patterns of variation and covariation. Developmental psychology, 37(5):620, 2001.

[44] Sandra Devin and Rachid Alami. An implemented theory of mind to improve human-robot shared plans execution. In Human-Robot Interaction (HRI), 2016 11th ACM/IEEE International Conference on, pages 319–326. IEEE, 2016.

[45] Yibiao Zhao, Steven Holtzen, Tao Gao, and Song-Chun Zhu. Represent and infer human theory of mind for human-robot interaction. In 2015 AAAI fall symposium series, volume 2, 2015.

[46] O Can Görür, Benjamin S Rosman, Guy Hoffman, and S Albayrak. Toward integrating theory of mind into adaptive decision-making of social robots to understand human intention. 2017.

[47] Brian Scassellati. Theory of mind for a humanoid robot. Autonomous Robots, 12(1):13–24, 2002.

[48] Stefanos Nikolaidis, Yu Xiang Zhu, David Hsu, and Siddhartha Srinivasa. Human-robot mutual adaptation in shared autonomy. In Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, pages 294–302. ACM, 2017.

[49] Connor Brooks and Daniel Szafir. Building second-order mental models for human-robot interaction. arXiv preprint arXiv:1909.06508, 2019.

[50] Cheryl A Bolstad and Mica R Endsley. Shared mental models and shared displays: An empirical evaluation of team performance. In proceedings of the human factors and ergonomics society annual meeting, volume 43, pages 213–217. SAGE Publications Sage CA: Los Angeles, CA, 1999.

[51] Marvin Minsky. A framework for representing knowledge. 1974.

[52] Sharolyn Converse, JA Cannon-Bowers, and E Salas. Shared mental models in expert team decision making. Individual and group decision making: Current issues, 221:221–46, 1993.

[53] Catholijn M. Jonker, M. Birna van Riemsdijk, and Bas Vermeulen. Shared mental models. In Marina De Vos, Nicoletta Fornara, Jeremy V. Pitt, and George Vouros, editors, Coordination, Organizations, Institutions, and Norms in Agent Systems VI, pages 132–151, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[54] Eduardo Salas, Nancy J Cooke, and Michael A Rosen. On teams, teamwork, and team performance: Discoveries and developments. Human factors, 50(3):540–547, 2008.

[55] Guy Hoffman. Evaluating fluency in human–robot collaboration. IEEE Transactions on Human-Machine Systems, 49(3):209–218, 2019.

[56] Stefanos Nikolaidis, Swaprava Nath, Ariel D Procaccia, and Siddhartha Srinivasa. Game-theoretic modeling of human adaptation in human-robot collaboration. In Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction, pages 323–331, 2017.

[57] Stefanie Tellex, Ross Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. Asking for help using inverse semantics. 2014.

[58] Ning Wang, David V Pynadath, and Susan G Hill. Trust calibration within a human-robot team: Comparing automatically generated explanations. In The Eleventh ACM/IEEE International Conference on Human Robot Interaction, pages 109–116. IEEE Press, 2016.

[59] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence, 2018.

[60] Luca Viganò and Daniele Magazzeni. Explainable security. arXiv preprint arXiv:1807.04178, 2018.

[61] Stuart Armstrong and Sören Mindermann. Occam's razor is insufficient to infer the preferences of irrational agents. In Advances in Neural Information Processing Systems, pages 5598–5609, 2018.

[62] Daniel Clement Dennett. The intentional stance. MIT press, 1989.

[63] György Gergely, Zoltán Nádasdy, Gergely Csibra, and Szilvia Bíró. Taking the intentional stance at 12 months of age. Cognition, 56(2):165–193, 1995.

[64] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In Icml, pages 663–670, 2000.

[65] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In Aaai, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[66] Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. Action understanding as inverse planning. Cognition, 113(3):329–349, 2009.

[67] Chris L Baker, Joshua B Tenenbaum, and Rebecca R Saxe. Goal inference as inverse planning. In Proceedings of the Annual Meeting of the Cognitive Science Society, volume 29, 2007.

[68] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. Artificial intelligence, 101(1-2):99–134, 1998.

[69] Chris Baker, Rebecca Saxe, and Joshua Tenenbaum. Bayesian theory of mind: Modeling joint belief-desire attribution. In Proceedings of the annual meeting of the cognitive science society, volume 33, 2011.

[70] Chris L Baker and Joshua B Tenenbaum. Modeling human plan recognition using bayesian theory of mind. Plan, activity, and intent recognition: Theory and practice, pages 177–204, 2014.

[71] Makoto Otsuka and Takayuki Osogami. A deep choice model. In Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[72] Takayuki Osogami and Makoto Otsuka. Restricted boltzmann machines modeling human choice. In Advances in Neural Information Processing Systems, pages 73–81, 2014.

[73] Dorsa Sadigh, Anca Dragan, Shankar Sastry, and Sanjit A Seshia. Active preference-based learning of reward functions. In Robotics: Science and Systems (RSS), 2017.

[74] Stefania Pellegrinelli, Henny Admoni, Shervin Javdani, and Siddhartha Srinivasa. Human-robot shared workspace collaboration via hindsight optimization. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 831–838. IEEE, 2016.

[75] Malayandi Palan, Nicholas C Landolfi, Gleb Shevchuk, and Dorsa Sadigh. Learning reward functions by integrating human demonstrations and preferences. arXiv preprint arXiv:1906.08928, 2019.

[76] Anca D Dragan, Kenton CT Lee, and Siddhartha S Srinivasa. Legibility and predictability of robot motion. In 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 301–308. IEEE, 2013.

[77] Herbert A Simon. Rational decision making in business organizations. The American economic review, 69(4):493–513, 1979.

[78] Minae Kwon, Erdem Biyik, Aditi Talati, Karan Bhasin, Dylan P Losey, and Dorsa Sadigh. When humans aren't optimal: Robots that collaborate with risk-aware humans. In Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction, pages 43–52, 2020.

[79] Piotr J Gmytrasiewicz and Edmund H Durfee. Rational coordination in multi-agent environments. Autonomous Agents and Multi-Agent Systems, 3(4):319–350, 2000.

[80] Yorick Wilks and Afzal Ballim. Multiple agents and the heuristic ascription of belief. Computing Research Laboratory, New Mexico State University, 1986.

[81] Sandy H Huang, David Held, Pieter Abbeel, and Anca D Dragan. Enabling robots to communicate their objectives. Autonomous Robots, 43(2):309–326, 2019.

[82] Piotr J Gmytrasiewicz and Prashant Doshi. A framework for sequential planning in multi-agent settings. Journal of Artificial Intelligence Research, 24:49–79, 2005.

[83] Stefanos Nikolaidis and Julie Shah. Human-robot cross-training: computational formulation, modeling and evaluation of a human team training strategy. In Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction, pages 33–40. IEEE Press, 2013.

[84] Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. In Advances in neural information processing systems, pages 3909–3917, 2016.

[85] Anagha Kulkarni, Yantian Zha, Tathagata Chakraborti, Satya Gautam Vadlamudi, Yu Zhang, and Subbarao Kambhampati. Explicablility as minimizing distance from expected behavior. arXiv preprint arXiv:1611.05497, 2016.

[86] Jin Joo Lee, Fei Sha, and Cynthia Breazeal. A bayesian theory of mind approach to non-verbal communication. In 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 487–496. IEEE, 2019.

[87] Bradley Hayes and Brian Scassellati. Effective robot teammate behaviors for supporting sequential manipulation tasks. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015.

[88] Tathagata Chakraborti, Subbarao Kambhampati, Matthias Scheutz, and Yu Zhang. Ai challenges in human-robot cognitive teaming. arXiv preprint arXiv:1707.04775, 2017.

[89] Anca D Dragan. Robot planning with mathematical models of human state and action. arXiv preprint arXiv:1705.04226, 2017.

[90] Herbert P Grice. Logic and conversation. In Speech acts, pages 41–58. Brill, 1975.

[91] Gordon Briggs and Matthias Scheutz. Facilitating mental modeling in collaborative human-robot interaction through adverbial cues. In Proceedings of the SIGDIAL 2011 Conference, pages 239–247, 2011.

[92] Bradley Hayes and Julie A Shah. Improving robot controller transparency through autonomous policy explanation. In Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction, pages 303–312. ACM, 2017.

[93] Carl Mueller, Aaquib Tabrez, and Bradley Hayes. Interactive constrained learning from demonstration using visual robot behavior counterfactuals.

[94] Andrea Thomaz, Guy Hoffman, and Maya Cakmak. Computational human-robot interaction. Foundations and Trends in Robotics, 4(2-3):105–223, 2016.

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

[96] John D Lee and Katrina A See. Trust in automation: Designing for appropriate reliance. Human factors, 46(1):50–80, 2004.

[97] Peter A Hancock, Deborah R Billings, Kristin E Schaefer, Jessie YC Chen, Ewart J De Visser, and Raja Parasuraman. A meta-analysis of factors affecting trust in human-robot interaction. Human factors, 53(5):517–527, 2011.

[98] Minae Kwon, Malte F Jung, and Ross A Knepper. Human expectations of social robots. In 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 463–464. IEEE, 2016.

[99] Michael Lewis, Katia Sycara, and Phillip Walker. The Role of Trust in Human-Robot Interaction, pages 135–159. Springer International Publishing, Cham, 2018.

[100] Sarah Strohkorb Sebo, Priyanka Krishnamurthi, and Brian Scassellati. "i don't believe you": Investigating the effects of robot trust violation and repair. In 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 57–65. IEEE, 2019.

[101] Zahra Zahedi, Alberto Olmo, Tathagata Chakraborti, Sarath Sreedharan, and Subbarao Kambhampati. Towards understanding user preferences for explanation types in model reconciliation. In 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 648–649. IEEE, 2019.

[102] Stefan-Dan Ciocirlan, Roxana Agrigoroaie, and Adriana Tapus. Human-robot team: Effects of communication in analyzing trust. In 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), pages 1–7. IEEE, 2019.

[103] Tathagata Chakraborti, Sarath Sreedharan, Sachin Grover, and Subbarao Kambhampati. Plan explanations as model reconciliation–an empirical study. arXiv preprint arXiv:1802.01013, 2018.

[104] Minae Kwon, Sandy H Huang, and Anca D Dragan. Expressing robot incapability. In Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, pages 87–95, 2018.

[105] Anagha Kulkarni, Yantian Zha, Tathagata Chakraborti, Satya Gautam Vadlamudi, Yu Zhang, and Subbarao Kambhampati. Explicable planning as minimizing distance from expected behavior. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, pages 2075–2077. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[106] Sebastian Wallkotter, Silvia Tulli, Ginevra Castellano, Ana Paiva, and Mohamed Chetouani. Explainable agents through social cues: A review. arXiv preprint arXiv:2003.05251, 2020.

[107] Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. Metrics for explainable ai: Challenges and prospects. arXiv preprint arXiv:1812.04608, 2018.

[108] Andrew Silva, Mariah Schrum, Erin Hedlund-Botti, Nakul Gopalan, and Matthew Gombolay. Explainable artificial intelligence: Evaluating the objective and subjective impacts of xai on human-agent interaction. International Journal of Human–Computer Interaction, pages 1–15, 2022.

[109] Anca D Dragan and Siddhartha S Srinivasa. Formalizing assistive teleoperation. MIT Press, July, 2012.

[110] Jorge Rios-Martinez, Anne Spalanzani, and Christian Laugier. From proxemics theory to socially-aware navigation: A survey. International Journal of Social Robotics, 7(2):137–153, 2015.

[111] Illah R Nourbakhsh, Katia Sycara, Mary Koes, Mark Yong, Michael Lewis, and Steve Burion. Human-robot teaming for search and rescue. IEEE Pervasive Computing, 4(1):72–79, 2005.

[112] Dorsa Sadigh, Nick Landolfi, Shankar S Sastry, Sanjit A Seshia, and Anca D Dragan. Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state. Autonomous Robots, 42(7):1405–1426, 2018.

[113] Martin Hägele, Klas Nilsson, J Norberto Pires, and Rainer Bischoff. Industrial robotics. In Springer handbook of robotics, pages 1385–1422. Springer, 2016.

[114] Vaibhav V Unhelkar, Przemyslaw A Lasota, Quirin Tyroller, Rares-Darius Buhai, Laurie Marceau, Barbara Deml, and Julie A Shah. Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time. IEEE Robotics and Automation Letters, 3(3):2394–2401, 2018.

[115] Matthew Gombolay, Ronald Wilcox, and Julie Shah. Fast scheduling of multi-robot teams with temporospatial constraints. 2013.

[116] Jimmy Baraglia, Maya Cakmak, Yukie Nagai, Rajesh Rao, and Minoru Asada. Initiative in robot assistance during collaborative task execution. In 2016 11th ACM/IEEE international conference on human-robot interaction (HRI), pages 67–74. IEEE, 2016.

[117] Eric Rosen, David Whitney, Elizabeth Phillips, Gary Chien, James Tompkin, George Konidaris, and Stefanie Tellex. Communicating robot arm motion intent through mixed reality head-mounted displays. In Robotics Research, pages 301–316. Springer, 2020.

[118] Michael Walker, Hooman Hedayati, Jennifer Lee, and Daniel Szafir. Communicating robot motion intent with augmented reality. In Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, pages 316–324, 2018.

[119] Matthew B Luebbers, Connor Brooks, Minjae John Kim, Daniel Szafir, and Bradley Hayes. Augmented reality interface for constrained learning from demonstration. In Proceedings of the 2nd International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI), 2019.

[120] Jason M Gregory, Christopher Reardon, Kevin Lee, Geoffrey White, Ki Ng, and Caitlyn Sims. Enabling intuitive human-robot teaming using augmented reality and gesture control. arXiv preprint arXiv:1909.06415, 2019.

[121] Matthew B Luebbers, Christine T Chang, Aaquib Tabrez, Jordan Dixon, and Bradley Hayes. Emerging autonomy solutions for human and robotic deep space exploration.

[122] Tom Williams, Qin Zhu, Ruchen Wen, and Ewart J de Visser. The confucian matador: Three defenses against the mechanical bull. In Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction, pages 25–33, 2020.

[123] Ryan Blake Jackson and Tom Williams. Language-capable robots may inadvertently weaken human moral norms. In 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 401–410. IEEE, 2019.

[124] Jaime Banks. A perceived moral agency scale: Development and validation of a metric for humans and social machines. Computers in Human Behavior, 90:363–371, 2019.

[125] Matthias Scheutz, Bertram Malle, and Gordon Briggs. Towards morally sensitive action selection for autonomous social robots. In 2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pages 492–497. IEEE, 2015.

[126] Tathagata Chakraborti and Subbarao Kambhampati. Algorithms for the greater good! on mental modeling and acceptable symbiosis in human-ai collaboration. arXiv preprint arXiv:1801.09854, 2018.

[127] Dorsa Sadigh, Shankar Sastry, Sanjit A Seshia, and Anca D Dragan. Planning for autonomous cars that leverage effects on human actions. In Robotics: Science and Systems, volume 2. Ann Arbor, MI, USA, 2016.

[128] Stefanos Nikolaidis and Julie Shah. Human-robot cross-training: computational formulation, modeling and evaluation of a human team training strategy. In Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction, pages 33–40. IEEE Press, 2013.

[129] Bradley Hayes and Brian Scassellati. Challenges in shared-environment human-robot collaboration. In "Collaborative Manipulation" Workshop at the 8th ACM/IEEE International Conference on Human-Robot Interaction., page 8, 2013.

[130] Stefanie Tellex, Ross Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. Asking for help using inverse semantics. 2014.

[131] Jennifer Casper and Robin R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 33(3):367–385, 2003.

[132] Jodi Forlizzi. How robotic products become social products: an ethnographic study of cleaning in the home. In Proceedings of the ACM/IEEE international conference on Human-robot interaction, pages 129–136. ACM, 2007.

[133] Kimitoshi Yamazaki, Ryohei Ueda, Shunichi Nozawa, Mitsuharu Kojima, Kei Okada, Kiyoshi Matsumoto, Masaru Ishikawa, Isao Shimoyama, and Masayuki Inaba. Home-assistant robot for an aging society. Proceedings of the IEEE, 100(8):2429–2441, 2012.

[134] Matthew Spenko, Haoyong Yu, and Steven Dubowsky. Robotic personal aids for mobility and monitoring for the elderly. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 14(3):344–351, 2006.

[135] Marcello Cirillo, Lars Karlsson, and Alessandro Saffiotti. Human-aware task planning for mobile robots. In Advanced Robotics, 2009. ICAR 2009. International Conference on, pages 1–7. IEEE, 2009.

[136] Rachid Alami, Aurélie Clodic, Vincent Montreuil, Emrah Akin Sisbot, and Raja Chatila. Toward human-aware robot task planning. In AAAI spring symposium: to boldly go where no human-robot team has gone before, pages 39–46, 2006.

[137] Julien Guitton, Matthieu Warnier, and Rachid Alami. Belief management for hri planning. BNC@ ECAI 2012, page 27, 2012.

[138] Sid Reddy, Anca Dragan, and Sergey Levine. Where do you think you're going?: Inferring beliefs about dynamics from behavior. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems 31, pages 1461–1472. Curran Associates, Inc., 2018.

[139] Jun-Ichi Imai and Masahide Kaneko. Development of robot which recognizes user's false beliefs using view estimation. In World Automation Congress (WAC), 2010, pages 1–6. IEEE, 2010.

[140] Taylor Kessler Faulkner, Scott Niekum, and Andrea L Thomaz. Robot dialog optimization via modeling of human belief updates. 2017.

[141] Ross A Knepper, Christoforos I Mavrogiannis, Julia Proft, and Claire Liang. Implicit communication in a joint action. In Proceedings of the 2017 acm/ieee international conference on human-robot interaction, pages 283–292. ACM, 2017.

[142] Anca D Dragan, Shira Bauman, Jodi Forlizzi, and Siddhartha S Srinivasa. Effects of robot motion on human-robot collaboration. In Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, pages 51–58. ACM, 2015.

[143] Satoru Satake, Takayuki Kanda, Dylan F Glas, Michita Imai, Hiroshi Ishiguro, and Norihiro Hagita. How to approach humans?: strategies for social robots to initiate interaction. In Proceedings of the 4th ACM/IEEE international conference on Human robot interaction, pages 109–116. ACM, 2009.

[144] Dražen Brščić, Tetsushi Ikeda, and Takayuki Kanda. Do you need help? a robot providing information to people who behave atypically. IEEE Transactions on Robotics, 33(2):500–506, 2017.

[145] Siddhartha Banerjee, Andrew Silva, Karen Feigh, and Sonia Chernova. Effects of interruptibility-aware robot behavior. arXiv preprint arXiv:1804.06383, 2018.

[146] J Gregory Trafton, Allison Jacobs, and Anthony M Harrison. Building and verifying a predictive model of interruption resumption. Proceedings of the IEEE, 100(3):648–659, 2012.

[147] Yi-Shiu Chiang, Ting-Sheng Chu, Chung Dial Lim, Tung-Yen Wu, Shih-Huan Tseng, and Li-Chen Fu. Personalizing robot behavior for interruption in social human-robot interaction. In Advanced Robotics and its Social Impacts (ARSO), 2014 IEEE Workshop on, pages 44–49. IEEE, 2014.

[148] Byung Cheol Lee and Vincent G Duffy. The effects of task interruption on human performance: a study of the systematic classification of human behavior and interruption frequency. Human Factors and Ergonomics in Manufacturing & Service Industries, 25(2):137–152, 2015.

[149] Andrea Bajcsy, Dylan P Losey, Marcia K O'Malley, and Anca D Dragan. Learning from physical human corrections, one feature at a time. In Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, pages 141–149. ACM, 2018.

[150] Ning Wang, David V Pynadath, and Susan G Hill. The impact of pomdp-generated explanations on trust and performance in human-robot teams. In Proceedings of the 2016 international conference on autonomous agents & multiagent systems, pages 997–1005. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

[151] Christoph Bartneck, Dana Kulić, Elizabeth Croft, and Susana Zoghbi. Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. International journal of social robotics, 1(1):71–81, 2009.

[152] Guy Hoffman. Evaluating fluency in human-robot collaboration. In International conference on human-robot interaction (HRI), workshop on human robot collaboration, volume 381, pages 1–8, 2013.

[153] David Abel. A theory of state abstraction for reinforcement learning. In Proceedings of the Doctoral Consortium of the AAAI Conference on Artificial Intelligence, 2019.

[154] LIAO Vera, Yunfeng Zhang, Jorge Andres Moros Ortiz, Amit Dhurandhar, and Ronny Luss. Providing ai explanations based on task context, September 14 2023. US Patent App. 17/654,439.

[155] Tony Belpaeme, James Kennedy, Aditi Ramachandran, Brian Scassellati, and Fumihide Tanaka. Social robots for education: A review. Science robotics, 3(21):eaat5954, 2018.

[156] Iolanda Leite, Carlos Martinho, and Ana Paiva. Social robots for long-term interaction: a survey. International Journal of Social Robotics, 5(2):291–308, 2013.

[157] Christopher J MacLellan, Kenneth R Koedinger, and Noboru Matsuda. Authoring tutors with simstudent: An evaluation of efficiency and model quality. In Intelligent Tutoring Systems: 12th International Conference, ITS 2014, Honolulu, HI, USA, June 5-9, 2014. Proceedings 12, pages 551–560. Springer, 2014.

[158] Tathagata Chakraborti, Sarath Sreedharan, Sachin Grover, and Subbarao Kambhampati. Plan explanations as model reconciliation. In 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 258–266. IEEE, 2019.

[159] Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models–a critical investigation. arXiv preprint arXiv:2305.15771, 2023.

[160] Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. Understanding the capabilities, limitations, and societal impact of large language models. arXiv preprint arXiv:2102.02503, 2021.

[161] Paul Robinette, Wenchen Li, Robert Allen, Ayanna M Howard, and Alan R Wagner. Overtrust of robots in emergency evacuation scenarios. In The Eleventh ACM/IEEE International Conference on Human Robot Interaction, pages 101–108. IEEE Press, 2016.

[162] Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Sam Gershman, and Finale Doshi-Velez. An evaluation of the human-interpretability of explanation. arXiv preprint arXiv:1902.00006, 2019.

[163] Bradley Hayes and Julie Shah. Improving robot controller transparency through autonomous policy explanation. In Proceedings of the 12th ACM/IEEE International Conference on Human-Robot Interaction, 2017.

[164] Daniel Hein, Steffen Udluft, and Thomas A Runkler. Interpretable policies for reinforcement learning by genetic programming. Engineering Applications of Artificial Intelligence, 76:158–169, 2018.

[165] Julian Skirzyński, Frederic Becker, and Falk Lieder. Automatic discovery of interpretable planning strategies. Machine Learning, pages 1–43, 2021.

[166] Andrew Silva, Matthew Gombolay, Taylor Killian, Ivan Jimenez, and Sung-Hyun Son. Optimization methods for interpretable differentiable decision trees applied to reinforcement learning. In International conference on artificial intelligence and statistics, pages 1855–1865. PMLR, 2020.

[167] Rohan Paleja, Letian Chen, Yaru Niu, Andrew Silva, Zhaoxin Li, Songan Zhang, Chace Ritchie, Sugju Choi, Kimberlee Chestnut Chang, Hongtei Eric Tseng, et al. Interpretable reinforcement learning for robotics and continuous control. arXiv preprint arXiv:2311.10041, 2023.

[168] Serena Booth, Yilun Zhou, Ankit Shah, and Julie Shah. Bayes-trex: a bayesian sampling approach to model transparency by example. arXiv preprint arXiv:2002.10248, 2020.

[169] Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review. arXiv preprint arXiv:2010.10596, 2020.

[170] Lakshita Dodeja, Pradyumna Tambwekar, Erin Hedlund-Botti, and Matthew Gombolay. Towards the design of user-centric strategy recommendation systems for collaborative human-ai tasks. International Journal of Human-Computer Studies, page 103216, 2024.

[171] Lindsay Sanneman and Julie A Shah. Validating metrics for reward alignment in human-autonomy teaming. Computers in Human Behavior, 146:107809, 2023.

[172] Kayla Boggess, Sarit Kraus, and Lu Feng. Toward policy explanations for multi-agent reinforcement learning. arXiv preprint arXiv:2204.12568, 2022.

[173] Tobias Kaupp, Alexei Makarenko, and Hugh Durrant-Whyte. Human–robot communication for collaborative decision making—a probabilistic approach. Robotics and Autonomous Systems, 58(5):444–456, 2010.

[174] Lindsay Sanneman, Mycal Tucker, and Julie Shah. An information bottleneck characterization of the understanding-workload tradeoff. arXiv preprint arXiv:2310.07802, 2023.

[175] Matthew B Luebbers, Aaquib Tabrez, Kyler Ruvane, and Bradley Hayes. Autonomous justification for enabling explainable decision support in human-robot teaming. 2023.

[176] David Andre and Stuart J Russell. State abstraction for programmable reinforcement learning agents. In AAAI/IAAI, pages 119–125, 2002.

[177] David Abel, Dilip Arumugam, Lucas Lehnert, and Michael Littman. State abstractions for lifelong reinforcement learning. In International Conference on Machine Learning, pages 10–19, 2018.

[178] Richard E Fikes and Nils J Nilsson. Strips: A new approach to the application of theorem proving to problem solving. Artificial intelligence, 2(3-4):189–208, 1971.

[179] Richard Bellman. A markovian decision process. Journal of Mathematics and Mechanics, pages 679–684, 1957.

[180] Bernhard H Korte, Jens Vygen, B Korte, and J Vygen. Combinatorial optimization, volume 1. Springer, 2011.

[181] "Gurobi Optimization LLC". Gurobi optimizer, 2019.

[182] Edward J McCluskey. Minimization of boolean functions. The Bell System Technical Journal, 35(6):1417–1444, 1956.

[183] Lindsay Sanneman and Julie A Shah. An empirical study of reward explanations with human-robot interaction applications. IEEE Robotics and Automation Letters, 7(4):8956–8963, 2022.

[184] Francisco Cruz, Charlotte Young, Richard Dazeley, and Peter Vamplew. Evaluating human-like explanations for robot actions in reinforcement learning scenarios. In 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 894–901. IEEE, 2022.

[185] OpenAI. Gpt-4 technical report, 2023.

[186] Upol Ehsan, Pradyumna Tambwekar, Larry Chan, Brent Harrison, and Mark O Riedl. Automated rationale generation: a technique for explainable ai and its effects on human perceptions. In Proceedings of the 24th International Conference on Intelligent User Interfaces, pages 263–274, 2019.

[187] Donghee Shin. The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable ai. International Journal of Human-Computer Studies, 146:102551, 2021.

[188] Jiun-Yin Jian, Ann M Bisantz, and Colin G Drury. Foundations for an empirically determined scale of trust in automated systems. International journal of cognitive ergonomics, 4(1):53–71, 2000.

[189] Kristin E Schaefer. Measuring trust in human robot interactions: Development of the "trust perception scale-hri". In Robust intelligence and trust in autonomous systems, pages 191–218. Springer, 2016.

[190] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In Advances in psychology, volume 52, pages 139–183. Elsevier, 1988.

[191] Christine T Chang, Matthew B Luebbers, Mitchell Hebert, and Bradley Hayes. Human non-compliance with robot spatial ownership communicated via augmented reality: Implications for human-robot teaming safety. In 2023 IEEE International Conference on Robotics and Automation (ICRA), pages 9785–9792. IEEE, 2023.

[192] Ewart J De Visser, Marieke MM Peeters, Malte F Jung, Spencer Kohn, Tyler H Shaw, Richard Pak, and Mark A Neerincx. Towards a theory of longitudinal trust calibration in human–robot teams. International journal of social robotics, 12(2):459–478, 2020.

[193] Aaquib Tabrez, Matthew B Luebbers, and Bradley Hayes. Descriptive and prescriptive visual guidance to improve shared situational awareness in human-robot teaming. In Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems, pages 1256–1264, 2022.

[194] Rohan Paleja, Andrew Silva, Letian Chen, and Matthew Gombolay. Interpretable and personalized apprenticeship scheduling: Learning interpretable scheduling policies from heterogeneous user demonstrations. Advances in neural information processing systems, 33:6417–6428, 2020.

[195] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognition, 47(6):2280–2292, 2014.

[196] Hong Jun Jeon, Smitha Milli, and Anca D Dragan. Reward-rational (implicit) choice: A unifying formalism for reward learning. arXiv preprint arXiv:2002.04833, 2020.

[197] Bradley Hayes and Brian Scassellati. Challenges in shared-environment human-robot collaboration. In "Collaborative Manipulation" Workshop at the 8th ACM/IEEE International Conference on Human-Robot Interaction., page 8, 2013.

[198] Adriana Tapus, Mataric Maja, and Brian Scassellatti. The grand challenges in socially assistive robotics. IEEE Robotics and Automation Magazine, 14(1):N–A, 2007.

[199] Joost Broekens, Marcel Heerink, Henk Rosendal, et al. Assistive social robots in elderly care: a review. Gerontechnology, 8(2):94–103, 2009.

[200] Sebastian Thrun and Lorien Pratt. Learning to learn. Springer Science & Business Media, 2012.

[201] Carl Mueller, Jeff Venicx, and Bradley Hayes. Robust robot learning from demonstration and skill repair using conceptual constraints. 2018.

[202] Aaron St. Clair and Maja Mataric. How robot verbal feedback can improve team performance in human-robot task collaborations. In Proceedings of the tenth annual acm/ieee international conference on human-robot interaction, pages 213–220, 2015.

[203] Dorsa Sadigh, Anca D Dragan, Shankar Sastry, and Sanjit A Seshia. Active preference-based learning of reward functions. In Robotics: Science and Systems, 2017.

[204] Çetin Meriçli, Manuela Veloso, and H Levent Akın. Task refinement for autonomous robots using complementary corrective human feedback. International Journal of Advanced Robotic Systems, 8(2):16, 2011.

[205] Matthias Scheutz, Evan Krause, Brad Oosterveld, Tyler Frasca, and Robert Platt. Spoken instruction-based one-shot object and action learning in a cognitive robotic architecture. In Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, 2017.

[206] Tom Williams, Gordon Briggs, Bradley Oosterveld, and Matthias Scheutz. Going beyond literal command-based instructions: Extending robotic natural language interaction capabilities. In Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

[207] Daniel H Grollman and Aude Billard. Donut as i do: Learning from failed demonstrations. In In International Conference on Robotics and Automation, 2011.

[208] Daniel H Grollman and Aude G Billard. Robot learning from failed demonstrations. International Journal of Social Robotics, 2012.

[209] Sebastian Wrede, Christian Emmerich, Ricarda Grünberg, Arne Nordmann, Agnes Swadzba, and Jochen Steil. A user study on kinesthetic teaching of redundant robots in task and configuration space. Journal of Human-Robot Interaction.

[210] Terrence Fong, Charles Thorpe, and Charles Baur. Collaborative control: A robot-centric model for vehicle teleoperation, volume 1. Carnegie Mellon University, The Robotics Institute Pittsburgh, 2001.

[211] Erick Romero Kramer, Argentina Ortega Sáinz, Alex Mitrevski, and Paul G Plöger. Tell your robot what to do: Evaluation of natural language models for robot command processing. In Robot World Cup. Springer, 2019.

[212] Matthew B Luebbers, Connor Brooks, Carl L Mueller, Daniel Szafir, and Bradley Hayes. Arc-lfd: Using augmented reality for interactive long-term robot skill maintenance via constrained learning from demonstration.

[213] Nichola Abdo, Cyrill Stachniss, Luciano Spinello, and Wolfram Burgard. Robot, organize my shelves! tidying up objects by predicting user preferences. In 2015 IEEE international conference on robotics and automation (ICRA).

[214] Andreea Bobu, Andrea Bajcsy, Jaime F Fisac, and Anca D Dragan. Learning under misspecified objective spaces. arXiv preprint arXiv:1810.05157, 2018.

[215] Nils Wilde, Dana Kulić, and Stephen L Smith. Learning user preferences in robot motion planning through interaction. In 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018.

[216] Maya Cakmak and Andrea L Thomaz. Designing robot learners that ask good questions. In 2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 17–24. IEEE, 2012.

[217] Erdem Bıyık, Malayandi Palan, Nicholas C Landolfi, Dylan P Losey, and Dorsa Sadigh. Asking easy questions: A user-friendly approach to active reward learning. arXiv preprint arXiv:1910.04365, 2019.

[218] Ivan Volosyak, Oleg Ivlev, and Axel Graser. Rehabilitation robot friend ii-the general concept and current implementation. In 9th International Conference on Rehabilitation Robotics. IEEE, 2005.

[219] Léonard Jaillet and Josep M Porta. Path planning under kinematic constraints by rapidly exploring manifolds. IEEE Transactions on Robotics, 2012.

[220] Edward Loper Bird, Steven and Ewan Klein. Natural Language Processing with Python. O'Reilly Media Inc., 2009.

[221] George A. Miller. Wordnet: A lexical database for english. 1995.

[222] Bruce H Deatherage. Auditory and other sensory forms of information presentation. Human engineering guide to equipment design, pages 123–160, 1972.

[223] Haikun Huang, Ni-Ching Lin, Lorenzo Barrett, Darian Springer, Hsueh-Cheng Wang, Marc Pomplun, and Lap-Fai Yu. Automatic optimization of wayfinding design. IEEE transactions on visualization and computer graphics, 24(9):2516–2530, 2017.

[224] THJ Collett and Bruce A MacDonald. Developer oriented visualisation of a robot program. In Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction, pages 49–56, 2006.

[225] Tijn Kooijmans, Takayuki Kanda, Christoph Bartneck, Hiroshi Ishiguro, and Norihiro Hagita. Interaction debugging: an integral approach to analyze human-robot interaction. In Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction, pages 64–71, 2006.

[226] Daniel Szafir and Danielle Albers Szafir. Connecting human-robot interaction and data visualization. In Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction, pages 281–292, 2021.

[227] Jack Gale, John Karasinski, and Steve Hillenius. Playbook for uas: Ux of goal-oriented planning and execution. In International Conference on Engineering Psychology and Cognitive Ergonomics, pages 545–557. Springer, 2018.

[228] Nisar Ahmed, Mark Campbell, David Casbeer, Yongcan Cao, and Derek Kingston. Fully bayesian learning and spatial reasoning with flexible human sensor networks. In Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems, pages 80–89, 2015.

[229] Umang Bhatt, Javier Antorán, Yunfeng Zhang, Q Vera Liao, Prasanna Sattigeri, Riccardo Fogliato, Gabrielle Melançon, Ranganath Krishnan, Jason Stanley, Omesh Tickoo, et al. Uncertainty as a form of transparency: Measuring, communicating, and using uncertainty. In Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, pages 401–413, 2021.

[230] Mark Colley, Benjamin Eder, Jan Ole Rixen, and Enrico Rukzio. Effects of semantic segmentation visualization on trust, situation awareness, and cognitive load in highly automated vehicles. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pages 1–11, 2021.

[231] Christopher Reardon, Kevin Lee, John G Rogers, and Jonathan Fink. Communicating via augmented reality for human-robot teaming in field environments. In 2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), pages 94–101. IEEE, 2019.

[232] Savannah Paul, Christopher Reardon, Tom Williams, and Hao Zhang. Designing augmented reality visualizations for synchronized and time-dominant human-robot teaming. In Virtual, Augmented, and Mixed Reality (XR) Technology for Multi-Domain Operations, volume 11426, page 1142607. International Society for Optics and Photonics, 2020.

[233] Marlena R Fraune, Ahmed S Khalaf, Mahlet Zemedie, Poom Pianpak, Zahra NaminiMianji, Sultan A Alharthi, Igor Dolgov, Bill Hamilton, Son Tran, and ZO Toups. Developing future wearable interfaces for human-drone teams through a virtual drone search game. International Journal of Human-Computer Studies, 147:102573, 2021.

[234] Alexander Kunze, Stephen J Summerskill, Russell Marshall, and Ashleigh J Filtness. Augmented reality displays for communicating uncertainty information in automated driving. In Proceedings of the 10th international conference on automotive user interfaces and interactive vehicular applications, pages 164–175, 2018.

[235] Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. Explainable AI: interpreting, explaining and visualizing deep learning, volume 11700. Springer Nature, 2019.

[236] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. Advances in neural information processing systems, 30, 2017.

[237] Peter Lipton. Contrastive explanation. Royal Institute of Philosophy Supplements, 27:247–266, 1990.

[238] Tim Miller. Contrastive explanation: A structural-model approach. arXiv preprint arXiv:1811.03163, 2018.

[239] John R Frost. The theory of search: a simplified explanation. Soza Limited, 1997.

[240] Michał Wysokiński, Robert Marcjan, and Jacek Dajda. Decision support software for search & rescue operations. Procedia Computer Science, 35:776–785, 2014.

[241] Lu Yadong and Zhou Ya. Optimal search and rescue model: Updating probability density map of debris location by bayesian method. International Journal of Statistical Distributions and Applications, 1(1):12, 2015.

[242] Renan Luigi Martins Guarese and Anderson Maciel. Development and usability analysis of a mixed reality gps navigation application for the microsoft hololens. In Computer Graphics International Conference, pages 431–437. Springer, 2019.

[243] Danyel Fisher. Hotmap: Looking at geographic attention. IEEE transactions on visualization and computer graphics, 13(6):1184–1191, 2007.

[244] James V Bradley. Complete counterbalancing of immediate sequential effects in a latin square design. Journal of the American Statistical Association, 53(282):525–528, 1958.

[245] David Roxbee Cox and Nancy Reid. The theory of the design of experiments. CRC Press, 2000.

[246] Michael Lewis, Katia Sycara, and Phillip Walker. The role of trust in human-robot interaction. In Foundations of trusted autonomy, pages 135–159. Springer, Cham, 2018.

[247] Aaquib Tabrez, Matthew B Luebbers, Kyler Ruvane, Ashley H Rabin, Kevin W King, William Gerichs, and Bradley Hayes. Hierarchical multi-agent reinforcement learning with explainable decision support for human-robot teams. 2024.

[248] András Gulyás, József Bíró, Gábor Rétvári, Márton Novák, Attila Kőrösi, Mariann Slíz, and Zalán Heszberger. The role of detours in individual human navigation patterns of complex networks. Scientific Reports, 10(1):1098, 2020.

[249] Salvador Aguinaga, Aditya Nambiar, Zuozhu Liu, and Tim Weninger. Concept hierarchies and human navigation. In 2015 IEEE International Conference on Big Data (Big Data), pages 38–45. IEEE, 2015.

[250] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal on scientific Computing, 20(1):359–392, 1998.

[251] Yair Censor. Pareto optimality in multiobjective problems. Applied Mathematics and Optimization, 4(1):41–59, 1977.

[252] Matthew B Luebbers, Aaquib Tabrez, Kyler Ruvane, and Bradley Hayes. Autonomous Justification for Enabling Explainable Decision Support in Human-Robot Teaming. In Proceedings of Robotics: Science and Systems, Daegu, Republic of Korea, July 2023.

[253] Daniel Kahneman. Thinking, fast and slow. macmillan, 2011.

[254] Filipa Correia, Carla Guerra, Samuel Mascarenhas, Francisco S Melo, and Ana Paiva. Exploring the impact of fault justification in human-robot trust. In Proceedings of the 17th international conference on autonomous agents and multiagent systems, pages 507–513, 2018.

[255] Erik Cambria, Lorenzo Malandri, Fabio Mercorio, Mario Mezzanzanica, and Navid Nobani. A survey on xai and natural language explanations. Information Processing & Management, 60(1):103111, 2023.

[256] Finale Doshi-Velez, Mason Kortz, Ryan Budish, Chris Bavitz, Sam Gershman, David O'Brien, Kate Scott, Stuart Schieber, James Waldo, David Weinberger, et al. Accountability of ai under the law: The role of explanation. arXiv preprint arXiv:1711.01134, 2017.

[257] Ruikun Luo, Na Du, and X Jessie Yang. Evaluating effects of enhanced autonomy transparency on trust, dependence, and human-autonomy team performance over time. International Journal of Human–Computer Interaction, 38(18-20):1962–1971, 2022.

[258] Guy Hoffman, Maya Cakmak, and Crystal Chao. Timing in human-robot interaction. In Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction, pages 509–510, 2014.

[259] Matthew B Luebbers, Aaquib Tabrez, and Bradley Hayes. Augmented reality-based explainable ai strategies for establishing appropriate reliance and trust in human-robot teaming. In 5th International Workshop on Virtual, Augmented, and Mixed Reality for HRI.

[260] Thomas B Sheridan. Human–robot interaction: status and challenges. Human factors, 58(4):525–532, 2016.

[261] Sarath Sreedharan, Tathagata Chakraborti, and Subbarao Kambhampati. Balancing explicability and explanation in human-aware planning. In 2017 AAAI Fall Symposium, pages 61–68. AI Access Foundation, 2017.

[262] Raymond Sheh. Explainable artificial intelligence requirements for safe, intelligent robots. In 2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR), pages 382–387. IEEE, 2021.

[263] X Jessie Yang, Vaibhav V Unhelkar, Kevin Li, and Julie A Shah. Evaluating effects of user experience and system transparency on trust in automation. In Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction, pages 408–416, 2017.

[264] Nicholas Conlon, Daniel Szafir, and Nisar Ahmed. Investigating the effects of robot proficiency self-assessment on trust and performance. arXiv preprint arXiv:2203.10407, 2022.

[265] Sarath Sreedharan, Anagha Kulkarni, and Subbarao Kambhampati. Balancing communication and behavior. In Explainable Human-AI Interaction: A Planning Perspective, pages 95–105. Springer, 2022.

[266] Tathagata Chakraborti, Anagha Kulkarni, Sarath Sreedharan, David E Smith, and Subbarao Kambhampati. Explicability? legibility? predictability? transparency? privacy? security? the emerging landscape of interpretable agent behavior. In Proceedings of the international conference on automated planning and scheduling, volume 29, pages 86–96, 2019.

[267] Sule Anjomshoae, Amro Najjar, Davide Calvaresi, and Kary Främling. Explainable agents and robots: Results from a systematic literature review. In 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019, pages 1078–1088. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[268] Avi Rosenfeld and Ariella Richardson. Explainability in human–agent systems. Autonomous Agents and Multi-Agent Systems, 33(6):673–705, 2019.

[269] Richard F Thompson and William A Spencer. Habituation: a model phenomenon for the study of neuronal substrates of behavior. Psychological review, 73(1):16, 1966.

[270] Kalanit Grill-Spector, Richard Henson, and Alex Martin. Repetition and the brain: neural models of stimulus-specific effects. Trends in cognitive sciences, 10(1):14–23, 2006.

[271] Matthew Grizzard, Ron Tamborini, John L Sherry, René Weber, Sujay Prabhu, Lindsay Hahn, and Patrick Idzik. The thrill is gone, but you might not know: Habituation and generalization of biophysiological and self-reported arousal responses to video games. Communication Monographs, 82(1):64–87, 2015.

[272] Jeffrey L Jenkins, Bonnie Brinton Anderson, Anthony Vance, C Brock Kirwan, and David Eargle. More harm than good? how messages that interrupt can make us vulnerable. Information Systems Research, 27(4):880–896, 2016.

[273] Ronald A Howard. Information value theory. IEEE Transactions on systems science and cybernetics, 2(1):22–26, 1966.

[274] Walter Mischel. The marshmallow test: Why self-control is the engine of success. Little, Brown New York, 2015.

[275] Dan Amir and Ofra Amir. Highlights: Summarizing agent behavior to people. In Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, pages 1168–1176, 2018.

[276] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. Harv. JL & Tech., 31:841, 2017.

[277] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In International Conference on Machine Learning, pages 2376–2384. PMLR, 2019.

[278] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10800–10809, 2020.

[279] Juntao Tan, Shuyuan Xu, Yingqiang Ge, Yunqi Li, Xu Chen, and Yongfeng Zhang. Counterfactual explainable recommendation. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pages 1784–1793, 2021.

[280] Derek Doran, Sarah Schulz, and Tarek R Besold. What does explainable ai really mean? a new conceptualization of perspectives. arXiv preprint arXiv:1710.00794, 2017.

[281] Herbert A Simon. Bounded rationality. Utility and probability, pages 15–18, 1990.

[282] Scott Plous. The psychology of judgment and decision making. Mcgraw-Hill Book Company, 1993.

[283] Christoph Bartneck, Dana Kulić, Elizabeth Croft, and Susana Zoghbi. Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. International journal of social robotics, 1:71–81, 2009.

[284] Guy Hoffman and Xuan Zhao. A primer for conducting experiments in human–robot interaction. ACM Transactions on Human-Robot Interaction (THRI), 10(1):1–31, 2020.

[285] Matthew Gombolay, Xi Jessie Yang, Bradley Hayes, Nicole Seo, Zixi Liu, Samir Wadhwania, Tania Yu, Neel Shah, Toni Golen, and Julie Shah. Robotic assistance in the coordination of patient care. The International Journal of Robotics Research, 37(10):1300–1316, 2018.

[286] Stephen R Dixon and Christopher D Wickens. Automation reliability in unmanned aerial vehicle control: A reliance-compliance model of automation dependence in high workload. Human factors, 48(3):474–486, 2006.

[287] Abhinav Dahiya, Alexander M Aroyo, Kerstin Dautenhahn, and Stephen L Smith. A survey of multi-agent human–robot interaction systems. Robotics and Autonomous Systems, 161:104335, 2023.

[288] Shih-Yun Lo, Elaine Schaertl Short, and Andrea L Thomaz. Planning with partner uncertainty modeling for efficient information revealing in teamwork. In Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction, pages 319–327, 2020.

[289] Aaquib Tabrez, Ryan Leonard, and Bradley Hayes. One-shot policy elicitation via semantic reward manipulation. arXiv preprint arXiv:2101.01860, 2022.

[290] Emily Jensen, Sriram Sankaranarayanan, and Bradley Hayes. Automated assessment and adaptive multimodal formative feedback improves psychomotor skills training outcomes in quadrotor teleoperation. arXiv preprint arXiv:2405.15982, 2024.

[291] Vinitha Ranganeni and Maya Cakmak. Accessible tele-operation interfaces for assistive robots. In Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, pages 91–93, 2024.

[292] Carlos Quintero-Pena, Peizhu Qian, Nicole M Fontenot, Hsin-Mei Chen, Shannan K Hamlin, Lydia E Kavraki, and Vaibhav Unhelkar. Robotic tutors for nurse training: Opportunities for hri researchers. In 2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), pages 220–225. IEEE, 2023.

[293] Yao Rong, Peizhu Qian, Vaibhav Unhelkar, and Enkelejda Kasneci. I-cee: Tailoring explanations of image classification models to user expertise. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, pages 21545–21553, 2024.

[294] Peizhu Qian and Vaibhav Unhelkar. Evaluating the role of interactivity on improving transparency in autonomous agents. In Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems, pages 1083–1091, 2022.

[295] Cunjun Yu, Yiqing Xu, Linfeng Li, and David Hsu. Coach: Cooperative robot teaching. In Conference on Robot Learning, pages 1092–1103. PMLR, 2023.

[296] KwangYong Lee and Gerard Jounghyun Kim. Tele-augmentation for remote ar coaching. In Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology, pages 197–197, 2015.

[297] Varad Dhat, Nick Walker, and Maya Cakmak. Using 3d mice to control robot manipulators. In Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, pages 896–900, 2024.

[298] Omayma Mahjoub, Ruan de Kock, Siddarth Singh, Wiem Khlifi, Abidine Vall, Kale-ab Tessera, and Arnu Pretorius. Efficiently quantifying individual agent importance in cooperative marl. arXiv preprint arXiv:2312.08466, 2023.

[299] Jake Brawer, Kayleigh Bishop, Bradley Hayes, and Alessandro Roncone. Towards a natural language interface for flexible multi-agent task assignment. In Proceedings of the AAAI Symposium Series, volume 2, pages 167–171, 2023.

[300] Peizhu Qian, Harrison Huang, and Vaibhav Unhelkar. Pps: Personalized policy summarization for explaining sequential behavior of autonomous agents. In Proceedings of the 2024 AAAI/ACM Conference on AI, Ethics, and Society, 2024.

[301] Devleena Das, Siddhartha Banerjee, and Sonia Chernova. Explainable ai for robot failures: Generating explanations that improve user assistance in fault recovery. In Proceedings of the 2021 ACM/IEEE international conference on human-robot interaction, pages 351–360, 2021.

[302] Yotam Amitai, Guy Avni, and Ofra Amir. Asq-it: Interactive explanations for reinforcement-learning agents. arXiv preprint arXiv:2301.09941, 2023.

[303] Zahra Zahedi, Sailik Sengupta, and Subbarao Kambhampati. 'why didn't you allocate this task to them?'negotiation-aware task allocation and contrastive explanation generation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, pages 10243–10251, 2024.

[304] Zibin Dong, Yifu Yuan, Jianye Hao, Fei Ni, Yao Mu, Yan Zheng, Yujing Hu, Tangjie Lv, Changjie Fan, and Zhipeng Hu. Aligndiff: Aligning diverse human preferences via behavior-customisable diffusion model. arXiv preprint arXiv:2310.02054, 2023.

[305] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large language models meet personalization. arXiv preprint arXiv:2304.11406, 2023.

[306] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. July 2023.

[307] Andrew Szot, Max Schwarzer, Harsh Agrawal, Bogdan Mazoure, Walter Talbott, Katherine Metcalf, Natalie Mackraz, Devon Hjelm, and Alexander Toshev. Large Language Models as Generalizable Policies for Embodied Tasks, October 2023. arXiv:2310.17722 [cs].

[308] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. SayPlan: Grounding Large Language Models using 3D Scene Graphs for Scalable Robot Task Planning. August 2023.

[309] Atharva Gundawar, Mudit Verma, Lin Guan, Karthik Valmeekam, Siddhant Bhambri, and Subbarao Kambhampati. Robust planning with llm-modulo framework: Case study in travel planning. arXiv preprint arXiv:2405.20625, 2024.

[310] Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Kaya Stechly, Mudit Verma, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. Llms can't plan, but can help planning in llm-modulo frameworks. arXiv preprint arXiv:2402.01817, 2024.

[311] Huao Li, Yu Quan Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Michael Lewis, and Katia Sycara. Theory of mind for multi-agent collaboration via large language models. arXiv preprint arXiv:2310.10701, 2023.

[312] Jiannan Xiang, Tianhua Tao, Yi Gu, Tianmin Shu, Zirui Wang, Zichao Yang, and Zhiting Hu. Language models meet world models: Embodied experiences enhance language models. Advances in neural information processing systems, 36, 2024.