

**Spatially-Grounded Communication for Mental Model
Alignment in Human-Robot Teams**

by

Matthew B. Luebbers

B.A., Cornell University, 2018

M.S., University of Colorado Boulder, 2021

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

Daniel Szafir

Nisar Ahmed

Michael Gleicher

Luebbers, Matthew B. (Ph.D., Computer Science)

Spatially-Grounded Communication for Mental Model Alignment in Human-Robot Teams

Thesis directed by Prof. Bradley Hayes

There is great potential for humans and autonomous robots, each possessing their own capabilities and strengths, to perform tasks collaboratively across a number of domains, achieving greater performance than either could on their own. As is true for human-human teams, however, human-robot teams require a great deal of coordination. In shared tasks complex enough to see emergent benefits from teamwork, high-performing teams tend to possess well-aligned mental models regarding the task and each member's role within it, quickly communicating to rectify those models during times of mismatched expectation. To achieve the same benefits in human-robot teams requires a similar fluency of communication. However, since robots and humans reason in vastly different planning spaces, communicating effectively is non-trivial. Robot plans and rationale are often derived from mathematical optimization, which is difficult for human teammates to understand. Likewise, human decision-making patterns are difficult to quantify and are subject to significant noise, hindering their usefulness for optimization-based planners. Team fluency can be greatly improved by bridging human and robot task representations within the context of communication. In this thesis, I will discuss my research developing novel systems, algorithms, and interfaces for explicitly synchronizing mental models via agent-to-agent communication during live human-robot collaboration, spanning tasks ranging from tabletop manipulation to environment navigation and search. In particular, I will focus on spatially-grounded communication methods (augmented reality-based visualization presented in-situ at key locations within a shared environment, and natural language communication tied to such spatially-grounded features). Such methods leverage shared context between human and robot teammates, allowing for compact bi-directional communication of environment and task information, thus facilitating the alignment of mental models between agents and improving objective and subjective measures of team performance.

Acknowledgements

First, I would like to extend my sincere thanks to my steadfast advisor for the past six years: Brad Hayes. I couldn't have asked for a better mentor. The effect his guidance and example has had on my growth as a researcher, and as a science communicator, is hard to overstate. I have benefited greatly from his constant support and optimism, and his uncanny ability make any challenge seem surmountable. Next, I'd like to thank my wonderful labmates, past and present, from the Collaborative AI and Robotics Lab, as well as the broader Colorado computer science and robotics community. I'd especially like to acknowledge my peer co-authors, each of whom worked side by side with me in the lab to produce some portion or other of the research contained in this thesis: Aaquib Tabrez, Connor Brooks, Carl Mueller, Yi-Shiuan Tung, Christine Chang, and Kyler Ruvane. I learned something new from collaborating with each of them. I would also like to thank all of my co-workers and co-interns from my multiple summers spent at NASA's Jet Propulsion Laboratory, especially those who served as my internship mentors: Galen Hollins, Jeng Yen, Trevor Reed, Kris Wehage, and Jackie Ryan. Those internships gave me opportunities I never would have dreamed of: getting to work on, and even help operate, the Mars rovers Curiosity and Perseverance. Those experiences serve as a constant reminder of why I chose to research robotics in the first place. Lastly, a big thank you to my friends and family, who supported and encouraged me throughout my research journey, and whose contributions to this thesis have not gone unnoticed.

Contents

Chapter	
1	1
1.1	1
1.2	4
1.3	4
1.4	5
1.4.1	5
1.4.2	6
1.4.3	7
1.4.4	9
1.4.5	13
1.4.6	15
2	18
2.1	18
2.2	19
2.2.1	19
2.2.2	21

2.2.3	Concept Constrained Learning from Demonstration	25
2.2.4	Augmented Reality System Design	26
2.2.5	System Validation	32
2.2.6	Conclusion	35
3	Augmented Reality for Reducing Uncertainty in Human Motion in Shared-Space Interaction	37
3.1	Motivation	37
3.2	Workspace Optimization Techniques to Improve Prediction of Human Motion During Human-Robot Collaboration	39
3.2.1	Introduction	39
3.2.2	Related Work	41
3.2.3	Legible Workspace Generation	43
3.2.4	Evaluation	47
3.2.5	Results	52
3.2.6	Conclusion	54
3.3	Human Non-Compliance with Robot Spatial Ownership Communicated via Aug- mented Reality: Implications for Human-Robot Teaming Safety	55
3.3.1	Introduction	55
3.3.2	Related Work	57
3.3.3	The FENCES System	59
3.3.4	Experiment Design	60
3.3.5	Results	63
3.3.6	Discussion and Findings	69
3.3.7	Design Recommendations	71
4	Explainable Visual Guidance for Human-Robot Search Tasks	72
4.1	Motivation	72

4.2	Descriptive and Prescriptive Visual Guidance to Improve Shared Situational Awareness in Human-Robot Teaming	74
4.2.1	Introduction	74
4.2.2	Background and Related Work	76
4.2.3	Algorithmic Approach	78
4.2.4	AR-based Visual Guidance Design	83
4.2.5	Experimental Validation	85
4.2.6	Results and Discussion	90
4.3	Hierarchical Multi-Agent Reinforcement Learning with Explainable Decision Support for Human-Robot Teams	96
4.3.1	Practical Limitations of the MARS Framework	96
4.3.2	Hierarchical Min-Entropy Algorithm for Robot-Supplied Suggestions	99
4.3.3	Hierarchical Environment Creation	101
4.3.4	Hierarchical Guidance Design	104
4.3.5	Simulation Testbed	106
4.3.6	Algorithmic Performance Evaluation	109
4.3.7	Guidance-Type Study Design	115
4.3.8	Conclusion	117
5	Spatially-Grounded Justifications for Robotic Decision Support	119
5.1	Motivation	119
5.2	Autonomous Justification for Enabling Explainable Decision Support in Human-Robot Teaming	120
5.2.1	Introduction	120
5.2.2	Background & Related Work	123
5.2.3	Definition of Application Domain	125
5.2.4	Justification Framework: Timing	127

5.2.5	Justification Framework: Content	133
5.2.6	Experimental Evaluation	138
5.2.7	Results	142
5.2.8	Recommendations & Potential Applications	148
5.2.9	Conclusion	153
6	Conclusion and Future Directions	155
6.1	Summary of Individual Contributions	155
6.2	Directions for Future Work	156
6.2.1	Cooperative Reward Design	156
6.2.2	Remedial Actions for Low Robot Confidence	158
6.2.3	Leveraging Psychology for Improved Human Modeling	159
	Bibliography	161

Tables

Table

2.1	ARC-LfD Constraint Parametrization	29
3.1	Multivariate Gaussian Model Determinants and Traces	53
3.2	FENCES Survey Responses	66
3.3	FENCES Mean Responses by Condition	66
4.1	H-MARS 1 Human 1 Robot Simulation Table	113
4.2	H-MARS 1 Human 5 Robots Simulation Table	115
5.1	Justification Timing Study Results	133
5.2	Justification Study Subjective Scales	141
5.3	Per-Condition Justification Content Study Results: Objective	142
5.4	Bucketed Justification Content Study Results: Objective	143
5.5	Per-Condition Justification Content Study Results: Subjective	146
5.6	Bucketed Justification Content Study Results: Subjective	147

Figures

Figure

2.1	ARC-LfD Side by Side	21
2.2	ARC-LfD System Architecture	27
2.3	ARC-LfD Interaction Flowchart	28
2.4	ARC-LfD Visualizations	29
2.5	Menu Interaction in ARC-LfD	31
2.6	ARC-LfD Case Study I	33
2.7	ARC-LfD Case Study II	34
2.8	ARC-LfD Case Study III	34
3.1	Workspace Configuration Side by Side	40
3.2	Workspace Configuration System Diagram	40
3.3	Tabletop Experiment Configurations	48
3.4	Legibility Search Space	49
3.5	Tabletop Experiment Accuracy	51
3.6	Tabletop Time-Series Multivariate Gaussians	54
3.7	FENCES AR Interface	56
3.8	FENCES Block Assembly Task	62
3.9	FENCES Mean Distance Plot	66
3.10	FENCES Perceived Intelligence vs. Head Check Plot	67

4.1	MARS Prescriptive & Descriptive Guidance	74
4.2	MARS Algorithm Flow	80
4.3	MARS Experimental Conditions	87
4.4	MARS Experiment Performance & Compliance Results	93
4.5	2D Environment Graph Partition	100
4.6	MARS-style and Regional Descriptive Guidance	105
4.7	H-MARS Descriptive Guidance	106
4.8	H-MARS Testbed Inter-Regional Guidance	107
4.9	H-MARS Testbed Intra-Regional Guidance	108
4.10	H-MARS Algorithmic Evaluation Environment	110
4.11	H-MARS 1 Human 1 Robot Results	112
4.12	H-MARS 1 Human 5 Robots Results	114
5.1	Autonomous Justification Examples	122
5.2	MARS Interaction Loop with Justification Module	126
5.3	Justification Types	135
5.4	Drone Guidance Example	139
5.5	Compliance Rate by Justification Type	143
5.6	Interpretability by Justification Type	147
5.7	Justification Type Use-Case Taxonomy	149
5.8	Augmented Reality Justification Testbed	152

Chapter 1

Introduction and Literature Review

1.1 Motivation

Robotic deployments have traditionally fallen into one of two distinct operational paradigms: autonomy or teleoperation. Both have well-established niches throughout the landscape of robotic domains. Some mobile robots (Roombas, self-driving cars) can operate autonomously, while others (bomb disposal robots, space probes) are typically commanded remotely by a human operator. Likewise, for manipulation tasks, assembly lines often employ autonomous robotic systems, while in operating rooms, surgical robots are teleoperated by a human surgeon.

However, these approaches have inherent weaknesses - autonomous robots often struggle with complex and under-defined domains, and can fail in unexpected and potentially catastrophic ways, especially when encountering abnormal scenarios outside their programming or training data. Teleoperation, on the other hand, is not an efficient use of resources, bearing large labor and communication costs which limit scalability, while also being ineffective and unwieldy for many tasks. In an attempt to alleviate these weaknesses, a third paradigm has received much attention in the past twenty years: humans and robots working together as teammates [189, 185, 71].

The key insight here is that humans and autonomous systems excel at different things [79]. By appropriately leveraging each agent's specialized capabilities for joint tasks, team performance can exceed that of teams comprised solely of human or robotic agents. Simply integrating humans into the types of multi-agent planners seen in robotics is extremely difficult, however, since the inherent uncertainty of human behavior hinders efficient optimization. Just like in human-human

teams, when the shared task is complex enough, no teamwork, and thus no performance gains, can be exhibited without plan synchronization. To achieve this, teammates must establish a shared mental model regarding the task and each member’s role within it [249].

In this thesis, the term ‘mental model’ refers to any abstract representation of reality that is used for reasoning and making decisions within a given environment [278]. Though the term originates from human psychology, it can easily be extended to represent the diverse task and environment representations held by robotic agents, which are explicitly defined, as well as the more implicit representations and thought processes used by humans. The concept of a mental model is a slightly broader form of the related term ‘situational awareness’, which refers to understanding of an environment and its changes over time within the context of decision-making [73]. In addition to possessing well-aligned knowledge of an environment, referred to as achieving shared situational awareness [188], teammates with shared mental models are also well-aware of their counterparts’ decision-making and roles as they evolve within a joint task.

The degree of mental model synchronization among teammates is highly correlated with team performance [171]. Unlike human-human teams, though, robots do not yet have access to the implicit social and communication skills that allow teammates to efficiently synchronize mental models with each other [44]. An effective robot teammate should maintain an explicit, formal model of each agent’s conception of the task in progress and its related uncertainties, using communicative acts to rectify and synchronize those models whenever they diverge.

A central problem to address for robot-to-human communication is translating back and forth between the “languages” each agent understands with respect to planning: the states, actions, and mathematical optimization of reward functions for robots, and the language, visualizations, or behavior that are interpretable by humans. Finding compact ways to facilitate this translation is at the core of my research.

The end-goal of this process is bidirectional - not only do we want humans to correctly interpret otherwise opaque robot behavior and understand sources of environmental or plan uncertainty, we also want the act of communication to psychologically nudge human behavior in ways benefi-

cial to team performance [255], especially in domains where robot teammates possess knowledge or expertise that human teammates do not. For example, we may want humans to behave more predictably, in order to aid in multi-agent optimization. In domains where robot decision-making is not always optimal, we may also want to adjust the content of communications conditioned on the robot’s confidence level, nudging humans to accept robot actions or guidance in regions of high confidence without much mental effort, while nudging humans to think more critically and effortfully in regions of low confidence [130].

One technique I employ throughout this thesis is augmented reality (AR) visualization, a technology whose capabilities have already been demonstrated across a number of robotic domains [209, 267, 39]. AR possesses the unique ability to project data directly onto the environment. This in-situ visualization gives shared environmental context for human and robot teammates, in essence providing features for free and enabling compact and interpretable visual communication. What’s more, since AR visuals can be projected through a head-mounted display, there is no need to context switch to a different screen to receive and interpret incoming information [97, 115]. I also take inspiration from the field of explainable AI (xAI), whose techniques have been shown not only to increase human understanding of opaque learning models [105, 29], but also to promote team fluency and improve shared awareness in human-robot tasks [37, 35, 246].

The work presented in this thesis introduces a variety of novel computational methods and interfaces for explicitly synchronizing mental models between human and robot teammates. My work leverages spatially-grounded communication modalities (augmented-reality enabled visual interfaces, presenting information in-situ at key locations within a shared environment, as well as natural language communication tied to those spatially grounded features) to improve human-robot team fluency and performance across multiple partially observable, collaborative domains. These domains include interaction with mobile aerial robot teammates in warehouse fulfilment and environmental search tasks, and interaction with manipulator robots in tabletop assembly tasks.

1.2 Thesis Statement

In human-robot collaborative tasks in uncertain environments, the maintenance of well-aligned mental models between human and robot teammates regarding the shared task and each agent’s role within it has the capacity to greatly improve team performance. Bidirectional, spatially-grounded communication interfaces can be leveraged to improve this mental model alignment during real-time interaction, by:

- (1) Visualizing robot intentions and decision-making, helping to reveal mental model discrepancies between human and robot teammates.
- (2) Enabling expert humans to correct a robot’s mental model through the injection of new information, thus improving robot performance.
- (3) Enabling expert robots to correct a human’s mental model through the provision of explainable guidance and justifications, thus improving human performance.

1.3 Overview

The remainder of **Chapter 1** will present a comprehensive overview of literature on the use of mental models in human-robot teaming, largely as presented in our article entitled “A Survey of Mental Modeling Techniques in Human-Robot Teaming” published in Current Robotics Reports [249]. Next, **Chapter 2** describes a novel augmented reality interface called ARC-LfD, which allows human operators to teach new skills to robots via learning from demonstration, visualize those learned skills in environmental context, and adapt those skills to changing environments and task setups [158]. **Chapter 3** introduces a pair of augmented reality interfaces designed to make human motion in close proximity to robot teammates more predictable, thus enhancing safety and team fluency, and allowing for improved human modeling in a shared tabletop assembly task [258] and a shared warehouse stocking task [40].

Chapter 4 introduces a multi-agent algorithm for environmental search tasks called MARS,

handling both the command of robotic agents, as well as the provision of explainable guidance to human teammates, delivered via augmented reality interface [250]. The chapter concludes by introducing a hierarchical variant of the MARS algorithm called H-MARS, which is capable of adjusting the granularity of its guidance to the current phase of search, simplifying both guidance and computation, and allowing for the algorithm’s use in large, unstructured environments [251]. **Chapter 5** builds upon the findings of Chapter 4 by presenting a method for autonomously generating and timing multi-modal justifications of robot-provided guidance during periods of mismatched expectation, enhancing human compliance with robot teammate suggestions [160]. Lastly, **Chapter 6** summarizes the contributions of this thesis and suggests directions for future research stemming from its findings. Each chapter describing novel research begins with a “Motivation” section, indicating how that research fits within the broader context of the thesis.

1.4 Literature Review

1.4.1 Introduction

Traditionally, robots have worked separately to humans. Even in potentially collaborative environments like manufacturing, industrial robots most often operate in physically separated sections of the assembly floor. This work scheme of rigidly divided responsibility and prohibited human-robot interaction (HRI) prevails for reasons of safety and simplicity, but limits applications of these robots to strictly defined, well structured, repetitive tasks [74]. Advances in autonomy are rapidly improving robots’ ability to interact with, and even directly collaborate alongside human teammates, opening up a wide range of new and impactful applications that leverage the unique skills of human and robot alike [189, 183, 71, 206].

A key aspect of effective and fluent teamwork among humans is maintaining awareness of what teammates are likely to do or need, so as to coordinate actions. Humans tend to be adept at this task, able to communicate plans and preferences easily understandable by their teammates [44]. Robots, however, do not have the benefit of human intuition. They must instead rely on

explicit mathematical formalisms in order to approximate the mental states of human teammates and plan accordingly. This literature review focuses on characterizing recent work in developing these formalisms, known as **mental models**. In the following sections, we discuss the context and aims of mental model research for human-robot teaming, as well as describe and categorize the common methodologies, usage, and evaluation of such techniques.

1.4.2 Mental Models

Mental models, also referred to as **mental representations** in psychology, are organized knowledge structures that allow individuals to interact with their environment [278]. 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 [54]. In essence, mental models serve the crucial purpose of helping people to describe, explain, and predict events in their environment [171]. 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 [50, 169]. 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 [200]. 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 [93]. Accordingly, several architectures for human-robot teaming in HRI incorporate aspects of a ToM for other agents [63, 288, 94, 219, 152, 186].

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 [25]. 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 [171, 21, 176]. 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 [49, 128]. 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.

1.4.3 Mental Models in Human-Robot Teaming

Teamwork is the collaborative effect of a group's effort toward achieving a common goal [216]. 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 [171].

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” [116]. 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 [49, 184].
- **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) [9, 247].
- **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 [246, 254].
- **Explainability:** Knowledge sharing and expectation matching also have importance for behavior explainability [271, 173, 263]. 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.

1.4.4 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.

1.4.4.1 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 [8].

One of the simplest approaches is based on the principle of rationality [62, 88]: 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 [182, 289].

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) [12, 14]. 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) [129], and reconstructs an

agent’s joint belief state and reward function using Bayesian inference based on observations of the agent’s behavior [11, 13].

From a planning and decision-making point of view, the noisy rational choice model (also known as Boltzmann rational) [192, 191] 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 [25, 212, 196, 193, 70].

Humans frequently deviate from rational behavior due to specific biases such as time pressures, loss aversion, and the like [259]. 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 [234]. 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 [186]. 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 [143].

1.4.4.2 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 [90, 275]. 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 [123].

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 [89] 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 [25].

1.4.4.3 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 [171]. 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 [185]. 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 [107].

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 [184].

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 (behavior). 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 [70]. 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 [140].

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 [148]. 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 (natural language and visualization). There are numerous techniques for deliberately generating communicative content to share with a human collaborator. 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 [113, 33, 69]. 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 [99] of effective conversational communication (quality, quantity, and relevance) [24]. 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 [247, 174, 114, 37].

Explicit communication may also include visual information presentation in addition to, or in place of, natural language [245]. In scenarios where data to be communicated is complex, evolving, and re-referenced, visual communication has been shown to lead to better comprehension by a human recipient [60]. Robotic behavior frequently possesses a spatio-temporal component, which aligns well with the aforementioned criteria. Visualization is frequently used in human-robot teaming for tasks such as environmental navigation, search and inspection, and fault recovery [122, 45, 136]. The form factor of provided visualizations is highly dependent on the nature of the shared task, often taking the form of HUD-style monitor interfaces for tasks with a remote human teammate [87, 95], while recent work has leveraged augmented reality (AR) interfaces to provide in-situ visualizations for shared-space collaboration [209, 267].

1.4.5 Evaluation Methods

In this section, we discuss evaluation methods employed in human-robot teaming for each of the desirable traits characterised in Section 1.4.3.

Team Fluency. Fluency, the metric for well synchronized meshing of joint actions between humans and robots, is difficult to measure and optimize in practice [256]. 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) [117]. 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 [116].

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 [184] and similarity metrics between human and robot notions of “correct action sequence” in an evolving task [185]. 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 [116, 185]. Nikolaidis et al. have additionally showed that accounting for individual differences in humans’ willingness to adapt to a robot is positively correlated with trust [186].

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 [149]. Based on previous studies, robot performance is considered to be the most influential factor for trust [109], likely due to the importance of the agent’s ability to meet expectations [145]. Other factors with positive relationships to trust are minimizing system fault occurrence, system predictability, and transparency [150]. 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) [150, 225].

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

can be explicit or implicit, as seen in Section 1.4.4. 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) [284, 43, 35]. 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 [70, 144, 141].

Explainability. Explainability deals with the understanding of the mechanisms by which a robot operates and the ability to explain robots' behavior or underlying logic [246, 114]. 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) [269].

1.4.6 Emerging Fields & Discussion

Mental models have proven beneficial for many human-robot teaming applications such as assistive and healthcare robotics [71], social path planning and navigation [206], search and rescue [189], and autonomous driving [213, 143]. 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 [108], 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 [260]. 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 [38, 91]. 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) [113, 17].

Furthermore, developments in augmented reality (AR) technology have shown promise for industrial and field 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 [209], and mobile robots [267], 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 [157, 98].

Behavior manipulation, also known as policy elicitation, 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) in order to successfully complete a task, either through implicit or explicit communication [246, 214, 37]. This represents an emerging variety of robotic decision support. Various challenges related to behavior manipulation include accurately modeling human behavior [246], leveraging human models to find failure modes [248], and succinctly generating persuasive human intelligible semantic or visual updates (or executing mitigating actions) [114, 250, 160]. With the currently observed rate of increase in agents' capability for social behavior and persuasive

natural language generation, especially with the emergence of highly generalizable large language models such as ChatGPT, human teammates run the risk of over-relying on potentially flawed autonomous guidance. Proper calibration of trust remains an open problem in robotic decision support [207].

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 literature review, 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. As evident in this summary, there are many exciting developments within this space, as well as many open and challenging problems to drive future research.

Chapter 2

Bidirectional Augmented Reality Interface for Enhancing Robot Skill Demonstration and Repair

2.1 Motivation

In this chapter, we present a novel augmented reality (AR) system for constrained robot learning from demonstration (LfD) called ARC-LfD. ARC-LfD allows learned robot skills to be visualized directly in environmental context through an AR headset, so human operators can tell prior to skill execution whether their intent has been correctly captured by the robot, and quickly remedy errors. Additionally, ARC-LfD allows for the injection of high-level constraint information to quickly adapt a base set of trajectories (representing a general skill) to changing environments and task requirements without requiring additional demonstrations. This ability to rapidly edit learned skills in-situ via AR interface improves the flexibility of LfD methods for long-term robotic deployments. We evaluate the system using a series of case studies, showcasing ARC-LfD’s ability to adapt existing skills to multiple target environment and task specifications through the strategic introduction of high-level constraints to portions of the trajectory. This work was presented at the IEEE International Conference on Robotics and Automation (**ICRA 2021**) [158].

Within the broader context of this thesis, this chapter explores the idea of bidirectional spatially-grounded information transfer, enabling alignment of human and robot task representations. On one hand, the robot is able to communicate its current learned skill model to a human supervisor, visualized as a trajectory directly in environmental context. This acts as a debugging step, allowing the human to critically evaluate the robot’s understanding of the environment and

task prior to execution, and identify any discrepancies in need of repair.

On the other hand, the human is able to communicate their own high-level task knowledge, through the creation and assignment of constraints within the same spatially-grounded AR interface, directly shaping robot learning. In LfD setups such as this, humans take on an expert teaching role. They are assumed to have a higher base knowledge compared to the robot, representing something close to ground truth that the robot’s behavior should approach. Because of this, the decisional authority regarding the identification of model discrepancies ultimately rests with the human, and the primary direction of information transfer is human-to-robot. In future chapters, we explore cases where this relationship is inverted - where robots possess more knowledge than their human teammates, and must both identify when mental models diverge, and choose how best to rectify them when they do.

2.2 ARC-LfD: Using Augmented Reality for Interactive Long-Term Robot Skill Maintenance via Constrained Learning from Demonstration

2.2.1 Introduction

Robot learning from demonstration (LfD) methods enable users to teach desired skills to robots without programming or other forms of robot-specific knowledge [7, 10]. The predominant focus of LfD research to date has been on the initial learning process itself, rather than the maintenance and adaptation of learned models. In a shift of focus to the latter, we introduce **Augmented Reality for Constrained Learning from Demonstration** (ARC-LfD): a system that combines an augmented reality (AR) interface and constrained learning from demonstration [180] to enable users to teach a robot new skills as well as verify, repair, and edit existing skills. ARC-LfD demonstrates a novel approach to LfD that can mitigate problems arising from poor quality demonstrations, changes in the environment, and adaptations to the task procedure.

When using LfD methods for robot instruction, safe deployment necessitates verification that a skill has been learned properly after the skill has been demonstrated and taught. While

verification can be done in simulation, this requires a high-fidelity model of the environment in order for the visualization of the learned skill to be shown in the proper context (and obtaining such a model may be a technical endeavor). After this step is completed, the robot may begin executing the learned skill as long as the environment stays constant, but even small changes in the robot’s environment or the desired skill may require an entirely new set of demonstrations to fix it. This requirement for rigidity of environment and task can make long-term deployment and maintenance of the skill difficult in practice.

One approach to handling this rigidity is the creation of end-to-end policy learning systems that aim to model skills more generally. However, such systems may demand a prohibitive number of demonstrations or require unavailable simulation environments to capture user intent, and aren’t designed to accommodate user selection of task constraints. Our approach emphasizes transparency and adaptability in a system designed for online skill editing and validation necessary for long-term robot deployment. Through AR visualization, ARC-LfD safely demonstrates to users what skill has been learned and how executing that skill will cause the robot to move through the environment. The AR interface also facilitates the visualization and editing of constraints, enabling users to see how these constraints interact with objects or points of interest in the environment. Furthermore, constraint editing through AR allows the entire training process to take place in-situ without requiring context-switching between the real environment and a 2D display.

The contributions of the ARC-LfD system are as follows:

- (1) AR visualizations of learned skills, in-situ robot behavior, and constraints without needing a model of the entire environment.
- (2) An iterative process to verify, repair, and edit existing skills through AR using visualized constraints employed by the underlying LfD algorithm.
- (3) Three case studies that illustrate how the system enables skill adaptation with no further demonstration.

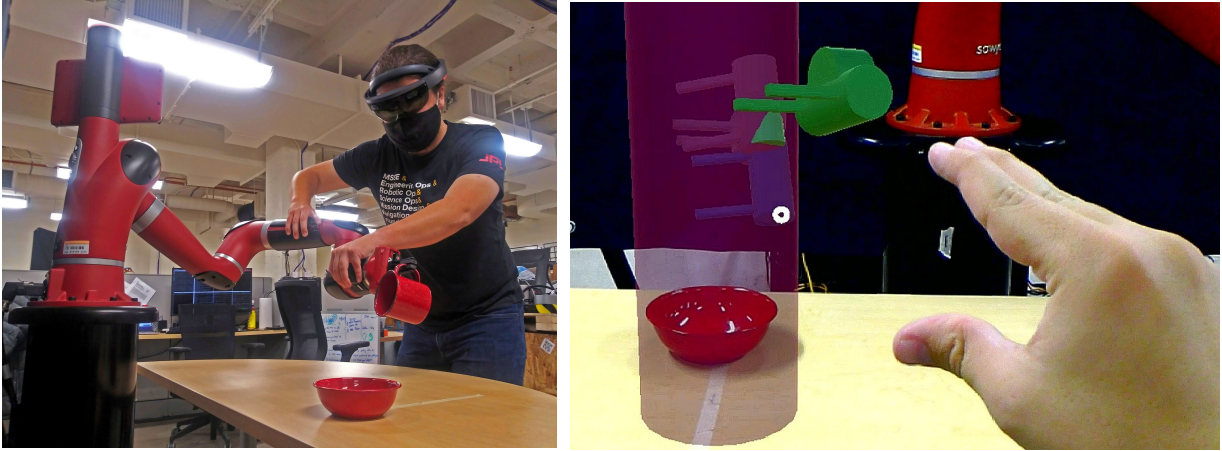


Figure 2.1: ARC-LfD combines augmented reality with constrained learning from demonstration to create a system that enables the teaching, verification, editing, and updating of robot skills using in-situ visualizations.

2.2.2 Related Works

2.2.2.1 Learning from Demonstration

Robot learning from demonstration (LfD) encompasses a set of methods that strive to learn successful robot behavior models from human input [10]. A human operator interacts with a robotic system through some mode of demonstration, usually through kinesthetic demonstration (e.g., physical interaction), teleoperation (e.g., remote control), or passive observation (e.g., motion tracking observation). While the mode may vary, demonstrations ideally communicate the nature of the skill to the robot such that the learned model effectively resembles some latent ground truth model held by the demonstrator [7]. The methods by which robotic systems learn such models span a broad spectrum, but are generally categorized into three classes: 1) plan learning, 2) functional optimization, and 3) policy learning [203]. Most importantly, LfD methods enable non-roboticists to quickly teach robots useful skills and forgo the need for expert robotics programming knowledge.

ARC-LfD uses an LfD method that falls under the policy learning categorization, where the goal is to learn models that output either robot trajectories or low-level actions directly. Work by Akgun et al. [3] introduces Keyframe-based LfD, a method that learns a sequential waypoint (i.e.

keyframe) model of a skill through the clustering of demonstrated trajectories. Keyframe models essentially produce coarse trajectories for the robot to execute by employing motion planning algorithms to traverse from waypoint to waypoint.

ARC-LfD utilizes an enhanced variant of this technique called **Concept Constrained Learning from Demonstration** (CC-LfD) [180]. During demonstration, users annotate behavioral constraints (through real-time dictation) to be applied to the learned model. Akin to prior work in learning from human teachers [28, 125], this algorithm is motivated by the insight that although traditional state data captured by the robotic learner does encode certain aspects of the task, the users’ internal model might have latent information not communicated through traditional kinesthetic demonstration. Thus, by enabling the user to also communicate behavioral constraints (e.g., “a cup must remain upright until over the bowl”), the robotic learning system is given additional information that helps produce a more robust and successful model. To this end, CC-LfD requires far fewer demonstrations to teach a successful skill model than robot state demonstration trajectories alone produce, and enables post-hoc skill repair and adaptation through constraint application.

Keyframe-based LfD methods are agnostic to the mode of demonstration as they operate on the resulting trajectories. However, ARC-LfD utilizes kinesthetic demonstration, where users physically manipulate the robotic system to produce demonstration trajectories. Akgun et al. [4] showed that kinesthetic demonstration generally produces more successful skill models and is the preferred mode of demonstration by end-users when compared with teleoperation. However, Wrede et al. [279] described how kinesthetic demonstration can be limited by non-experts users’ lack of robotics knowledge. For example, they showed that resultant models learned through kinesthetic demonstration perform poorly when users guide robots close to configuration space Jacobian singularities. Furthermore, Villani et al. [264] surveyed a multitude of industrial environments in which robots are deployed, describing highly variable and potentially dangerous collaborative environments and tasks. Such environments challenge kinesthetic demonstration as complex structures and dangerous conditions make kinesthetic demonstration infeasible to model or unsafe for humans.

Given these concerns, safety and adaptability become paramount, both for the design of safe human-robot collaborative environments [228] and for the mechanisms by which robots build skill models [19, 180]. The ARC-LfD system utilizes an AR interface that enables users to both visualize learned skills and to define a strict set of behavioral restrictions via the application and editing of constraints. The benefit is twofold: 1) constraint application helps facilitate encoding additional information, shifting the burden of end-user expertise away from robotics and towards the task consideration, and 2) AR enables a user to operate in an environment where certain features (dangerous objects, difficult arrangement, etc.) that make kinesthetic demonstration either infeasible or dangerous can be virtualized, communicating skill-essential behavioral restrictions as encoded constraints.

2.2.2.2 Augmented Reality Interfaces for Robotics

In order to facilitate an additional visual interface for an LfD system without requiring user context-switching [115], we use AR. AR interfaces for robotics have a proven track record [97, 243], enabling new methods of enhancing robotic control [282, 272, 268, 26], collaboration in human-robot teaming [39, 208], safe movement in shared spaces [267, 209], and communication of robot knowledge [134, 137, 64]. Motivated by this existing body of work, we use AR to create an interface for LfD that previews learned skills and allows editing of constraints directly in the robot’s environment.

Through ARC-LfD, users are able to examine a sample trajectory from a learned skill visualized in AR through an overlay in the workspace environment. Such skill visualization is intended to improve safety as the operator can “preview” robot behavior without the need for actual skill execution [138]. Prior work has established this potential through user studies: Walker et al. [267] conducted a user study which found that showing flying robot paths in AR made users more efficient and comfortable when sharing an environment with these robots. Similarly, Rosen et al. [209] found that AR visualization of possible robotic arm trajectories improved participants’ accuracy and quickness in identifying collisions with objects in the environment. These studies substanti-

ate the notion that AR visualizations of robot trajectories may improve user understanding with respect to the path a robot will take and how that trajectory will interact with the environment.

In addition to visualizing the robot’s possible future movement, ARC-LfD supplies visual cues that describe the robot’s ability to adhere to user supplied behavioral constraints on a learned skill. This is akin to helping users understand the internal state of the robot, another functionality that has been explored within the space of AR for human-robot interaction. Through AR, information such as the robot’s battery life [137] or sensor readings [134] can be communicated to users through a heads-up display. This is particularly useful when performing complex tasks such as controlling a robot as it prevents disruptive context-switching when averting attention away from the environment towards a 2D display [115]. Using AR to visualize a robot’s knowledge in the form of a learned skill or action can also provide a realistic demonstration of this knowledge without requiring extensive modeling of the environment to use in simulation [64].

The final type of interaction supported by AR in ARC-LfD is the ability to create and manipulate constraints on a learned skill. Visualizing constraints in the physical environment allows users to see the exact effect of applying these constraints [239]. Yamamoto et al. [282] illustrated that applying virtual constraints was an effective tool for robot-assisted surgery, allowing surgeons to specify thresholds that the robot should not cross. In our case, the constraints are both shown and edited in the environment in which the skill will be executed, allowing users to move constraints around physical objects to ensure the skill can be performed safely.

Generally, we are motivated in designing ARC-LfD by a rich history of research into LfD as well as strong results from prior work at the intersection of AR and robotics that demonstrate AR interfaces outperform 2D and tablet-based interfaces for visualizing information critical to human-robot interactions. In the next two sections, we describe the algorithmic basis for ARC-LfD followed by the design and capability features of the AR interface.

2.2.3 Concept Constrained Learning from Demonstration

As shown in Figure 2.2, ARC-LfD consists of two components communicating via the Robot Operating System (ROS): a Concept Constrained Learning from Demonstration subsystem (CC-LfD), which serves as a backend for skill learning, and an AR subsystem for visualization and user interactions with a learned skill. In this section, we present an overview of CC-LfD; however, we point the reader to the original paper for a more thorough review: [180].

CC-LfD is an augmentation of keyframe-based learning from demonstration [3] that incorporates the ability to utilize constraints, consisting of concepts (e.g., “X is above Y”, “Z is powered on”, etc.) encoded as Boolean planning predicate classifiers, to produce a more representative learned model of the demonstrated skill. The motivation behind incorporating predicate-based constraints is to overcome the limited capacity of demonstrated robot state (e.g., end-effector state) trajectories alone to encode all critical aspects of a skill that a human operator intends the robot to learn. For example, when teaching a robot a cup carrying task, robot state data alone will not adequately capture the concept of “keeping a cup upright.” By leveraging logical combinations of predicate-based constraints, CC-LfD biases waypoint sampling from learned keyframes, resulting in a dramatic reduction in the required number of demonstrations to both train a successful model and repair a poorly performing skill as compared to introducing additional high-quality demonstrations.

The CC-LfD algorithm requires a set of demonstrated robot trajectories annotated with constraints. These trajectories are aligned via Dynamic Time Warping [215] to preserve point-to-point spatio-temporal similarity across trajectories. Once the trajectories are aligned, annotated constraints are combined via a Boolean logical *AND* across all demonstrations. Sequential clusters of aligned trajectory points provide the basis for the nodes of a directed acyclic graph representative of a learned skill.

Individual keyframe models are created by fitting distributions on the data within each cluster. Keyframes inherit the set of constraint annotations preserved during the alignment step. Importantly, constraint set change-point regions demarcate special keyframes of data known as

boundary keyframes. Consecutive keyframes whose variational distance is below a threshold parameter are culled from the keyframe graph. This produces a more sparse keyframe representation and eliminates backtracking behavior during skill execution. Boundary keyframes are never deleted as they represent pertinent structural change-points for the learned skill. To better ensure each keyframe is representative of a constraint-compliant distribution, a rejection sampling step produces a constraint-compliant set of points that is used to rebuild the keyframe distributions. Finally, skill execution is accomplished by sequentially sampling constraint-compliant waypoints from a directed path through the keyframe graph, subsequently constructing motion plans between waypoints.

ARC-LfD introduces an advancement over CC-LfD by enabling post-hoc application of constraints as opposed to requiring constraint application during demonstration. This new approach facilitates an iterative update process that alters keyframe constraints and the corresponding distributions, providing the basis for ARC-LfD to achieve skill adaptation. ARC-LfD first generates an initial keyframe model of the skill (Fig. 2.3, Step 1), which is visualized as an instantiation of the keyframe waypoints that the robot will execute (Fig. 2.4). This visualization includes the validity of each waypoint relative to the keyframe’s applied constraints (Fig. 2.3, Step 2). Using the AR interface, the user generates new constraints, or edits existing constraints (Fig. 2.3, Step 3), and assigns them to a chosen keyframe. This initiates a model rebuilding phase where keyframe distributions are relearned using the same rejection sampling and distribution fitting steps as CC-LfD (Fig. 2.3, Step 4). If the user is satisfied with the visualized robot behavior, skill execution can proceed as carried out by the CC-LfD algorithm (Fig. 2.3, Step 5).

2.2.4 Augmented Reality System Design

The second subsystem of ARC-LfD (see Fig. 2.2) is an AR interface deployed on a HoloLens, a mixed reality headset developed by Microsoft. A headset was chosen over alternative tablet-based passthrough AR solutions due to its hands-free nature, freeing users’ hands for interaction with the robot, and its ability to show different imagery to different eyes, enabling superior depth perception [172]. Users wearing the HoloLens are able to see holographic visualizations of relevant keyframes

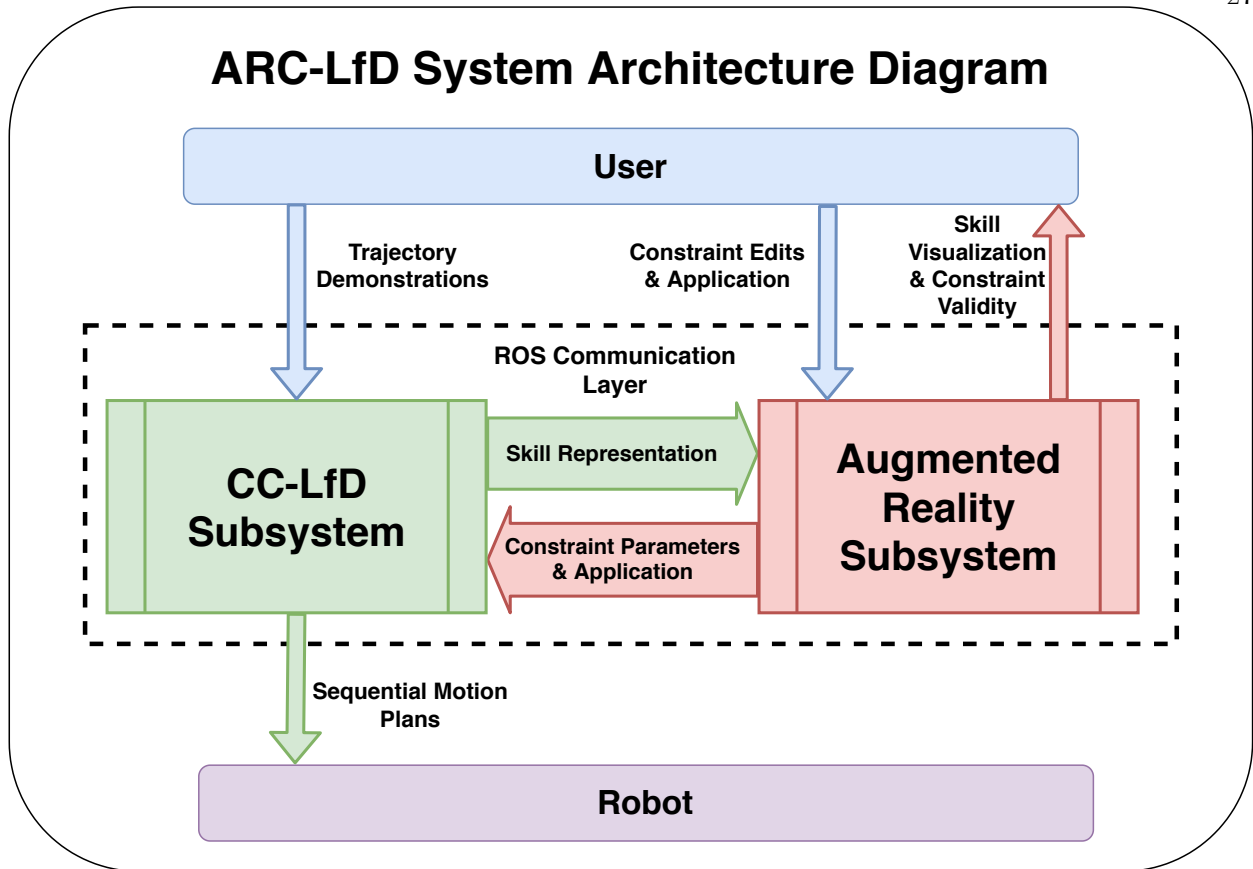


Figure 2.2: A diagram of the ARC-LfD system architecture. The user (blue) supplies the initial demonstrations to the CC-LfD subsystem (green). During the editing phase, the user also supplies constraint edits and their keyframe application to the AR subsystem (red). In return, the AR subsystem supplies skill and keyframe constraint validity visualizations to the user. Through a Robot Operating System (ROS) communication layer, the CC-LfD and AR subsystems exchange skill representation, constraint parameterization, and constraint application information. Finally, the CC-LfD subsystem provides sequential motion plans for the robot (purple) to execute.

and constraints projected onto the robot’s workspace. User interaction is achieved through performing pinching gestures known as *air taps* on these visualizations and on menu buttons pinned above the robot (see Fig. 2.1).

2.2.4.1 Skill & Constraint Representation

For a given skill, each keyframe generated by CC-LfD is sent to the AR interface and visualized as a hologram of the robot’s end-effector, whose position and rotation are representative of a randomly sampled valid waypoint within that keyframe. The combination of these keyframe visu-

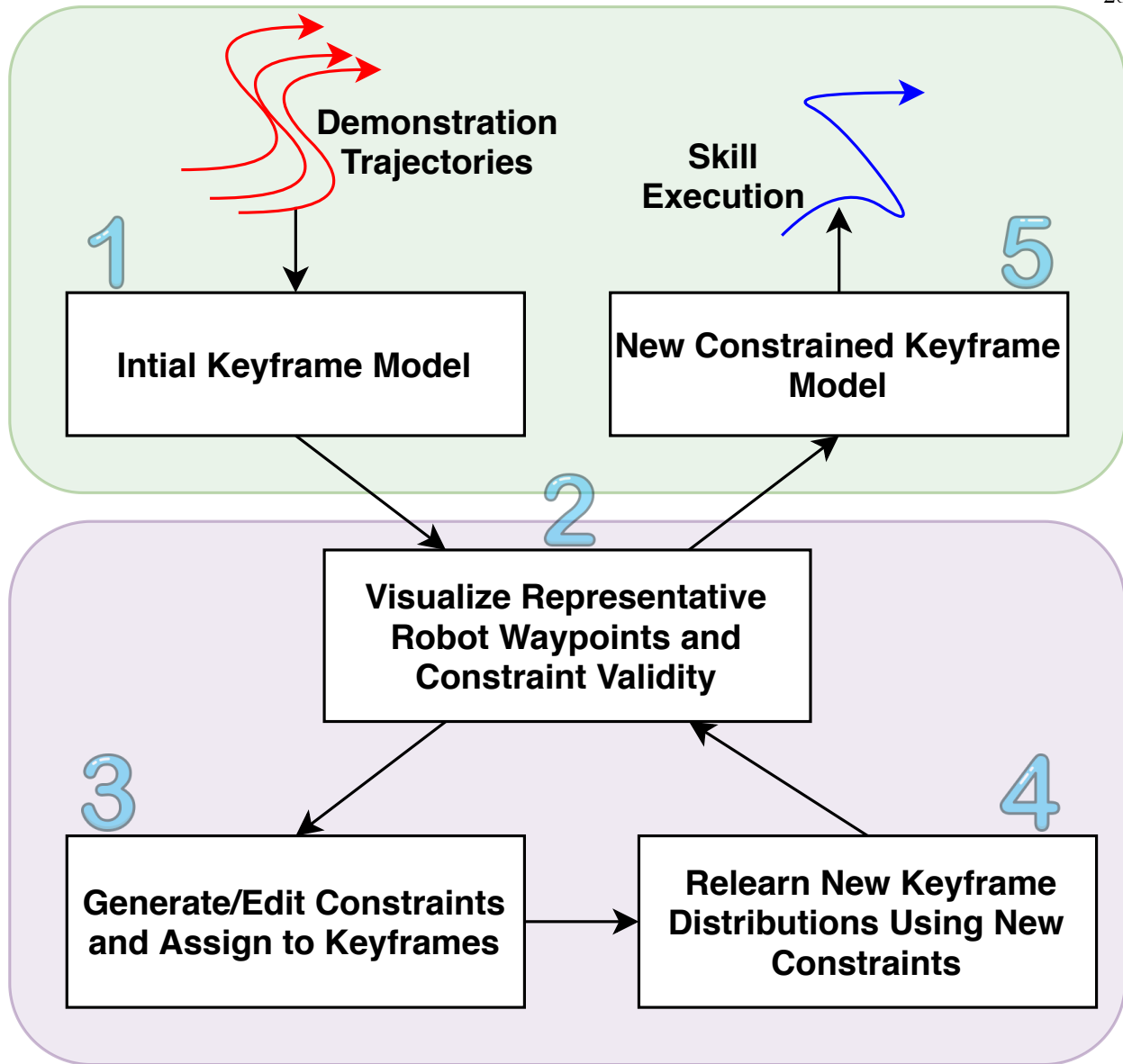


Figure 2.3: Flowchart indicating how ARC-LfD integrates into CC-LfD. Steps 2, 3, and 4 repeat until the user is satisfied. The pink region (bottom) indicates AR-based steps whereas the green region (top) indicates that AR is not strictly required.

alizations traces out a trajectory that the robot would follow to execute the skill. To aid the user in evaluating a candidate trajectory at a glance, the end-effector holograms are colored in a gradient from green to gray to indicate the ordering of the keyframes, and any waypoints in violation of an applied constraint are colored bright red (see Fig. 2.4).

Our test implementation incorporates three constraint types, representing a subset of possible

Table 2.1: Editable constraints and adjustable parameters in ARC-LfD

Editable Constraints			
Constraint Type	AR Visualization	Parameters	Example
Height	Plane w/ Arrows	Reference Height, Direction	Fig. 2.4, top-right
Orientation	Cone and Fan	Direction, Affordance Angle	Fig. 2.4, bot-left
Over/Under	Cylinder	Position, Validity Radius	Fig. 2.4, bot-right

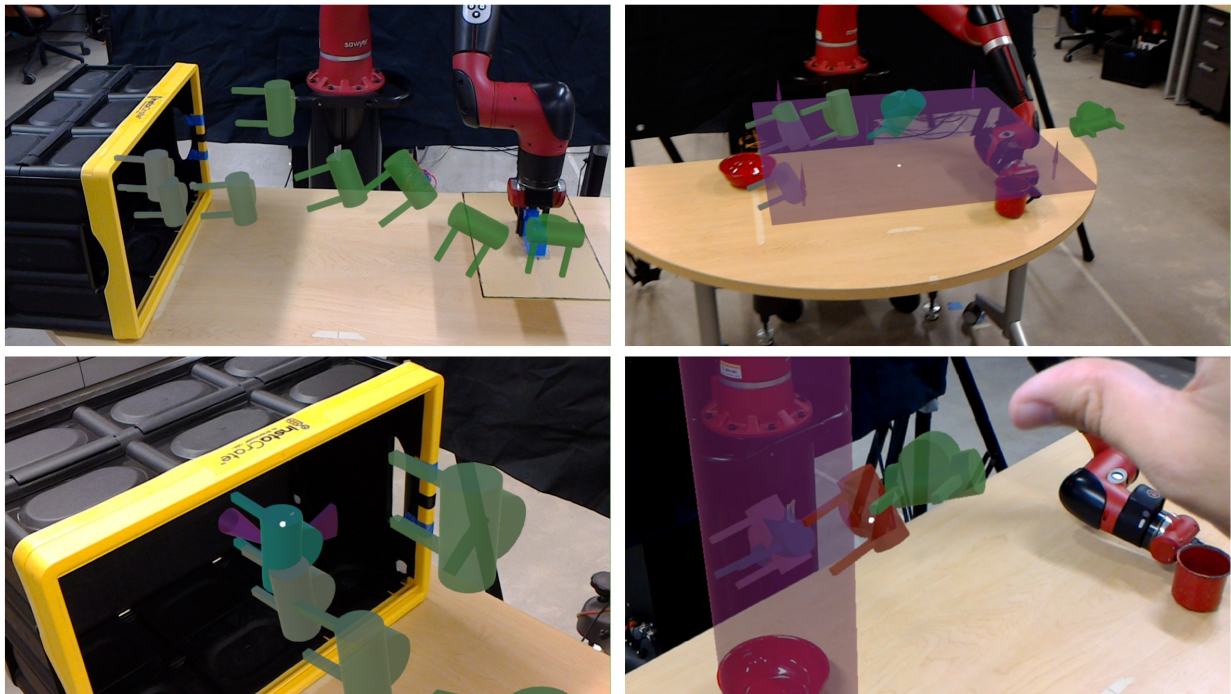


Figure 2.4: ARC-LfD allows the user to visualize trajectories as a series of keyframes (top left). Selecting a keyframe will show holograms representing any constraints active at that keyframe, such as the height constraint (top right) indicating the end-effector must stay above the plane, the orientation constraint (bottom left) overlaid on the selected end-effector to show its proper rotation, and the over-under constraint (bottom right) indicating the end-effector must stay within the cylinder. Note that in the bottom right image, one keyframe has the over-under constraint applied, but is not located inside the cylinder, placing it in violation of the constraint, and coloring its hologram red to alert the user.

parametric, predicate-based constraint templates for ARC-LfD, selected to provide coverage over a number of common robotic manipulation task setups. These are height constraints (the robot’s end-effector must stay above or below a given height), orientation constraints (the robot’s end-

effector must maintain a given rotation, within a given affordance), and over-under constraints (the robot’s end-effector must stay above a given location, within a given radius). Each constraint type has its own associated visualization: a plane with arrows indicating the valid direction for height constraints, a cone and fan overlaid onto an end-effector showing the affordance for each axis for orientation constraints, and a cylinder representing the radius around a target for over-under constraints. When the user selects a keyframe with a constraint applied, that constraint hologram appears, positioned, rotated, and scaled according to its parameters, and colored a translucent purple to maximize visibility of the trajectory and environment. For a summary of these constraints, their AR visualizations, their editable parameters, and references to examples, see Table 2.1.

2.2.4.2 Constraint Editing & Application

ARC-LfD lets users edit existing constraints and create new ones from a template via the AR interface (see Fig. 2.5). The user accesses the constraint editing interface by selecting a constraint type and slot with the menu buttons above the robot. The user will then have the trajectory visualization cleared from their view and a lone constraint visualization will be rendered. The user can edit the parameters of their chosen constraint type (see Fig. 2.5), seeing the visualization update in real-time, which allows them to match constraints to environmental features (e.g., placing an over-under constraint on top of a target object for a pick-and-place task).

Once a user is satisfied with their new constraint, they press a confirmation button, which synchronizes the representation across the AR and CC-LfD subsystems of ARC-LfD. They are then able to apply that constraint to a keyframe or range of keyframes through the constraint application menu until they have added the constraint to the desired areas of the skill trajectory. Once this process is complete, and the trajectory has been satisfactorily inspected, the user selects the “Send to Robot” button to send the new constraint application to the CC-LfD subsystem, which initiates a rebuilding and resampling of the skill. After the CC-LfD subsystem has relearned a set of new keyframe distributions, it sends them back to the AR subsystem and updates the trajectory visualization to inform the user if the system adequately captured their intent, and whether the

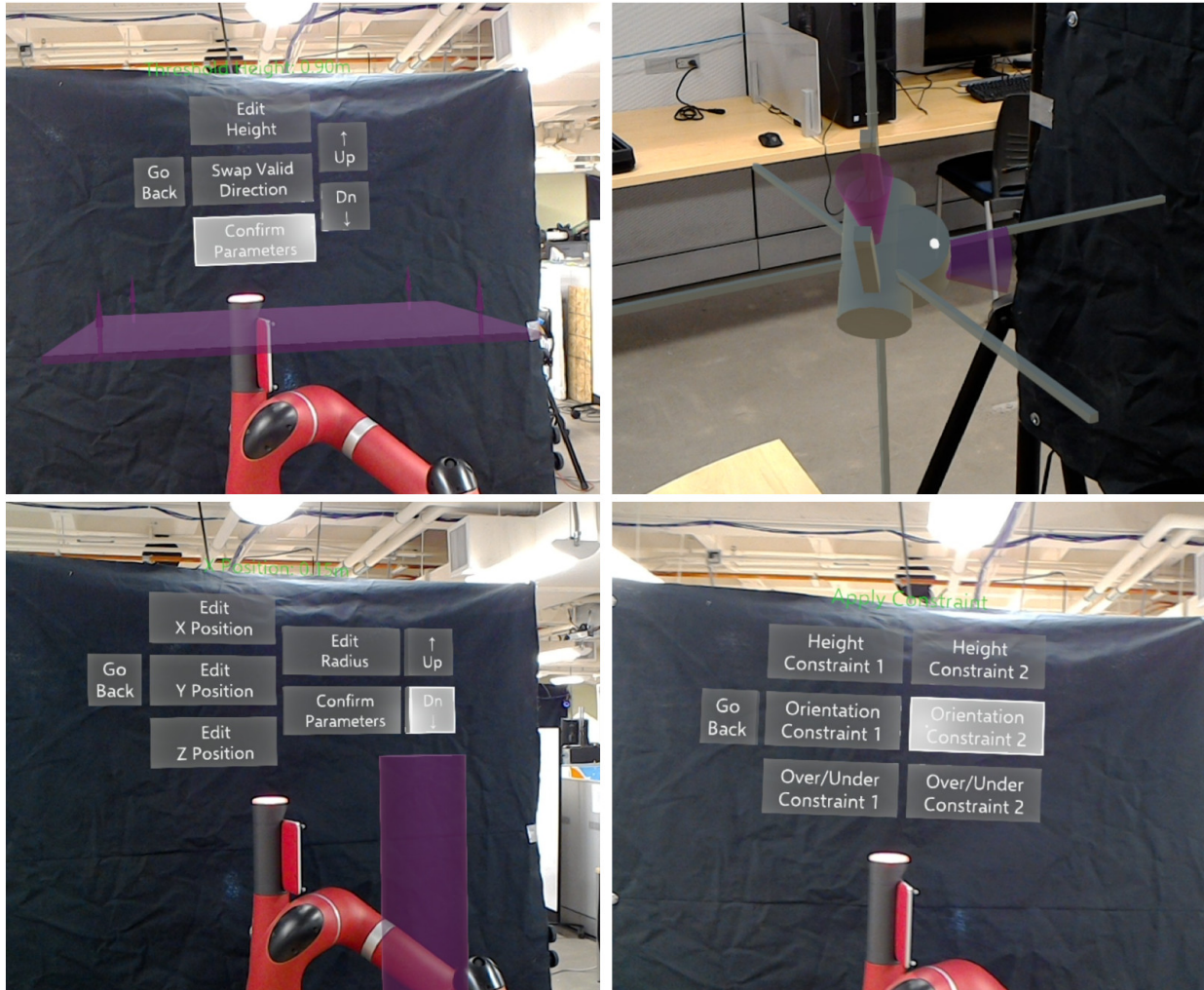


Figure 2.5: Users can customize constraints from templates via the AR interface. After selecting a height (top left), orientation (top right), or over-under (bottom left) constraint, they edit its parameters and see the corresponding visualization update in real-time. Once satisfied, they can apply the newly edited constraint to the model by selecting it from the application menu (bottom right), and by selecting which keyframes the constraint should apply to. After this process, they will send a request to the robot to rebuild and revisualize the model using any new constraints, and evaluate whether the robot has correctly learned the skill.

skill is likely to be executed successfully. This process of trajectory evaluation, constraint editing, and constraint application can be repeated until the user is satisfied.

2.2.5 System Validation

In order to validate the ARC-LfD system, we examine its operation within three test cases representative of potential task scenarios asked of robot manipulators. These case studies exemplify how ARC-LfD allows a user to demonstrate a skill, visualize the learned skill, then adapt the learned skill to two different environment setups (an “initial setup” and “secondary setup”) using edited constraints. One of our research team members acted as a user to demonstrate the system’s functionality. Eight kinesthetic demonstrations were provided as the basis for each skill using the Rethink Robotics Sawyer platform. Once the ARC-LfD system had generated a skill model learned from these demonstrations, the user was shown a sample trajectory of this skill. The user then edited and applied constraints with consideration given to the specific environment setup. ARC-LfD used the applied constraint to adapt the initial learned skill and sent a representation of the updated skill back to the user for visual inspection. Finally, the skill was executed on the robot.

These case studies demonstrate situations in which ARC-LfD allows a user to assess and edit a skill in response to changes in the environment or task setup. This illustrates a novel capability over CC-LfD as a user can craft and visualize constraint annotations to ensure successful model adaptation to differing task setups sans additional demonstrations. In these example applications, the entire process (skill visualization, creation and application of a constraint, skill updating within the CC-LfD subsystem, visualization of the updated skill, and approval of execution) took an average of 120 seconds per skill. Videos of the execution from each case study can be found at: <https://youtu.be/G9TJIKVod4A>.

2.2.5.1 Case Study I (Precise Placement)

The first case study emulates situations in which the goals of the task are modified after initial demonstrations are given. In this task, the robot’s objective was to place a rectangular object into an upright crate, with minimal clearance. If the object was placed using the wrong orientation, a collision with the crate would occur. The user first provided demonstrations with

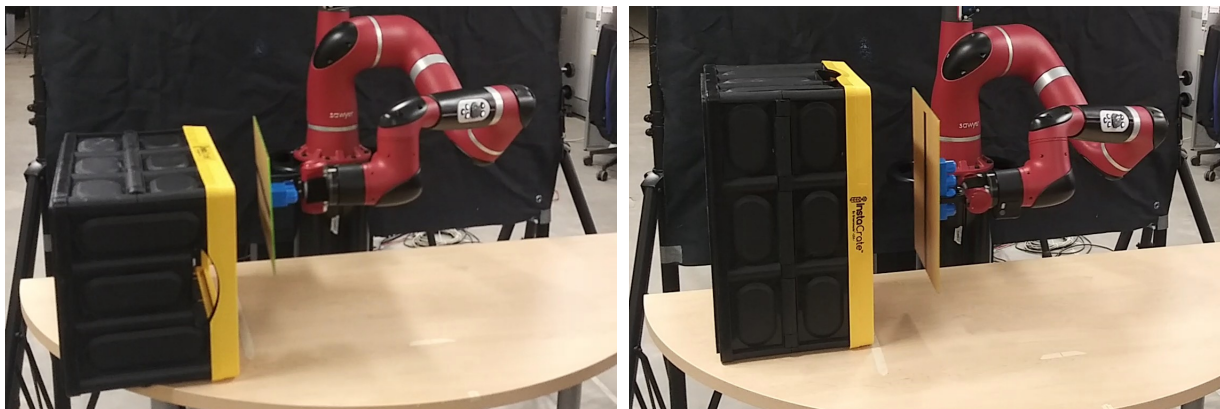


Figure 2.6: For Case Study I, the robot inserts a rectangular object into a similarly-sized rectangular crate. In this case study, the user applies orientation constraints to the final keyframes in the trajectory in order to match the initial setup (left) with a horizontal crate or the secondary setup (right) with a vertical crate.

varied orientations of the object. We evaluate the task for two different orientations of the crate, horizontal and vertical, with no additional demonstrations provided between conditions. In both cases, the user applied an orientation constraint to the last few keyframes of the task specifying the respective desired orientation. With the added constraints, the ARC-LfD system enabled the robot to successfully place the object without collision. The setup of this case study is shown in Figure 2.6.

2.2.5.2 Case Study II (Changing Environment)

In the second case study, the robot’s task involved moving an object from one side of a table to another. This task is representative of pick-and-place kitting tasks with known initial/goal locations but configurations of obstacles that may change over time. For this case study, the user provided 8 demonstrations of moving the robot’s arm across the table from right to left. The initial environment setup had no obstacles in the way. In the test condition, we placed stacked foam obstacles halfway across the table. By applying a height constraint, the user is able to edit the skill so that the robot could still complete the task without colliding with the new obstacles, without requiring additional demonstrations. This case study exemplifies how a generic constraint can be

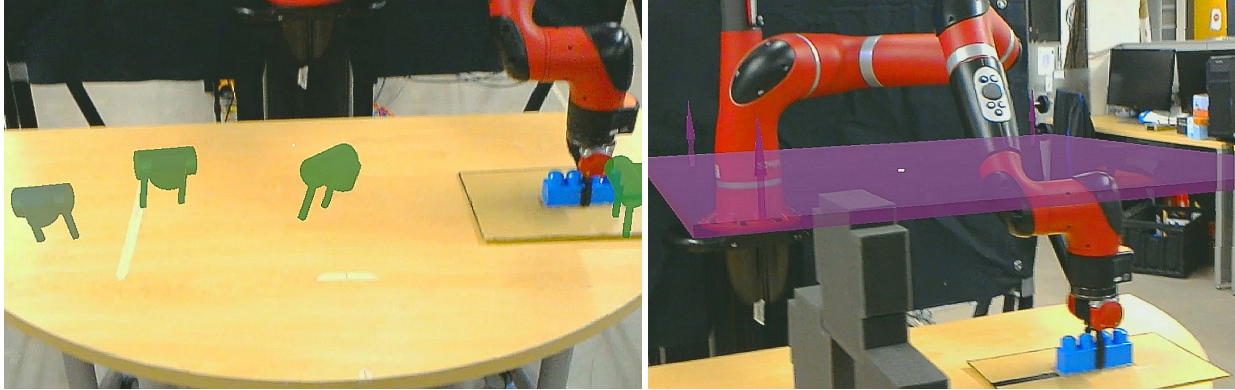


Figure 2.7: In Case Study II, the robot completes a pick-and-place task either with or without an obstacle present. The initial environment (left) has no obstacles on the table, allowing the robot to freely move the object from right to left across the table. The test condition setup (right) introduces an obstacle halfway across the table, requiring the user to apply a height constraint that ensures the robot lifts its payload over the obstacle to complete the task.

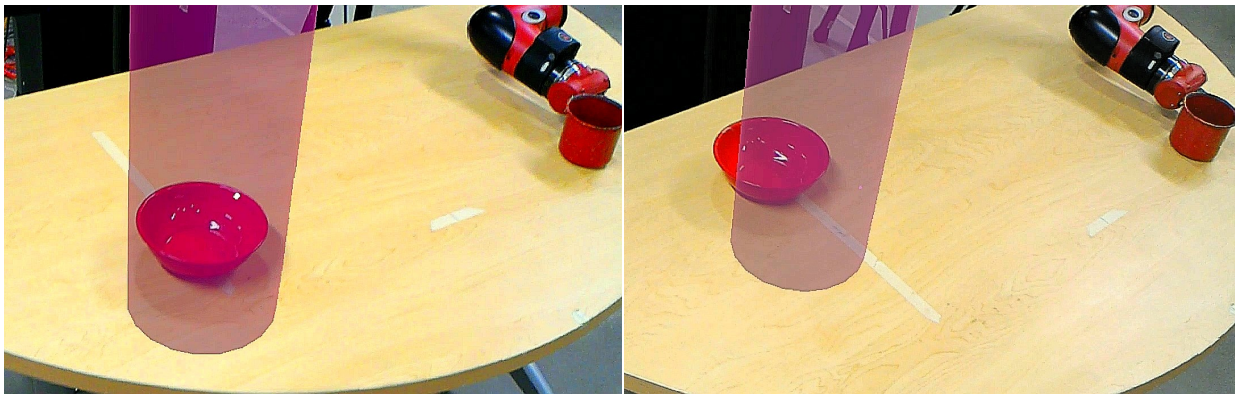


Figure 2.8: Case Study III involves the robot pouring a cup into a bowl positioned at different points on the table. In the initial setup (left), the bowl is placed toward the front of the table, while in the test condition (right), the bowl is placed further back. In both cases, the user applies an over-under constraint to the trajectory representation in order to ensure the pouring motion takes place at the correct position.

used in lieu of a simulated collision obstacle required by motion planning. Images from this case study are given in Figure 2.7.

2.2.5.3 Case Study III (Changing Goal)

The third and final case study we conducted involved a task in which the robot poured a cup of material into a receptacle. The modification for this case study consisted of moving the receptacle to a different position. Using ARC-LfD’s over-under constraint, the user was able to specify where on the table the pouring part of the task should begin. This allowed the robot to execute the cup pouring task successfully with two different end goal positions without any new demonstrations. Figure 2.8 illustrates the environment setup and constraint applications for this case study.

2.2.5.4 Discussion

These three case studies exhibit the functionality of ARC-LfD and its ability to make LfD systems more robust. Case Studies I and III illustrate that ARC-LfD can make a set of demonstrations robust to changes in the task, provided sufficient variance of demonstrations in the set: through application of constraints to an existing skill, the robot can execute an altered version of a task. Case Study II shows how ARC-LfD can make learned skills robust to changes in the environment through using constraints that alter the skill trajectory to fit a new execution context. Furthermore, the interface of ARC-LfD enables users to conduct these alterations after demonstrations have been given, allowing for any-time editing of a skill. In addition to its functionality for verifying and previewing skills directly in the environment, ARC-LfD introduces a method for maintaining robotic skills even if the particulars of task and environment shift over time. We posit that ARC-LfD presents a safer-by-construction alternative to general end-to-end policy learning systems, trading generally unneeded levels of model expressivity for system transparency, enabling successful safer skill execution across a broad range of robotics tasks.

2.2.6 Conclusion

ARC-LfD is proposed as a step toward producing practical, real-world-ready LfD systems that allow non-roboticists to conduct training and evaluation of robotic systems. The use of AR

for in-situ visualizations relaxes the requirement of a model of the environment to use in simulation for verification of learned skills. Through visualizing a sample trajectory directly in the environment, users can preview the robot’s skill execution contextualized by the actual environment itself. The control flow of ARC-LfD provides an improvement over CC-LfD, allowing users to separate demonstration from constraint application.

Finally, the proposed constraint editing interface relaxes the static environment assumption often levied for successful LfD skill deployment. ARC-LfD enables direct skill repair and editing, creating constraints contextualized in the environment and applying them to keyframes of an existing skill. Thus, ARC-LfD fills a critical technical gap in LfD systems, enabling long-term skill assessment and validation as the environment or task requirements change over time.

Chapter 3

Augmented Reality for Reducing Uncertainty in Human Motion in Shared-Space Interaction

3.1 Motivation

This chapter covers a pair of novel techniques, both of which use augmented reality visualizations to influence and constrain the movements of human teammates, thus reducing the inherent uncertainty of human motion, and allowing for more tractable human modeling within human-aware robot planning frameworks. In particular, these works focus on domains with close-proximity human-robot collaboration: improvements in human predictability in such domains not only has positive impacts on task fluency, but also safety through the avoidance of space conflicts and collisions.

The first work (Chapter 3.2) introduces a method for improving the prediction of human trajectories in close-proximity human-robot collaboration through both the arrangement of physical objects in the shared workspace prior to interaction and the projection of “virtual obstacles” in augmented reality. This combined technique alters the workspace to optimize for the legibility of human motions from the perspective of a robot teammate, reducing uncertainty in human trajectories, improving human goal prediction, and leading to more fluent interaction. The technique is evaluated in a collaborative pick-and-place domain with a manipulator robot, where the robot makes real-time predictions of human goals using a time series multivariate Gaussian prediction model. Through the use of our workspace organization and augmented reality projection technique, the performance of such prediction models are improved, while simultaneously requiring less train-

ing data. This work was presented at the ACM/IEEE International Conference on Human-Robot Interaction (**HRI 2024**) [258], where it received a nomination for best technical paper.

The second work (Chapter 3.3) introduces FENCES, a system which dynamically divides floor space in warehouse-style environments between human and robot owners, projecting those regions through an AR headset. The regions of ownership change in response to requests made by human and robotic agents, facilitating space negotiation. The AR-visualized regions are designed to simultaneously inform human teammates where robots are likely to move and work, while also restricting human movement to pre-approved areas, mitigating collisions on a busy shared floor. We explore the psychological impacts of the FENCES interface through a human-subjects study involving humans and aerial drones working in close proximity in a warehouse environment. Through this study, we find that the FENCES interface generates an overly inflated sense of safety by human teammates around potentially dangerous aerial robots. This often leads to willful violations of allotted space for increased convenience, as humans trust the robots to accommodate them. From experimental results and post-experiment interview responses, we derive a number of design recommendations for shared space negotiation systems. This work was presented at the IEEE International Conference on Robotics and Automation (**ICRA 2023**) [40].

Though the FENCES system contains an element of explicit human-to-robot communication (humans are able to directly indicate their goal, influencing the space in the shared environment allotted to them), in both works, information primarily flows from robot teammates to human teammates. Both the workspace optimization system and FENCES are responsible for making the decisions regarding what visualizations are shown to the human, while providing little to no rationale about the decisions it has made. This opacity in robot-expert systems has a tendency to lead to confusion and frustration in human teammates, as evidenced by user reactions in the FENCES study [40]. Many participants advocated for the inclusion of explicit visualizations indicating both the intentions and decision-making rationale of robot teammates. This concept is explored extensively in Chapters 4 and 5 of this thesis.

3.2 Workspace Optimization Techniques to Improve Prediction of Human Motion During Human-Robot Collaboration

3.2.1 Introduction

In human-robot collaborative tasks, shared mental models between agents enable the awareness and joint understanding required for effective teamwork [249]. With no shared notion of the task to be completed, the inherent stochasticity and opacity of human decision-making makes robot planning difficult [229]. To this end, prior research efforts have focused on developing robots that can predict human behavior [211, 147, 166], generating motion plans to safely interact in a shared environment [196, 106]. However, these methods are limited by the quality of robot predictions of a human collaborator’s intention and resultant behavior. With inaccurate human models or unexpected human behavior diverging from past experiences, the robot may produce unsafe interactions [168], especially in safety-critical or close-proximity settings [261].

To address the inherent challenges of accurately predicting human motion early in a demonstrated trajectory, our key insight is that robots can take an active role in structuring the environment to reduce the variance of human motion caused by dense and overlapping task spaces, thereby improving the performance of human behavior models. In this work (see Fig. 3.1), we introduce an algorithmic approach for a robot to configure a shared human-robot workspace prior to interaction in order to improve a robot’s ability to predict the human collaborator’s goals during task execution. As detailed in Fig. 3.2, we present an objective function that scores potential workspace configurations in terms of how legible the actions of a human teammate are likely to be when performing a task in that environment. We use the mathematical formulation of legibility from [70], which computes the probability of successfully predicting an agent’s goal given an observation of a snippet of its trajectory. Our approach finds workspace configurations that maximize legibility over the valid goals at each stage of task execution.

Each candidate workspace configuration combines a potential arrangement of physical objects and projection of “virtual obstacles” in augmented reality (AR) in the environment. While the

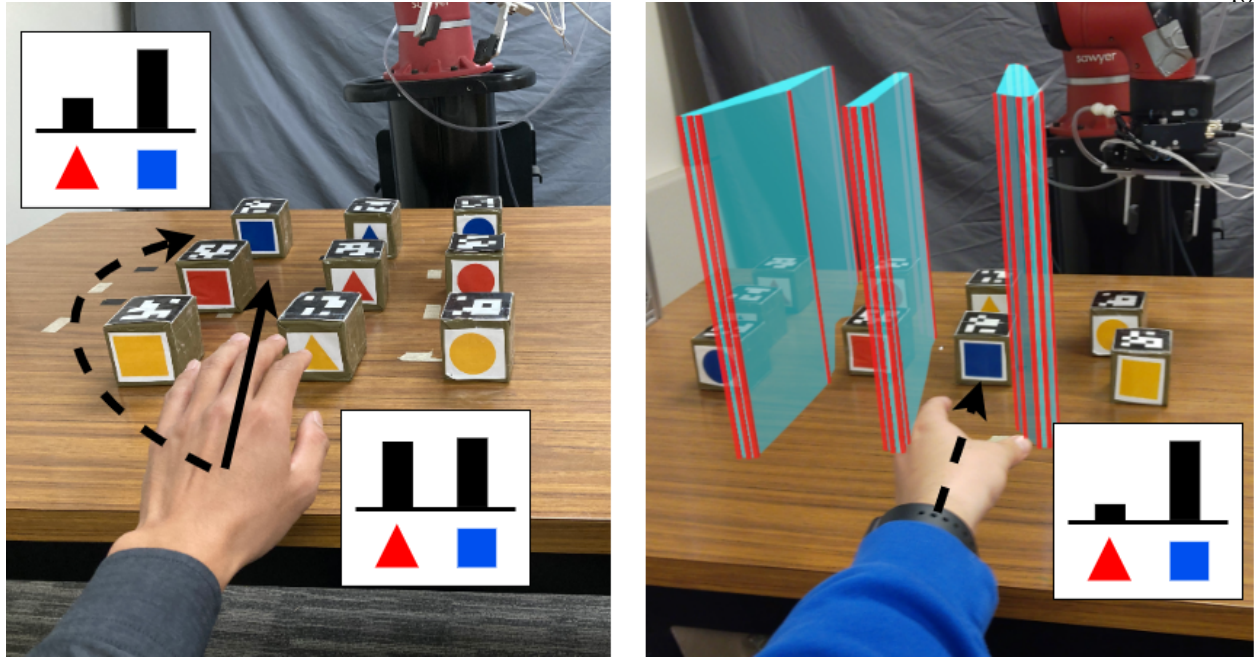


Figure 3.1: Workspace configuration affects the robot’s ability to correctly predict the human’s goal – the blue square cube. **Left:** The legible path (dotted) requires the human to take a circuitous route while the natural path (solid) is not legible. **Right:** Our approach generates a workspace configuration by arranging physical objects and projecting “virtual obstacles” in AR (cyan and red barriers), in order to induce naturally legible paths from the human.

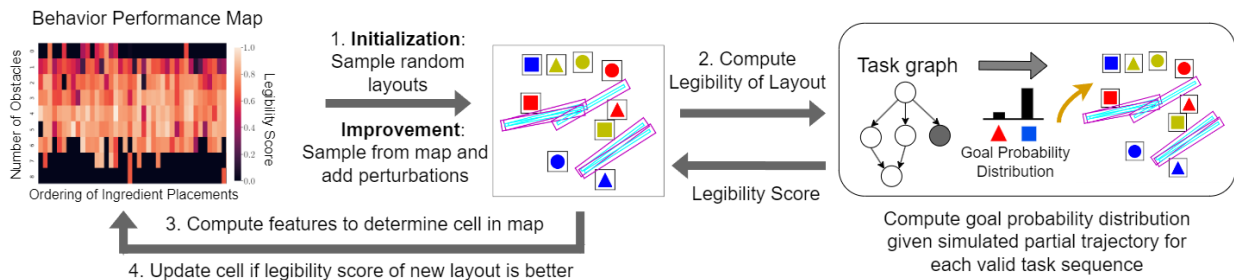


Figure 3.2: Our approach for generating workspace configurations that enable accurate human goal predictions. **(1)** In the initialization phase, we sample random environment layouts to populate the behavior performance map, which stores diverse and high performing solutions. This is followed by the improvement phase where we sample directly from the map and add perturbations to test whether the legibility is improved. **(2)** In both phases, we compute the legibility of the sampled layout by computing the probability of predicting the correct goal at each stage of the task execution. **(3)** We compute the features of the sampled layout to determine its location in the map. **(4)** The map is updated if the legibility score of the sampled layout is better than the existing one.

arrangement of physical objects can be achieved prior to the shared task via a simple composition of robotic pick and place actions, the addition of augmented reality-based virtual obstacles is particularly effective in imposing explicit constraints on the possible motions of the human, without requiring additional physical changes to the environment. Note that our approach does not restrict the human to a particular goal and retains their ability to decide on the goal they’re reaching towards. Humans also have the flexibility to alter their decisions midway through a task, since goal prediction is performed at every time step.

We efficiently explore the space of workspace configurations using a quality diversity (QD) algorithm called Multi-dimensional Archive of Phenotypic Elites, or MAP-Elites [179]. Instead of finding a single optimal solution, MAP-Elites produces a map of performant solutions along dimensions of a feature space chosen by the designer. MAP-Elites enables efficient and extensive exploration of complex search spaces, leading to higher quality solutions as compared to other search algorithms [57, 187].

We empirically demonstrate that our workspace optimization approach improves the accuracy of human goal prediction via time series multivariate Gaussian model. We collect human motion data in a tabletop experiment involving a robotic manipulator, comparing our approach to alternative workspace arrangement strategies. In summary, we present two primary contributions: 1) an algorithm for optimizing the placement and arrangement of physical and virtual objects in a shared human-robot workspace for maximizing human legibility, and 2) an evaluation of that algorithm via human subjects experiment, showing that our technique influences human behavior in ways that improve a robot collaborator’s prediction model.

3.2.2 Related Work

3.2.2.1 Planning using Human Motion Models

In human-robot collaboration, the robot needs to predict human motion in order to coordinate its actions with those of the human. Prior work has developed motion planning algorithms that

models human motion and generates robot plans to safely interact with humans in tabletop [146] and navigation [84, 197] settings. To account for the uncertainty in human motion prediction, prior work has used partially observable Markov decision processes (POMDPs) to determine optimal actions for a robot given a probabilistic belief over the human’s intended goals [196, 106]. These methods rely on the human motion model to achieve safe interactions, but human motion is inherently highly variable as humans can always move unexpectedly. The difficulty of the problem can be seen by the variety of approaches taken toward accurate human motion prediction: Gaussian models [153, 201], dynamic movement primitives [162, 274], latent representation learning [27, 170], and imitation learning [166, 75], among various other methods [211]. Our work takes a different approach and addresses a fundamental challenge faced by all human motion prediction models; we reduce the uncertainty inherent in modeling the intentions of human collaborators by pushing them towards legible behavior via environment design. Our work improves human motion model predictions by increasing environmental structure to reduce uncertainties, facilitating more fluent human-robot interactions.

3.2.2.2 Environment Design in Robotics

Prior work has also explored designing or modifying environments in order to better achieve agent goals. Zhang et al. [286] proposed a framework for designing environments that optimize an agent’s reward, and Keren et al. [133] extended it for stochastic transitions. Kulkarni et al. [139] generates interpretable *robot* behaviors by modifying the environment. These works show the potential advantages of a robot using environment design to improve its task performance and interpretability. There has also been work on modifying the environment for collaborative teaming. Zhang et al. [287] optimizes warehouse layouts for multi-robot coordination, and Bansal et al. [16] explores robots moving objects, which are termed supportive actions, to reduce the likelihood of future collisions in a tabletop task. Our technique differs from prior work in that the robot modifies its environment with the explicit goal of improving its ability to predict human behavior for fluent collaboration with human teammates.

Finding the optimal environment by iterating through all possible workspace configurations quickly becomes intractable as the number of objects or possible states increases. Others have used quality diversity (QD) algorithms to generate diverse environments for evaluating the performance of shared autonomy algorithms [82] and explore diverse coordination behaviors as a result of environment design [81]. Inspired by the success of QD algorithms in finding diverse solutions in large search spaces [57, 187], we use MAP-Elites to search for environments that best elicit legible human behavior. Our work extends QD approaches to generate interaction scenarios that influence human behavior and address the robot’s limitations when collaborating with humans.

3.2.3 Legible Workspace Generation

In this section, we describe our approach for modifying the shared human-robot workspace to maximize legibility and enable more accurate human goal predictions (summarized in Fig. 3.2). We first present the legibility score (Section 3.2.3.1) which evaluates a workspace configuration based on the probability of predicting a given ground truth goal when provided a partial trajectory. As a task progresses, the set of goals that the human might reach for changes. Therefore, the objective function (Section 3.2.3.2) considers the possible subtask sequences and computes the legibility score for the valid goals at each stage of the task. Enumerating all possible workspace configurations and evaluating the objective function for each is not scalable; thus we use MAP-Elites (Section 3.2.3.3) to explore the solution space and approximate the best solution.

3.2.3.1 Legibility Score

To evaluate the legibility of a workspace configuration, we consider the probability distribution of predicting that the human is approaching goal G given an observed trajectory from start state S to intermediate point Q . We use the formulation developed by [70] shown in Equation 3.1.

$$\Pr(G|\xi_{S \rightarrow Q}) \propto \frac{\exp(-C(\xi_{S \rightarrow Q}) - C(\xi_{Q \rightarrow G}^*))}{\exp(-C(\xi_{S \rightarrow G}^*))} \quad (3.1)$$

The optimal human trajectory from point X to point Y with respect to cost function C is denoted by $\xi_{X \rightarrow Y}^*$. Equation 3.1 evaluates how cost efficient (with respect to C) going to goal

G is from start state S given the observed partial trajectory $\xi_{S \rightarrow Q}$ relative to the most efficient trajectory $\xi_{S \rightarrow G}^*$.

Let \mathcal{G} be the set of valid goals at the current time step. We develop a legibility score (Eqn. 3.2) for use in our optimization objective that, for every valid goal at a given time step in the task execution, maximizes the margin of prediction between the human’s chosen goal $G_{true} \in \mathcal{G}$ and all other valid goals. If the most likely goal is not G_{true} , the score is penalized by a fixed cost c multiplied by the length of the sampled human trajectory $|\xi_{S \rightarrow Q}|$. Otherwise, the score is the difference of the two highest probabilities (computed by the *margin* function). The notation $G_{(i)}$ denotes the i -th index of a sorted list of length n that represents the goal probabilities ordered from smallest likelihood to largest given the observed trajectory $\xi_{S \rightarrow Q}$.

$$\text{EnvLegibility}(G_{true}) = \begin{cases} -c|\xi_{S \rightarrow Q}|, & \text{if } \arg \max_{G \in \mathcal{G}} \Pr(G|\xi_{S \rightarrow Q}) \neq G_{true} \\ \text{margin}(\mathcal{G}|\xi_{S \rightarrow Q}) = G_{(n)} - G_{(n-1)}, & \text{otherwise} \end{cases} \quad (3.2)$$

3.2.3.2 Optimization for Task Legibility

To generate a workspace configuration with improved legibility of the agent’s goals for a task, we maximize the legibility score from Equation 3.2 for all valid subtask sequences (Eqn. 3.3). We use precedence constraints introduced by the structure of the task (i.e., which subtasks are prerequisites for other subtasks) to identify the set of valid subtasks (and thus valid goals \mathcal{G}) at any given time step.

$$\max_{T' \in \text{permutations}(T)} \sum \mathbb{1}\{\text{valid}(T')\} \times \sum_{t \in T'} \sum_{G \in \mathcal{G}} \text{EnvLegibility}(G) \quad (3.3)$$

Let T' represent a valid subtask sequence, consisting of all k subtasks of task T : t_1, \dots, t_k , ordered such that for each index i , subtask t_i has all precedence constraints satisfied. Each subtask has one or more goals that an agent can reach to complete it. For example, the task “Set Table” may have a subtask “Get/Place plate on place mat” with multiple satisfying goals (multiple place mats). Additionally, since there are often multiple valid subtask sequences given an observed set of subtasks t_1, \dots, t_i , \mathcal{G} represents the set of goals corresponding to all uncompleted subtasks with

satisfied precedence constraints. For example, after a plate is set, “Get/Place Fork” and “Get/Place Spoon” may be equally valid as the next subtask to perform. At this point, \mathcal{G} includes potential goals for both fork and spoon placement subtasks. The objective function considers all possible goals that the human might be reaching for at a given stage of task execution and maximizes the probability of correctly predicting the human’s chosen goal.

Algorithm 1: Workspace Generation with MAP-Elites

Input: Human Trajectory Generator G_H , Objective function F , measure function M

Initialize: Solution map $S \leftarrow \emptyset$, Solution values $V \leftarrow \emptyset$

```

1 for  $i = 1, \dots, N$  do
2   if  $i < N_{init}$  then
3     | Generate workspace  $w = \text{random\_workspace}()$ 
4   else
5     | Sample workspace from map  $w = \text{random}(S)$ 
6     | Run  $w = \text{improve\_workspace}(w)$ 
7     Determine features  $\mathbf{m} = M(w)$ 
8     Determine objective score  $s = F(w)$ 
9     if  $S[\mathbf{m}] = \emptyset$  or  $s > V[\mathbf{m}]$  then
10    | Update solution map  $S[\mathbf{m}] = w$ 
11    | Update solution values  $V[\mathbf{m}] = s$ 
12 return  $S, V$ 

```

3.2.3.3 Search using Quality Diversity

Iterating through all possible workspace configurations to find the optimal solution is intractable since the number of possible configurations is exponential in the number of goals, virtual obstacles, and size of the workspace. We use MAP-Elites [179] to approximate the optimal solution.

In Algorithm 1, MAP-Elites maintains a behavior performance map, or solution map S , that stores high performing solutions across features or behaviors of interest. To find the most legible workspace (objective function F), the designer chooses a set of features or behaviors (computed by measure function M) such as the distance between the objects or the number of virtual obstacles. The algorithm would then find the most legible workspace for each possible combination of features found. As input, the algorithm also requires a model G_H that outputs human trajectory given a goal. G_H can be learned from data via inverse optimal control [166] or approximated via shortest

Algorithm 2: improve_workspace

Input: Workspace configuration w , objective score of w s_w , objective function F , measure function M

Initialize: Best configuration $w^* \leftarrow w$, Best score $s^* = s_w$

```
1  $A = \text{available\_perturbations}(w)$ 
2 for  $a \in A$  do
3   |   Generate new workspace  $w' = \text{apply\_perturbation}(w, a)$ 
4   |   Compute legibility score  $s_{w'} = F(w')$ 
5   |   if  $s_{w'} > s^*$  then
6   |     |   Update best workspace  $w^* = w'$ 
7   |     |   Update best score  $s^* = s_w$ 
8 if  $w^* \neq w$  then
9   |   return improve_workspace( $w^*$ )
10 return  $w^*$ 
```

path to goal [70].

MAP-Elites consists of two phases: initialization and improvement. In the initialization phase, we randomly sample workspaces for N_{init} iterations (Line 3) and store them in the corresponding cell in the solution map by computing the features (Lines 7-11). In the solution map S , the cell associated with the vector of feature values m is denoted $S[m]$. For $N - N_{init}$ iterations, we perform the improvement phase, following differentiable quality diversity [80] to use gradient information to speed up search. We first randomly sample from the solution map (Line 5) and then empirically approximate the gradient (Line 6) to improve the solution. As detailed in Algorithm 2 (the *improve_workspace* function), Line 1 retrieves all available perturbations to the workspace (i.e. changing an item’s position, adding or removing a virtual obstacle). For each perturbation, a new workspace configuration w' is generated by applying the perturbation. We keep track of the current best workspace w^* in terms of the objective score (Lines 5-7). If there was an improvement to the workspace, we run Algorithm 2 again (Lines 8-9). Otherwise, a local minima has been found, and we return the best workspace found (Line 10).

3.2.4 Evaluation

We evaluate our approach in a tabletop collaborative pick and place task with a Sawyer robotic arm. To predict human goals given partial arm trajectories, we implement the time series multivariate Gaussian model proposed in [201] to determine which object the human is reaching for.

3.2.4.1 Hypotheses

H1: The legibility-maximized environment generated by our approach will enable more accurate predictions of the human goal throughout task execution, compared to alternative environment setups.

H2: The time series multivariate Gaussian prediction model will have better accuracy than predictions based on heuristics, such as predicting the goal that is closest to the current trajectory.

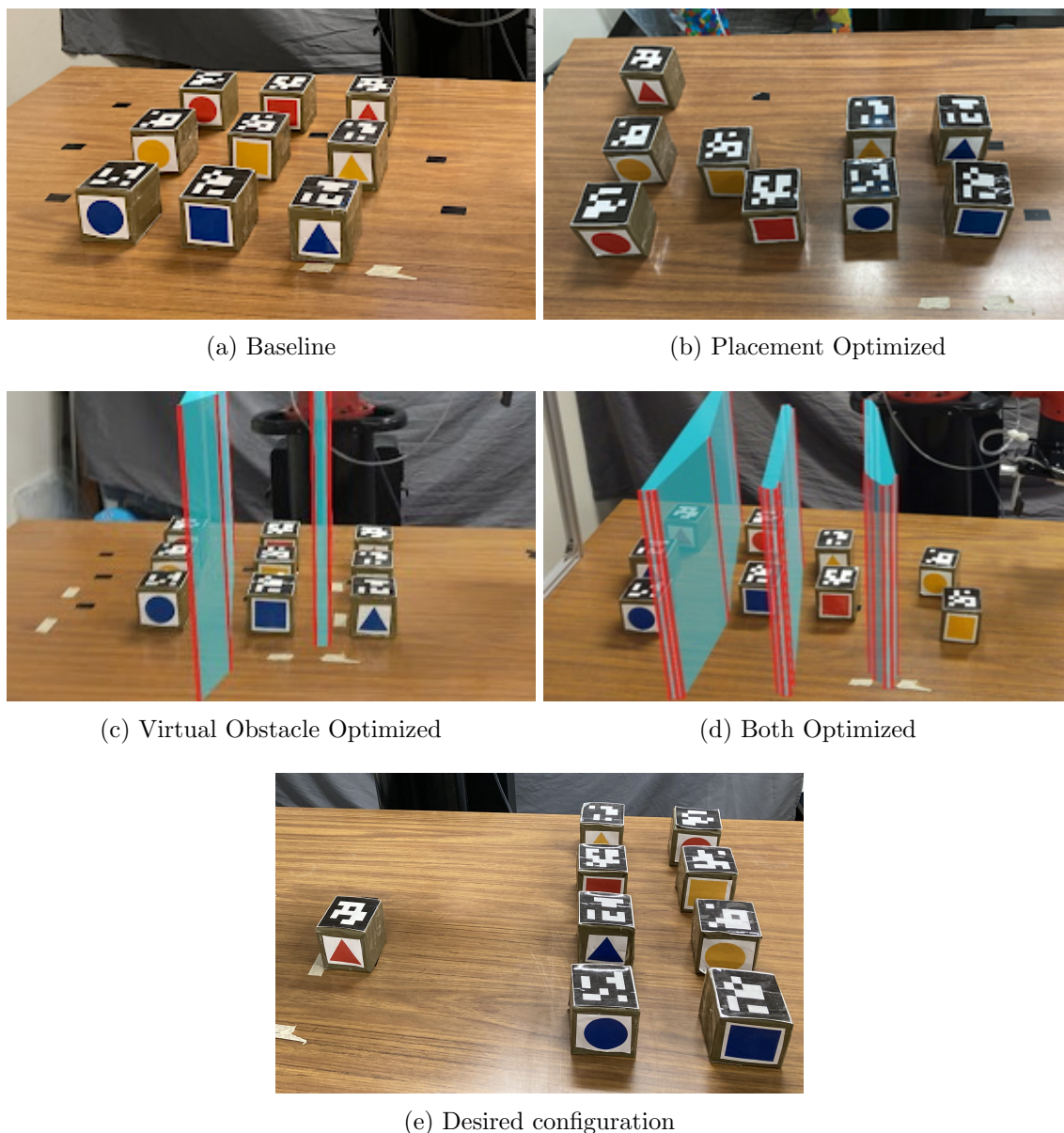


Figure 3.3: (a-d) initial cube configurations used in our tabletop experiment. The environment generated by our approach, (d) *Both Optimized*, optimizes object and virtual obstacle (shown in cyan with red edges) placements to elicit legible human motion. (e) The tabletop experiment involves the human and the robot collaboratively placing cubes into the desired configuration: two columns on the right with a given ordering.

H3: Prediction models trained in environments generated by our approach will be more data-efficient, requiring fewer examples to reach peak performance levels as compared to those trained in alternate environment setups.

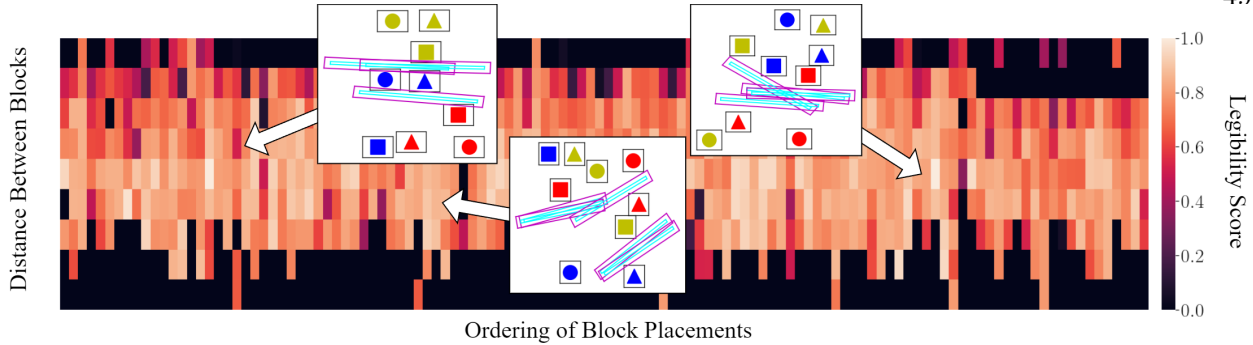


Figure 3.4: MAP-Elites is able to explore legible workspace configurations across combinations of features: in this case, the minimum distance between blocks and the ordering of block placements along the x-axis. This improves the best solution found in complex search spaces.

3.2.4.2 Tabletop Experiment

Experimental Setup: We conducted an IRB-approved human subjects study with 12 participants (8 male, 4 female), with an age range of 19 to 31 years old ($M = 24.42$, $SD = 3.37$), recruited from a university campus to collect human hand trajectory data in different tabletop workspace setups. We collected 8 trajectories per participant for each of the four conditions, presented in a randomized, counterbalanced order. The study involves the human and an autonomously operating Sawyer 7DoF manipulator robot working collaboratively to place cubes into a desired configuration (Fig. 3.3e). To prevent the robot from picking up the same cube, the participant is asked to first reach for a cube while the robot maintains a probability distribution over the possible cubes the human is reaching for in real time. Once the robot is sufficiently confident of the human’s goal, the robot will select its own cube to pick up and move to grasp it. The human and robot team continues to pick a cube each until the task is completed. The precedence constraints are set such that the first column of the desired configuration must be completed before the second column can start. This experiment evaluates homogeneous teams where the human and robot are able to perform all subtasks. Our approach generalizes to heterogeneous teams, in which case we modify T from Equation 3.3 to only include subtasks the human can perform.

Goal Prediction Model: We use a time series multivariate Gaussian model [201] to predict human goals given the human hand trajectory collected from the tabletop experiment. For a

training data set D , we use dynamic time warping (DTW) [181] to align the trajectories so they all have length K . For each time step $k \in K$ and goal G , we train a multivariate Gaussian with mean $\mu_G[k] = \frac{1}{|D|} \sum_{i=1}^{|D|} f_i[k]$ and covariance $\Sigma_G[k] = \frac{1}{|D|-1} \sum_{i=1}^{|D|} (f_i[k] - \mu[k])(f_i[k] - \mu[k])^T$ where $f_i[k]$ are the feature values of the i th trajectory at time step k . At prediction time, the probability of a goal G given a partial trajectory ξ is $P(G|\xi) \propto \prod_{k=1}^K [\mathcal{N}(\mu_G[k], \Sigma_G[k])]^{\frac{1}{K}}$ [201]. The goal with the highest probability is the predicted goal. We use two features, the x and y positions of the hand, captured via an Intel RealSense RGB camera and a real-time hand tracking algorithm [285].

Conditions: We compare workspace setups using our optimization against **Baseline**, where the cubes are initially sorted by their color and shape (Fig. 3.3a). We also perform an ablation study, removing the algorithm’s ability to organize the cubes or project virtual obstacles. As such, the remaining conditions are:

Placement Optimized: We optimize for legibility by only rearranging the cubes, with no virtual obstacles (Fig. 3.3b).

Virtual Obstacle Optimized: The cubes are sorted by their color and shape, and we optimize for legibility by projecting virtual obstacles (Fig. 3.3c).

Both Optimized: Our approach that optimizes for legibility by rearranging the cubes and projecting virtual obstacles (Fig. 3.3d).

Workspace Generation: The feature dimensions for MAP-Elites are the minimum distance between the cubes and the ordering of cubes by the x-axis. We do not add virtual obstacles in the initialization phase, but rather in the improvement phase. Due to the continuous state space and the difficulty of randomly sampling useful virtual obstacles, we choose fixed size virtual obstacles and insert them between two randomly selected cubes during the improvement phase. By employing this heuristic, we can effectively explore configurations that result in altered human reaching motions, which are approximated using shortest paths in a visibility graph. We experimented with virtual obstacles of different sizes and found that obstacles measuring 30cm x 1cm were sufficient in inducing distinct reaching motions.

1) Initialization. We randomly sample (x, y) positions for the cubes and ensure that

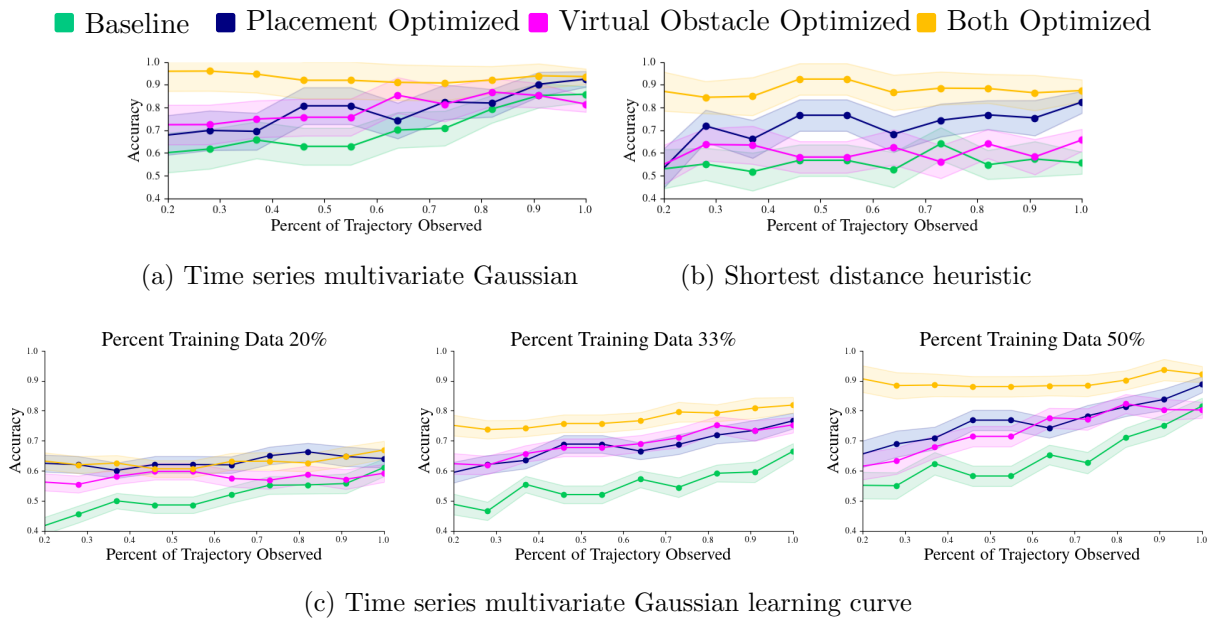


Figure 3.5: Tabletop experiment results: (a) Mean accuracy of time series multivariate Gaussian model. Our approach, *Both Optimized*, elicits significantly higher or comparable accuracy compared to the baselines. (b) Mean accuracy using shortest distance heuristic where the closest cube to the current hand position is the predicted goal. (c) Mean accuracy of time series multivariate Gaussian model with varying amounts of training data. *Both Optimized* elicits significantly higher accuracy compared to baselines starting at 33% training data, achieving around 90% accuracy with just 50% of the available training data.

they don't overlap. The positions are continuous values uniformly sampled within the permissible boundaries.

2) **Improvement.** The improvement step consists of first changing cube positions and then adding virtual obstacles. We sample new cube positions from a Gaussian centered at the cube with variance = 7cm. The virtual obstacles are placed between two randomly sampled cubes.

The boundaries of the algorithmically generated virtual obstacles are passed to an AR interface, implemented using a Microsoft HoloLens 2 head-mounted display. The AR interface renders those obstacles directly in the environmental context of the shared workspace as holograms of cyan barriers with red outlines (Fig. 3.3d). These barriers appear to the human as if they are physically located in the environment, and indicate regions of the environment the human should not enter.

3.2.5 Results

We ran a stratified 4-fold cross validation with 3 repeats and different randomization in each repetition. The stratified cross validation preserves the percentage of samples for each class (i.e., the valid goals at each time step) in each fold. Figure 3.5 shows the mean goal prediction accuracy as a function of the percentage of total trajectory observed. A prediction is defined as correct if the single highest probability value in the probability distribution is the same as the true target goal. A trajectory is defined as the human hand positions in the x - y plane when reaching towards a goal. We use the one-way analysis for variances (ANOVA) and perform post-hoc analysis using Tukey's HSD test for multiple comparisons to test for effects between condition pairs. We plot Tukey's Q critical value as the width of the shaded area in Figure 3.5 such that overlaps indicate an absence of a statistically significant difference.

Figure 3.5a shows the mean goal prediction accuracy when different percentages of the total trajectory is observed using the Gaussian model. The environment generated by our approach, *Both Optimized*, elicits significantly higher prediction accuracy than baseline environments when less than 50% of the trajectory has been observed. When using the shortest distance heuristic (Fig. 3.5b), where the closest cube to the current hand position is the predicted goal, all conditions elicit

lower accuracy compared to the predictions when using the Gaussian models. The shortest distance heuristic exhibits poor performance even when the entire trajectory is observed. Both hypotheses $H1$ and $H2$ are supported by these results.

To evaluate the learning curve of the Gaussian model, we perform stratified splits cross validation with 10 repeats. Figure 3.5c shows the performance as training data increases. *Both Optimized* elicits significantly higher accuracy than the baselines with 33% training data and achieves around 90% accuracy with 50% training data, supporting our hypothesis $H3$ that our approach generates environments where the training of prediction models is more data-efficient.

In Figure 3.6, we plot the mean and covariance of the learned time series multivariate Gaussian for each condition. Qualitatively, the Gaussian models in the *Both Optimized* condition have less covariance, a measure of uncertainty, compared to the Gaussian models trained in baseline environment configurations. Table 3.1 shows the average determinant and trace of the covariance matrices of the Gaussian models. The determinant is a measure of the magnitude of variation among the variables, and the trace is the sum of the variances of the individual variables but does not consider the correlations between the variables. The Gaussian models trained in the *Both Optimized* layout have lower values for both the determinant and trace compared to the models trained in baseline environments.

Table 3.1: The average determinant (det) and tract of the covariance matrices of the multivariate Gaussian models.

	Baseline	Placement Opt.	Virtual Obstacle Opt.	Both Optimized
Det	9.36e-06	8.50e-06	6.14e-06	1.09e-06
Trace	7.12e-03	5.90e-03	3.90e-03	2.77e-03

Overall, our results show that our workspace configuration approach reduces the uncertainty in human motion data and improves the accuracy of goal prediction models in tabletop shared-space tasks. The models resulting from our approach are also more data-efficient, requiring less

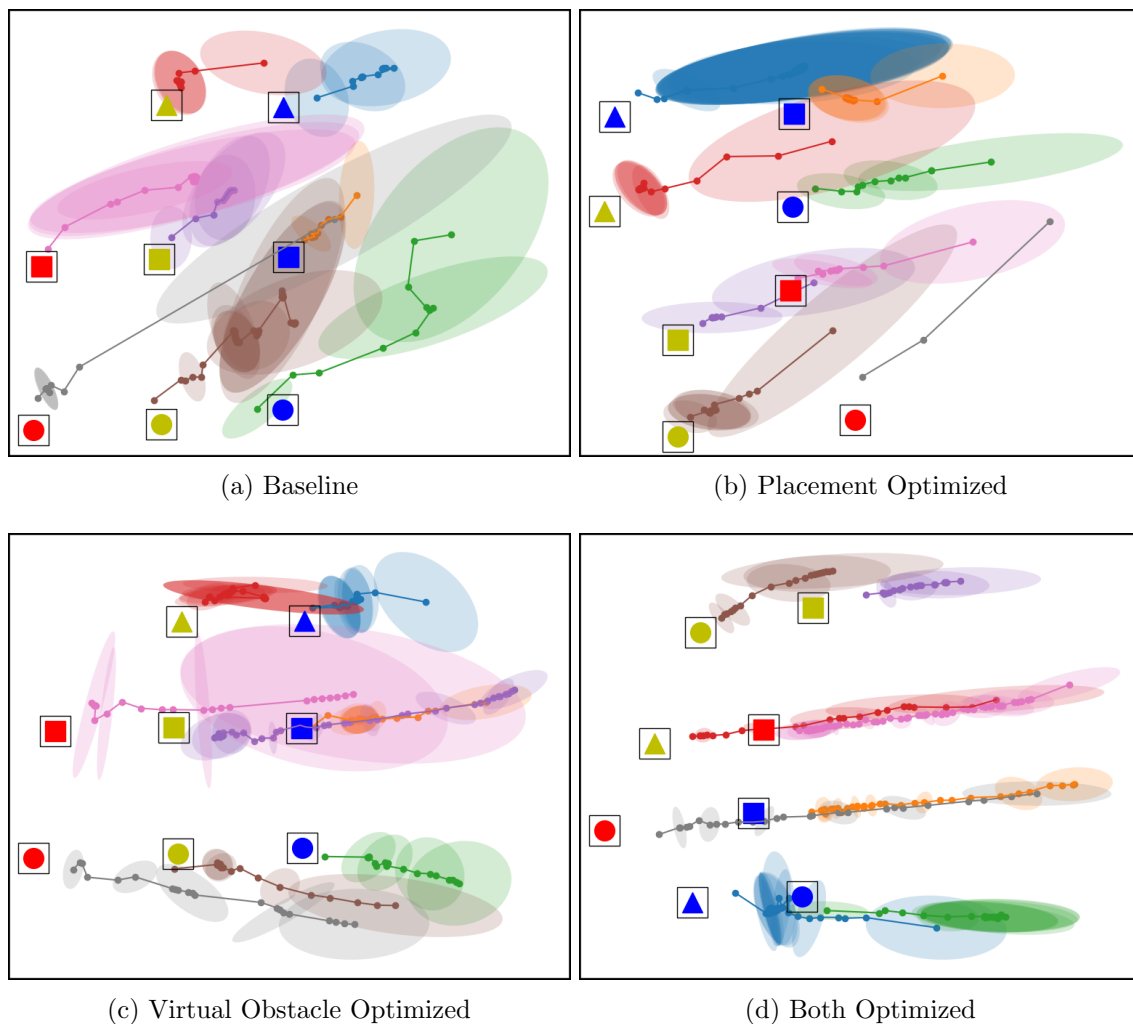


Figure 3.6: Top down view of the workspace plotting the mean and covariance of the time series multivariate Gaussian for each condition. The model in the *Both Optimized* condition has less covariance compared to the models trained in the other environment configurations.

data to reach their best performance levels as compared to those trained in baseline environments.

3.2.6 Conclusion

In this work, we introduce an algorithmic approach for autonomous workspace optimization to improve robot predictions of a human collaborator’s goals. By rearranging physical objects in the workspace and projecting AR-based virtual obstacles into the environment prior to interaction, the robot influences the human into more legible behavior during task execution, thereby

reducing the uncertainty of their motion. Through a human-subjects experiment, we show that our approach results in more accurate model predictions, while requiring less data to achieve these correct predictions. Importantly, we demonstrate that environmental adaptations can be discovered and leveraged to compensate for shortfalls of prediction models in otherwise unstructured settings. Through our results, we showcase the potential of workspace optimization for realizing fluency improvements in human-robot collaborative domains.

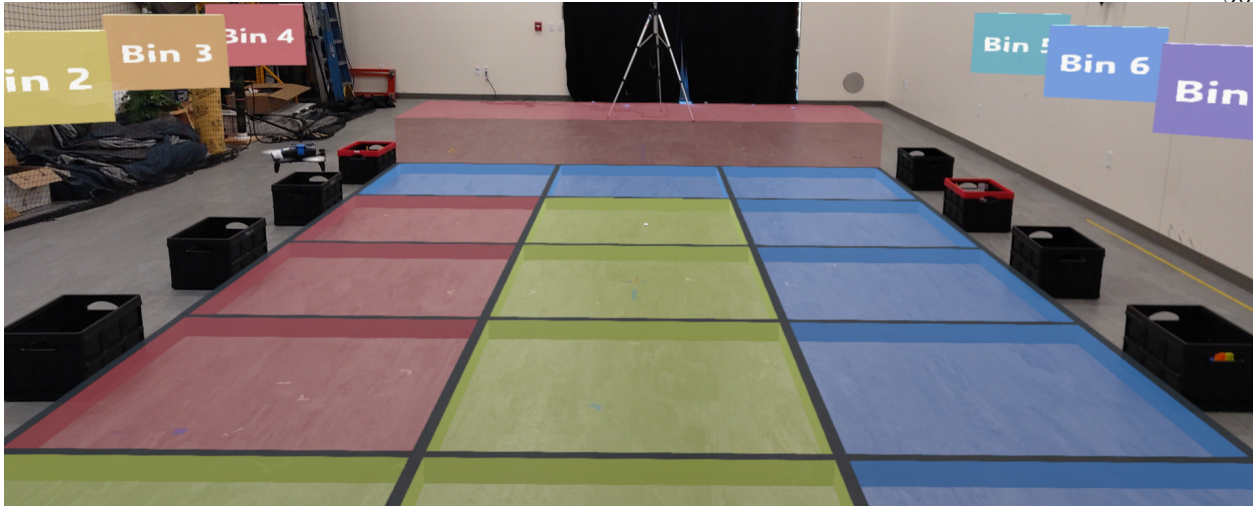
3.3 Human Non-Compliance with Robot Spatial Ownership Communicated via Augmented Reality: Implications for Human-Robot Teaming Safety

3.3.1 Introduction

Due to a confluence of technological availability and utility, humans and robots are increasingly operating in close proximity to each other. The current state of safety in human-robot collaborative and cooperative collocated tasks generally revolves around protecting the human from any contact with the robot, using physical barriers and sensors to pause robot operation in the vicinity of humans. This is often realized as robots installed within physical cages or within fences in a manufacturing environment, or as ground robots in a well-structured warehouse environment that stop when a human approaches wearing specially instrumented clothing. While effective at preventing negative interactions, these approaches tend to be inefficient and cause frustration.

Research on increasing predictability and interpretability of quadcopter robots by collocated humans [244, 267, 243, 238], in addition to work that predicts human movement [77], tends to assume that a robot should always defer or conform to human preferences independent of the rationale behind them. However, the practical alternative of expecting the human to conform to the robot's movements or demands is less explored. With increased deployment of robots in established processes within warehouse, manufacturing, and even space environments, we must find safe, efficient, and robust ways of collaborating with them.

This work surfaces insights about human compliance and non-compliance with robot instruc-



(a) View through the HoloLens from Home. Eight bins contain task components, a 3x5 grid indicates spatial ownership, and bin labels are selected to request access. Shown is a path to Bin 4.



(b) Looking towards Home in the ARHMD during the trap scenario (no return path) in the Shared Space condition.

Figure 3.7: Example views through the HoloLens ARHMD.

tions for spatial ownership as delivered via augmented reality in a collocated environment with important safety implications. These insights are gathered from an experiment where human and robotic agents held ownership over different areas of a warehouse floor. We designed and implemented the FENCES (Facilitation of Efficient Nonverbal Collocated Environment Safety) System

to enable this interaction. FENCES enables a user to request permission from an autonomous robot to traverse the work floor to reach bins containing parts needed for an assembly task. The robot, an autonomous free-flying quadcopter that is conducting an inventory task, gives permission by giving the human temporary ownership of parts of the floor indicated by hologram coloration (see Figure 3.1a).

We investigated user behavior and compliance with respect to the FENCES system through an Institutional Review Board-approved, between-subjects study with two conditions: (1) a shared-space condition where the human and robot occupied the floor concurrently, and (2) a turn-taking condition where the human and robot performed their tasks sequentially, with only one of them allowed on the work floor at a time. The main contributions of this work are our findings surrounding human compliance and the justifications they provide for non-compliance and the subsequent identification of critical design considerations for future AR-based safety systems to incorporate, with implications for safety, trust, and cognitive load.

3.3.2 Related Work

The FENCES system and the experimental design in this work are based on insights synthesized from collections of research within the multiple interconnected themes of communication, safety, augmented reality, and human-robot interaction, expanded upon in the subsections that follow.

Communication of Information in AR. McIntire et al. [172] find that stereoscopic 3D displays have equal or superior information communication performance as compared to non-stereo (2D) displays the majority of the time. Augmented reality (one form of stereoscopic 3D display) is a preferable option due to its dynamic visualization capabilities, non-obstruction of the visual field, and relative ease of use. Szafir and Szafir [245] indicate that most past research on human-robot interface design has centered around situational awareness and user control. While our system provides situational awareness in terms of spatial ownership, we look beyond control and towards back-and-forth communication between the human and the robot.

AR for Human-Robot Communication. A rich corpus of work on use cases and experiments exists regarding using AR for human-robot communication. Many systems are designed to improve communication **from the robot to the human**, such as providing insight into motion intent [51, 267, 244], assistive control predictability and legibility [26], aiding teleoperation [115], improving control handovers for autonomous vehicles [47], and using AR-assisted robot gestures [276]. Other systems exist that facilitate communication **from the human to the robot**, including programming or otherwise adjusting the system [158, 31, 202, 138], teleoperation [115], providing boundaries to the robot [240, 238], or functioning as a team [208, 39]. While our work builds on this growing body of research, we specifically address human *compliance* with a communicative system as it relates to safety.

AR and Safety. AR is increasingly used to improve worker safety in a variety of environments [236, 154]. Tatić and Tešić [253] presented a case study using AR to improve safety in an industrial environment by providing virtual safety instructions and other information. AR-equipped hard hats are also increasing in prevalence, indicating there is growing acceptance of using AR in high-risk environments. Our work leverages these findings and techniques in *spaces containing humans and robots*.

AR for Human Safety in Shared Spaces with Robots. A system from Choi et al. [42] provided safety signals in the form of a green, yellow, or red dot for low, medium, and high risk of danger in the corner of the user’s field of view. Makris et al. [167] also shaded regions of the workspace in red to denote the robot’s space or green to indicate the operator’s safe working area. In practice, for our system we found that users had difficulty distinguishing between yellow and green holograms, leading to our use of blue instead of green, but maintaining the overall principle of using color to denote ownership or imply safety.

Some primary applications for our findings include manufacturing and fulfillment centers. There are indications that humans working in close proximity to robots at Amazon Fulfillment Centers might alter their workflow to accommodate or support the work of their robot teammates [220], prompting the authors to ask how AR can further facilitate these human-robot teams. Ama-

zon has already initiated work on this front, as evidenced by the existence of a patent on an AR display for fulfillment center workers [61, 165].

In this work, we utilize augmented reality to provide both a communication modality and spatial ownership information for a person working collocated with an aerial robot and draw conclusions related to human compliance and safety.

3.3.3 The FENCES System

The FENCES system includes a Microsoft HoloLens 2 augmented reality head-mounted display (ARHMD), a Parrot Bebop 2 quadcopter robot, a Vicon Tracker motion capture camera system for tracking the robot and the user, and a computer performing sensor fusion, state management, and robot control. In the component descriptions below, the term “user” refers to the human participant.

FENCES was designed as a testbed for analyzing human behavior while interacting with AR and a collocated robot. Within the system, a user can request permission to traverse a controlled space in order to reach a specific goal location. Through the ARHMD, the user can see who has ownership of the spaces on the floor: the robot, themselves, or no one.

3.3.3.1 Microsoft HoloLens 2 ARHMD and User Interface

The Microsoft HoloLens 2 is capable of projecting images and text in the wearer’s field of view. The user interface was designed in Unity and consists of the following features, some of which can also be seen in Figures 3.1a and 3.1b: (1) A large 3-by-5 grid on the floor, with the 8 bins and table serving as boundaries. (2) The 1.5 x 1.5 meter grid squares are colored red, yellow, or blue, depending on whether they are “owned” by the robot, no one, or the human, respectively. (3) Billboards above each bin are labeled with a corresponding number and always face the user. They can be selected using a HoloLens “air tap” to indicate a user request. Audio feedback is provided when a bin/billboard is selected (“Bin [number] selected.”). (4) A “Home” billboard hovers above and behind the home base table. (5) Text confirming completion appears when the experiment has

ended. The ARHMD is the sole mode of communication between the user and the system. The user initiates a request to approach a bin by selecting its billboard, and the system may give permission to traverse the floor, indicated by shading the grid squares in blue that the user is permitted to enter.

3.3.3.2 Experiment Manager and Experimenter Interface

All of the robot goal locations, floor color configurations (and thus user access routes), and anticipated bin selections are *predetermined* by the experimenters and implemented as sequentially reachable states in the system. The states have transition criteria based on specific conditions being met: user location, robot location, and bin request.

3.3.4 Experiment Design

We designed this IRB-approved experiment ($n=20$) as a between-subjects study with two conditions. Participants were assigned pairwise randomly to conditions: odd numbered participants were randomly assigned a condition and the following even numbered participant received the opposite condition. Pairwise randomization is an unbiased assignment mechanism to ensure balanced cases when there is a guaranteed pair [76]. We recruited 22 participants, but two trials were discarded due to issues with the motion capture system. The participant population drew from students at our university and was 25% female, 5% nonbinary, and 70% male. On a scale from 1 (“Never interacted with”) to 5 (“Extensive experience with”), average experience across participants was 3.1 for robots and 2.4 for AR.

We deployed FENCES in an experimental flight space lab arranged to replicate an assembly environment with eight distributed parts bins along east and west sides (Figure 3.1a). The task space was approximately $8m \times 12m$. A table for the user’s workspace was at the south end, deemed “Home Base” for the human. Participants received an orientation at this table, which also contained the assembly workspace and instruction booklet. The experimenter and control equipment were behind protective netting to the west of the table.

After signing the consent form, participants read one page of instructions describing the experimental task. The activity involved constructing a small assembly with Mega Bloks according to a printed booklet of step-by-step instructions with words and photos (see Figure 3.8). They were instructed to collect the blocks from the bins in a strict order from the bins and told that they should only walk on the blue areas in the grid. They wore the Microsoft HoloLens 2 ARHMD described in Section 3.3.3.1, which provided the interface for users to request permission from the robot to traverse the space and obtain access to particular parts bins (see Figures 3.1a and 3.1b). Simultaneously, the quadcopter robot flew about the room, stopping at bins to simulate inventory checks.

The two conditions were designated “Shared Space” (SS) and “Turn-Taking” (TT). In the **SS condition**, the participant and the robot were permitted to work in the grid area simultaneously, in non-overlapping regions of the space. The robot never returned to Robot Home, a red, robot-only location at the north end of the grid analogous to the human’s “home base”. The entire task took approximately 15 minutes to complete in the SS condition. In the **TT condition**, the participant alternated with the robot occupying the floor space; while the robot conducted its inventory route, the participant was required to stay in their respective home base, and while the participant was collecting items from a bin and traversing the grid, the robot hovered at Robot Home. After each inventory excursion, the quadcopter returned to Robot Home via the same general path by which it had departed. Since the robot and the participant were never on the grid at the same time, the duration for the entire task increased to approximately 30 minutes. In both scenarios, the “ownership” of the grid squares (robot, human, or neutral/unowned) was communicated to the participant using the virtual grid described in Section 3.3.3.1 and pictured in Figures 3.1a and 3.1b.

These conditions were chosen to investigate behavior in two different yet equally relevant situations: one where the spatial ownership rationale was more recognizable (Shared Space) and one where the spatial ownership rationale and associated safety concerns were less obvious (Turn-Taking). Participants were not provided explicit explanations in either condition about why certain

1) Go to Bin 1 and collect these parts:



2) Assemble as shown below:



(a) An example of the instructions provided. Each appeared on separate pages for clarity. (b) The completed assembly of multicolored MegaBlocs.

Figure 3.8: The task (a) instructions and (b) final assembly.

regions were permitted or prohibited, only what the colors denoted. Because we were investigating behavior with respect to the floor ownership as designated in AR, we do not compare their behavior to an AR-free condition. Further, without any indicator of spatial ownership or a significant deviation of the quadcopter's behavior, travel through the space would have been prohibitively unsafe for participants.

Immediately after the task ended, participants answered verbal questions about their experience in an interview with an experimenter. They were asked about their thoughts and behavior during the experiment, as well as whether they perceived any inefficiencies and whether they felt unsafe. Finally, they responded to a survey consisting of Likert (5-point scale) and free response questions.

3.3.4.1 Trap Scenario

Partway through the experiment, the participant requests access to Bin 4 and access is granted (Figure 3.1a). Once the participant arrives at Bin 4, only the grid squares along the northern edge remain blue while the rest of the workspace floor turns red, effectively eliminating their route back to Home Base (Figure 3.1b). The system then begins a 60-second timer, after which the path back to Home Base will reappear. The quadcopter hovers adjacent to Bin 1 in the SS condition and hovers at the Robot Home in the TT condition.

3.3.4.2 Hypotheses

Through the system and experiment described above, we test the following hypotheses: **H1:** Participants will feel safer in TT than in SS due to the reduced proximity to the quadcopter. This will lead to increased deviations in TT, as participants will rely on potentially faulty reasoning (i.e., based only on priors and directly observable features) when determining whether to follow the system guidance. They will also spend more time on the grid in SS due to increased caution near the robot. **H2:** Longer or less direct routes will invoke more deviations from the blue path than shorter or more direct routes. Thus, the trap scenario will also invoke deviations that participants will self-justify.

3.3.5 Results

3.3.5.1 Mixed Methods in HRI

For a model of mixed methods analysis, we consulted Veling and McGinn's [262] recent survey of 73 qualitative research papers in human-robot interaction, specifically the categories of insights-driven, design, and hypothesis-driven studies. There is a substantial history of prior work in HRI that use qualitative and mixed methods [164, 273, 232]. Using widely accepted qualitative methods we gathered data in semi-structured interviews as well as textual analyses [262], and coded the responses for repeated key words and themes.

3.3.5.2 Trap Scenario

A striking 25% of participants chose to walk through the red and yellow regions to return to Home, disregarding the instructions they had received at the start to only walk through blue regions. Three were in the TT condition while two were in the SS condition, showing similar rates of non-compliance regardless of robot proximity.

- “I knew I was faster than it, so [wherever] it was gonna go I was gonna get out of dodge before it could get there.” (TT)

In fact, in one case it seemed that *because* a participant had high trust in the robot’s consistency, they disobeyed the floor colors to return to Home Base.

- “I can see that it’s safe, so [walked through the red].” (TT)

Eleven of the 20 participants became impatient or assumed a malfunction when the trap scenario began and selected the “Home” button as a solution; 7 participants considered requesting another bin to generate a path, such as one close to Home, or Bin 4 again (the bin where they were trapped); 2 participants admitted that they considered going around the experiment area, outside the grid entirely.

- “I did come close to wondering whether [to walk] around the outside because...nothing will be there...” (TT)

When asked why their path back to Home disappeared, participants generally thought that there was a software issue ($n=7$) or that the robot was claiming the area ($n=9$).

- “It seemed like there was a glitch so I broke the rule [and] went straight through.” (SS)

However, there was no significant correlation between participants’ reasoning about why the path disappeared and their decisions about what to do, suggesting that all of the reasons provided warrant consideration. Furthermore, we can see that when an autonomous system lags, users will not wait patiently; instead they desire ways to work around the lag.

3.3.5.3 Safety, Efficiency, and Trust

One of the most remarkable results from this study was that **all participants felt safe during the experiment, with the exact same distribution across both conditions**. Given the statement, “I felt safe throughout the exercise,” all responses were either 4 or 5, with an average of 4.7 (see Table 3.2). Furthermore, 7 participants mentioned the word “safe” in the interview *before* they were asked whether anything felt unsafe about the experience. Two participants used the word “safe” in their response to the question, “Did you find anything inefficient about this process?” One participant (SS condition) believed the system was too safe:

- “It’s overly safe...there’s not enough risk involved.” (SS)

When asked if they thought anything was inefficient about the system, of the 20 participants, 17 identified inefficiencies, while 3 did not. One TT participant described a SS environment that would be more efficient, but SS participants had suggestions as well:

- “There were...times where there was a yellow part that didn’t belong to anyone, and it still made me go around.” (SS)

Participants also volunteered their thoughts about trust, sometimes combined with issues of safety and efficiency:

- “...I trusted the robot to stay in its red areas.” (TT)
- “I trusted it. I think it was very safe at the cost of efficiency, I’d be comfortable with less safety if possible.” (SS)

As presented in Table 3.3, participants in the SS condition felt that the robot was more fair than those in the TT condition, suggesting a willingness to sacrifice safety for a perception of fairness. As expected and required by the experiment design, participants were closer on average to the robot in the SS condition (3.7 m) than in the TT condition (4.4 m), with $p < 0.0001$ (Figure 3.9).

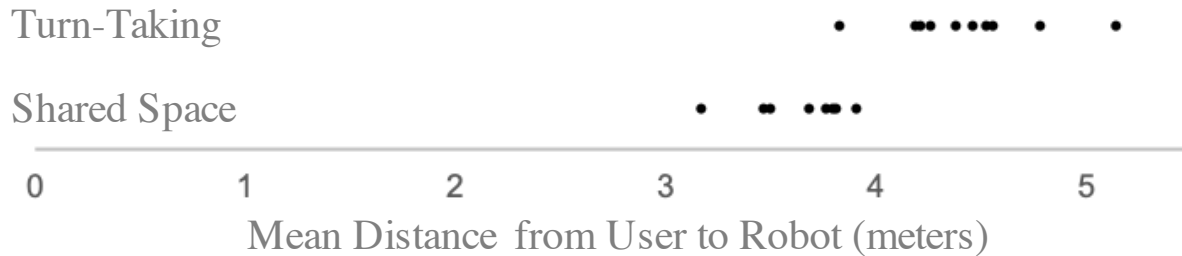


Figure 3.9: Mean distance from the robot sorted by condition. Across all participants, mean distance in SS = 3.7 meters, while mean distance for TT = 4.4 meters, $p < 0.0001$.

However, the self-reported feelings of safety showed identical data for the two conditions (Table 3.2), **suggesting that participants felt as safe nearly 3 meters from the robot as they did when it was waiting predictably in Robot Home.**

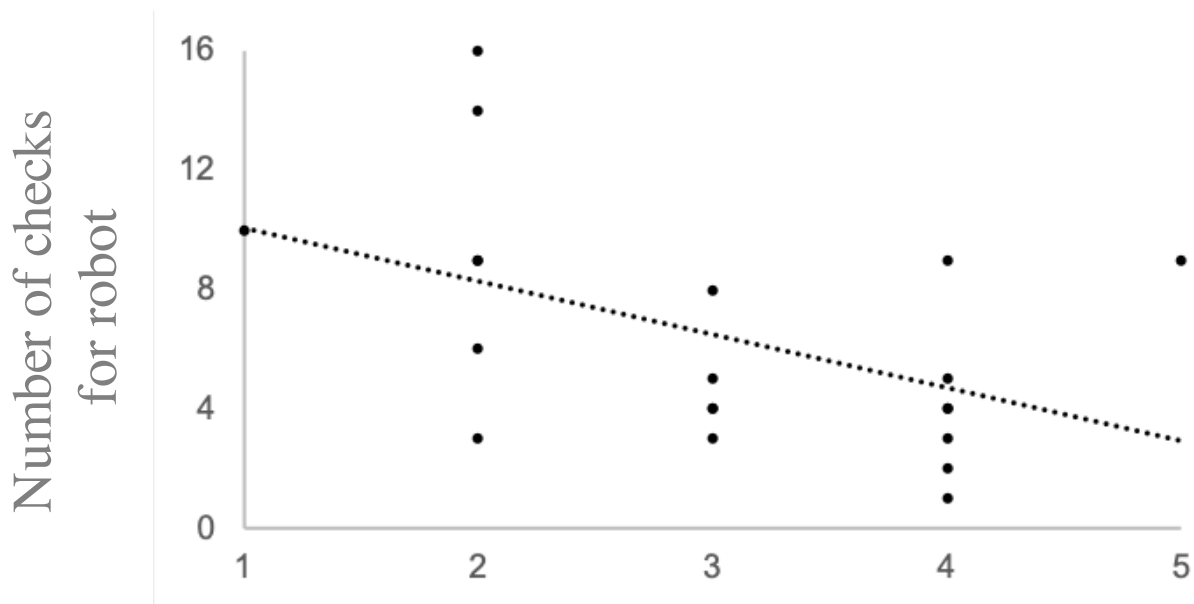
Table 3.2: Summary responses to select survey items. Parentheses indicate SS responses. 1 = Strongly disagree, 5 = Strongly agree.

Statement	1	2	3	4	5
I felt safe throughout the exercise.	0	0	0	6(3)	14(7)
I deviated from the given path during the exercise.	15(8)	0	0	1(1)	4(1)
I felt informed throughout the exercise.	1	1(1)	4(1)	7(3)	7(5)

Table 3.3: Mean responses, by condition, to select survey items. $*p < 0.05$

Statement	Shared Space	Turn Taking
I thought the robot was fair.*	4.0	3.1
I liked the way I interacted with the AR device.*	4.7	3.8
I thought the robot was very responsive to my requests.	3.7	3.0
I thought the robot was intelligent.	3.4	2.7

Participants in the SS condition responded statistically significantly more positively to the statement, “I thought the robot was fair,” (see Table 3.3), suggesting that the longer wait time in



“I thought the robot was intelligent.”
 1=Strongly Agree, 5=Strongly Disagree

Figure 3.10: Relationship between participant response to the item, “I thought the robot was intelligent” and how frequently they looked for the robot while on the grid ($p < 0.05$).

TT implied a level of unfairness.

- “The robot thought its priorities were more important.” (SS)

Participants further personified the robot and the system in some of their interview responses:

- “Sometimes you...had to...wait a little bit for it [the robot] to realize, ‘Wait, I don’t need that square, I can give it up.’” (SS)
- “It knew when I was on the field and when I wasn’t.” (TT)

We also noted how many times each participant checked the robot’s position by looking at it while they were on the grid. Data shown in Figure 3.10 indicate with statistical significance that the higher they perceived its intelligence, the fewer location checks a participant made. Repeated checks for the robot suggest that the human is engaged in tracking the robot. As multitasking

increases cognitive load [190], **this suggests that increasing the perception of intelligence can be a powerful way to reduce cognitive load.**

3.3.5.4 AR Interface

Despite having the same user interface across both conditions, participants in the SS condition responded statistically significantly more positively to the statement, “I liked the way I interacted with the AR device” (Table 3.3). They generally liked seeing everything that was in the AR view, except for 1 participant who stated that the grid hologram obscured the robot, making it difficult to see where the drone was, which he also said made him feel less safe. (This participant still responded with 4/5 to “I felt safe throughout the exercise.”) Participants consistently made the following suggestions for other information to share in AR: robot intent or priorities ($n=8$), a timer showing remaining wait time ($n=4$), task instructions ($n=4$), and an indicator for the robot location ($n=2$) were some of the most popular responses.

Participants had a number of suggestions for additional information they would like to see in the display. By showing the red robot-owned regions, we intended to convey the current and near-future movements of the quadcopter. However, over half the participants ($n=11$) desired even more insight into the robot’s intent, priorities, and planning, with which they felt that they could make their own decisions about how to move about the space. However it is unclear whether, with this additional insight, they would continue to stay within the blue grid squares or feel empowered to make their own, potentially deviant, choices for movement around the space. **This information could be useful when designing such systems to know what kind of deviations to expect and how to prime users to use the systems as intended.**

3.3.5.5 Support for Hypotheses

The first hypothesis addresses efficiency between the two conditions. However we found no significant difference after comparing the time SS participants spend on the grid to the TT participants’ time. We also analyzed the mean participant distance from the robot as compared

to participants' perceptions of safety. Looking at these data in concert, we see that despite SS participants being closer on average to the robot throughout the experiment (Figure 3.9), they were just as likely to report that they felt safe throughout (Table 3.2). While 75% of participants stayed within the blue regions throughout the experiment, the remainder deviated by walking through the red and yellow regions during the trap scenario. Considering this is a safety-critical system, we view 25% non-compliance as an alarming result. Participant deviations occurred in both SS and TT conditions, and some participants felt the grid ownership guidance was unnecessary. The data partially support H1 in that participants use faulty priors to justify feelings of safety, but there were no differences in perceived safety across the conditions. The data support H2.

3.3.5.6 Limitations

Experimenters were present in the same room as the participant for reasons of safety and practicality, enabling participants to communicate with the experimenters at will, which happened on three occasions. In those instances, a preplanned response was given that did not offer any information about the task or system. Additionally, the motion capture capability varied. Two participants were more difficult to track than others, requiring experimenter intervention to advance the system to its next state. This induced a level of variability in responsiveness to built-in triggers, such as floor colors changing upon the participant's return to Home Base. Participants also had mixed success learning the HoloLens "air tap" gesture, possibly affecting their impressions of the system. This work was also limited by the participant population: all were STEM majors; 70% identified as male.

3.3.6 Discussion and Findings

In our collocated, physically unprotected environment, participants had to rapidly draw conclusions about the robot's current state, its intentions for the future, and the trustworthiness of its communications. One of the most surprising results was the demonstrated and reported **overwhelming feelings of safety by all participants**. As explored in Section 3.3.5, all participants

shared that they felt **safe** throughout the experiment, some explicitly stated that they **trusted** the robot to stay in its red areas, and they generally felt **informed** throughout the exercise (Table 3.2). This resulted despite not receiving any explanations about the robot’s trustworthiness or reliability. Prior work has shown that humans tend to over-trust robots, even in high-risk situations and when they have experience with the robot misleading them [207]. Our work adds to the evidence of the potential to over-trust autonomous systems and leads to **Finding 1: Humans working in close proximity to robots appear willing to sacrifice some amount of safety to achieve increased efficiency.**

Lee and See [149] reported that written descriptions induce high levels of initial trust, and that trust in automation begins with faith, then dependability, and finally predictability. Our system initialized trust with the written task description and demonstrated consistency until the trap scenario; participants built on their levels of trust as the task progressed.

Research on trust and safety in high-risk situations contain some key ideas that are useful for understanding the behavior of our participants. Although much of that work relates to trust in people, we observed evidence that participants were personifying the robot. Furthermore, some even viewed it as intelligent (Figure 3.10). Pidgeon et al. [198] define *critical trust* as a “practical reliance on other people combined with a skepticism of the system” [104]. Prior work also demonstrates that it is possible to trust people but not trust dangerous situations; in our experiment, as the trap condition occurred, participants had established some level of trust with the robot, however the system behaved unexpectedly. Five participants then trusted the robot to continue behaving as it has been, simultaneously distrusting the floor colors, ignoring them to return home. Four other participants trusted the system to allow them a path back eventually and waited for this to occur. Further evidence indicates that “trust and distrust are unlikely to lie on the same dimension” [104]. We can conclude that an optimal model of safety requires both critical trust and distrust, leading to **Finding 2: Users desire insight into the decisions and priorities of an autonomous system to help them understand the reasoning behind its actions, decrease frustration, and help them make their own decisions about how to act during**

uncertainty.

In human-machine interactions that are facilitated by an interface, it is the interface that establishes shared expectations and trust [111]. The ARHMD and the AR visualizations play a crucial role in the participants' trust development. The virtual images and text are the only methods the system possesses to communicate any information to the user; aside from the actual robot behavior and any prior experience, almost all trust is derived via the ARHMD. By incorporating the suggested features from the participant responses - such as robot intent, prioritization, or wait time - trust and safety can be increased, which informs **Finding 3: Increasing the perception of a collocated robot's intelligence could significantly decrease a worker's cognitive load.**

3.3.7 Design Recommendations

Placing autonomous robots into a shared environment with humans introduces risks and safety considerations. Our study has demonstrated that augmented reality is not necessarily a clear solution to those problems; simply displaying spatial ownership does not dictate safety nor compliance, especially when unexpected events occur. We conclude with recommendations for collocated human-robot systems utilizing AR to aid communication, informed by our results and findings:

Recommendation 1: Provide deviation warnings to deter self-justified rule-breaking that could result in additional risk. **Recommendation 2:** Brief people about the robot's abilities and limitations as part of system training to mitigate intelligence and over-trust perceptions. **Recommendation 3:** Include live visual information to improve real-time understanding of system operation. **Recommendation 4:** Provide training on actions to take during uncertainty; enable the system with corresponding capabilities.

Chapter 4

Explainable Visual Guidance for Human-Robot Search Tasks

4.1 Motivation

This chapter introduces multiple frameworks focusing on the problem of collaborative navigation and search in large, partially observable environments. The core aim of this chapter is developing live communication techniques for integrating humans into multi-agent, heterogeneous planners. The first work (Section 4.2) introduces MARS, a multi-agent algorithm for multi-objective environmental navigation and search that simultaneously commands robotic agents and provides explainable guidance and decision support to human teammates, delivered via an augmented reality interface. We present multiple categories of autonomously-delivered guidance: prescriptive (direct recommendations for what actions the human should take), descriptive (latent environmental data that the human can use to aid in their decision-making), and a combination of both.

We evaluate these guidance types in a collaborative, 3D augmented reality game inspired by Minesweeper, where participants attempt to defuse mines hidden throughout an environment, assisted by a simulated robot teammate possessing a noisy mine detection sensor. We find the combined guidance type performs best in task performance, as well as subjective measures such as usefulness for decision-making and interpretability. With combined guidance, participants are able to receive direct recommendations to reduce thinking time, while also receiving the rationale behind those recommendations, allowing them to decide for themselves whether to follow the guidance or use their own judgment. This work was presented at the International Conference on Autonomous Agents and Multiagent Systems (**AAMAS 2022**) [250], where it received a nomination for best

student paper.

Though the results from this experiment yield promising design recommendations for robotic decision support, the Minesweeper domain is an abstract form of search task, with a small, discrete state space. Section 4.3 describes recent work on extending the MARS algorithm for use in large, realistic domains through the use of spatial hierarchy. This new algorithm, named H-MARS, is capable of reasoning, planning, and providing explainable guidance at multiple levels of spatial granularity, depending on the requirements of the search task in progress. We evaluate the H-MARS algorithm against other candidate algorithms for multi-agent search, including MARS and a non-reinforcement learning method, by running simulations of a mine detection and defusing task across multiple irregular urban environments and agent configurations, the largest of which involves planning over an area of 77 acres. Through this evaluation, we show that H-MARS is the most capable of providing valuable real-time decision support in large, realistic environments, leading to the highest average percentage of targets found. This work has not yet been published, though an earlier treatment of the H-MARS concept was presented at the Workshop on Explainability for Human-Robot Collaboration (**ExpHRC 2024**) at the HRI 2024 conference [251].

Within the context of this thesis, this chapter focuses on using visual communication to transfer large quantities of knowledge from robot teammates to humans, in ways that are actionable, and that contribute to mental model alignment. The robots in these experiments, by the nature of the tasks, possess information that human teammates lack, and so act as decision support tools, aiming to enhance human performance however they can. Additionally, this chapter touches on a particularly difficult aspect of robot-to-human communication: conveying noisy and uncertain information. MARS and H-MARS use spatially-anchored visualizations and color, with the goal of helping humans better internalize complex, uncertain data streams involving probabilities, allowing them to make use of such information during real-time collaboration.

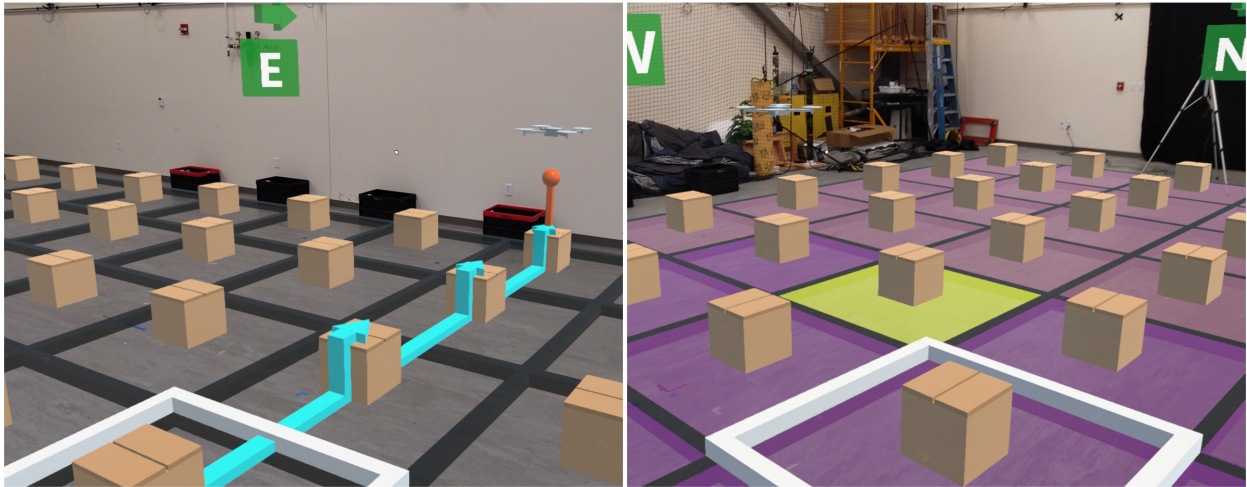


Figure 4.1: AR-based interfaces for prescriptive (Left) and descriptive guidance (Right) in the Minesweeper domain. In the prescriptive condition, suggested moves are shown as cyan arrows between grid squares, with suggested defuse actions indicated by the orange pin (underneath the virtual drone teammate). 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).

4.2 Descriptive and Prescriptive Visual Guidance to Improve Shared Situational Awareness in Human-Robot Teaming

4.2.1 Introduction

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. 4.1). 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. 4.1).

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 environ-

mental 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.

4.2.2 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 [122, 45, 136]. 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 [245]. 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 [87]. 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 [2].

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. [20]. Furthermore, Colley et al. showed that visualizing the internal information of autonomous vehicles improves trust and situational awareness [46]. 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 [204, 195]. Fraune et al. investigated the use of mixed reality interfaces for humans monitoring and commanding drone teams for search and rescue [83]. Kunze et al. show the effectiveness of AR to visually communicate uncertainty during automated driving [142].

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 [249, 36]. Furthermore, explainable AI (xAI) can help complex models become more understandable by human teammates, allowing for faster debugging when unexpected behaviors or failures occur [205, 114]. Visualization is a common modality for presenting explanations through xAI [217]. Visual information presentation is ideally suited to explanations that are complex, long, re-referenced, and which involve uncertainty or noise [60]. 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 [161], model-agnostic methods such as LIME [205], and saliency map methods such as Grad-CAM [226]).

Other recent studies have utilized case-based explanations as visualizations to expose overconfidence in models and visualize class boundaries [22]. A related technique is visual counterfactuals [155, 175] (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 [174, 178]. 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.

4.2.3 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.

4.2.3.1 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 [86, 280, 281]. 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 4.2.4.

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 $D = \{d_{12}, d_{13}, \dots, d_{(N-1)N}\}$, 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.
- $T : S \times A \rightarrow \Pi(S)$ is the state-transition function describing the model's state transition dynamics.
- $R : S \times A \times S \rightarrow \mathbb{R}$ 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:

$$R(s, a, s') = \alpha(0.5 - |0.5 - pmf(s')|) + \beta \sum_{n \in N} d_n - 1 \quad (4.1)$$

In Equation 4.1, 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 4.1 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.

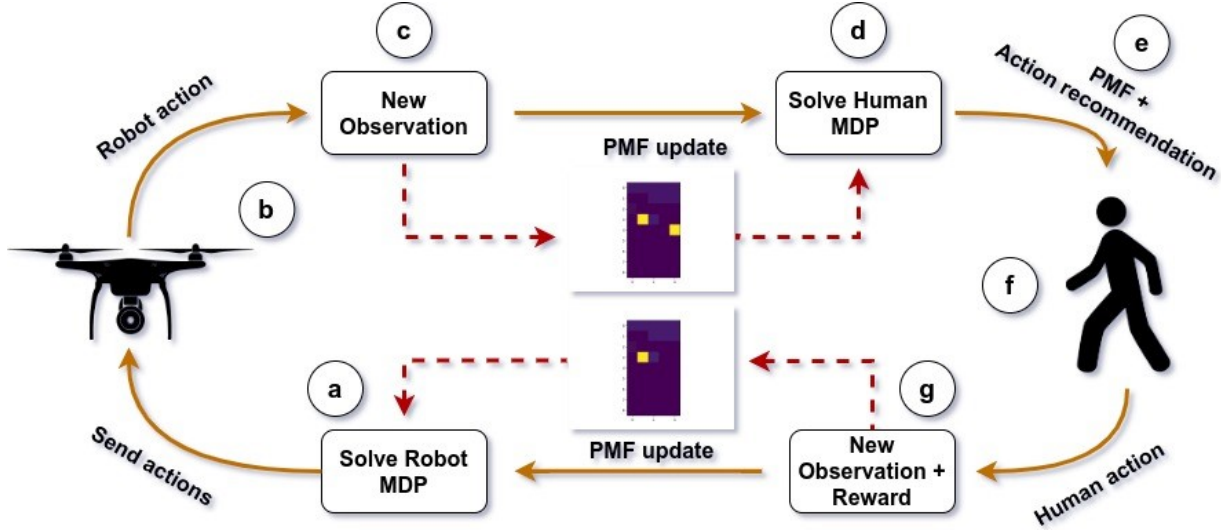


Figure 4.2: Algorithmic flow: a) the robot’s MDP is solved, parametrized by the PMF, and actions are sent to all agents, b) the robot takes an action and c) observes a new potential mine, updating the PMF (the new mine is visible as the rightmost yellow square), d) the updated PMF is used to solve the 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)

4.2.3.2 Generating Assistive Guidance

Here we present our approach for generating assistive guidance for human teammates in uncertain environments. Similarly to section 4.2.3.1, 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.

$$S = \begin{bmatrix} W \\ \text{goals_left} \in \mathbb{N} \\ \text{is_goal} \in \mathbb{B} \end{bmatrix}$$

- A is the set of possible task-relevant human actions.

- T and R are similarly defined as seen in Section 4.2.3.1.

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:

$$R(s, a, s') = pmf(s') - \beta * is_goal_s - penalty \quad (4.2)$$

where,

$$penalty = 1 - \alpha * goals_left$$

The first term of Equation 4.2 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 4.2 to get the expected immediate reward as follows:

$$\begin{aligned} \mathbb{E}(R) = (1 - pmf(s)) * (pmf(s') - penalty) + \\ pmf(s) * (pmf(s') - penalty - \beta) \end{aligned} \quad (4.3)$$

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

$$R(s, a, s') = \beta * is_goal_s - penalty \quad (4.4)$$

The first term of equation 4.4 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 4.2. Expanding Equation 4.4, the expected immediate reward is:

$$\begin{aligned} \mathbb{E}(R) = pmf(s) * (\beta - penalty) - \\ (1 - pmf(s)) * penalty \end{aligned} \quad (4.5)$$

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

Algorithm 3: 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 $\overline{S}_R = \{s_1, s_2, \dots, s_{n-1}\}$, 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 $\pi_R^* \leftarrow solve_policy(M_R, pmf)$;
- 4 $\overline{A}_R \leftarrow \pi_R^*(\overline{S}_R)$; // Get optimal actions for each robot
- 5 $\overline{S}_R \leftarrow send_actions(\overline{A}_R)$; // Send optimal actions
- 6 $pmf \leftarrow update_pmf(\overline{S}_R)$; // Get observations
- 7 $\pi_H^* \leftarrow solve_policy_human(M_H, pmf)$;
- 8 $\overline{A}_H \leftarrow rollout(\pi_H^*, s_h)[k]$; // Get actions for human
- 9 $recommend_action(\overline{A}_H, pmf)$
- 10 $s_h, r_h \leftarrow observe_human_action()$
- 11 $pmf \leftarrow update_pmf(r_h)$

4.2.3.3 Algorithm

In this section, we outline the details of MARS, as presented in Algorithm 3. 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 4.2, 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 4.2). 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 4.2.4. 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).

4.2.4 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 3. 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.

4.2.4.1 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. 4.1 Left), an AR visualization technique which has shown effectiveness

for navigation tasks [102].

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.

4.2.4.2 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. 4.1 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 [78].

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.

4.2.5 Experimental Validation

We evaluate the utility of the AR-based visual guidance modalities presented in Section 4.2.4 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.

4.2.5.1 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 4.3). 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 [23, 53]. 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.

4.2.5.2 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 [246, 230].

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.

4.2.5.3 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. 4.3). If they moved from a square containing a mine without defusing, the mine would be unintentionally detonated. Unlike Minesweeper, this did not end the game; participants were simply told beforehand that this would contribute to a low score.

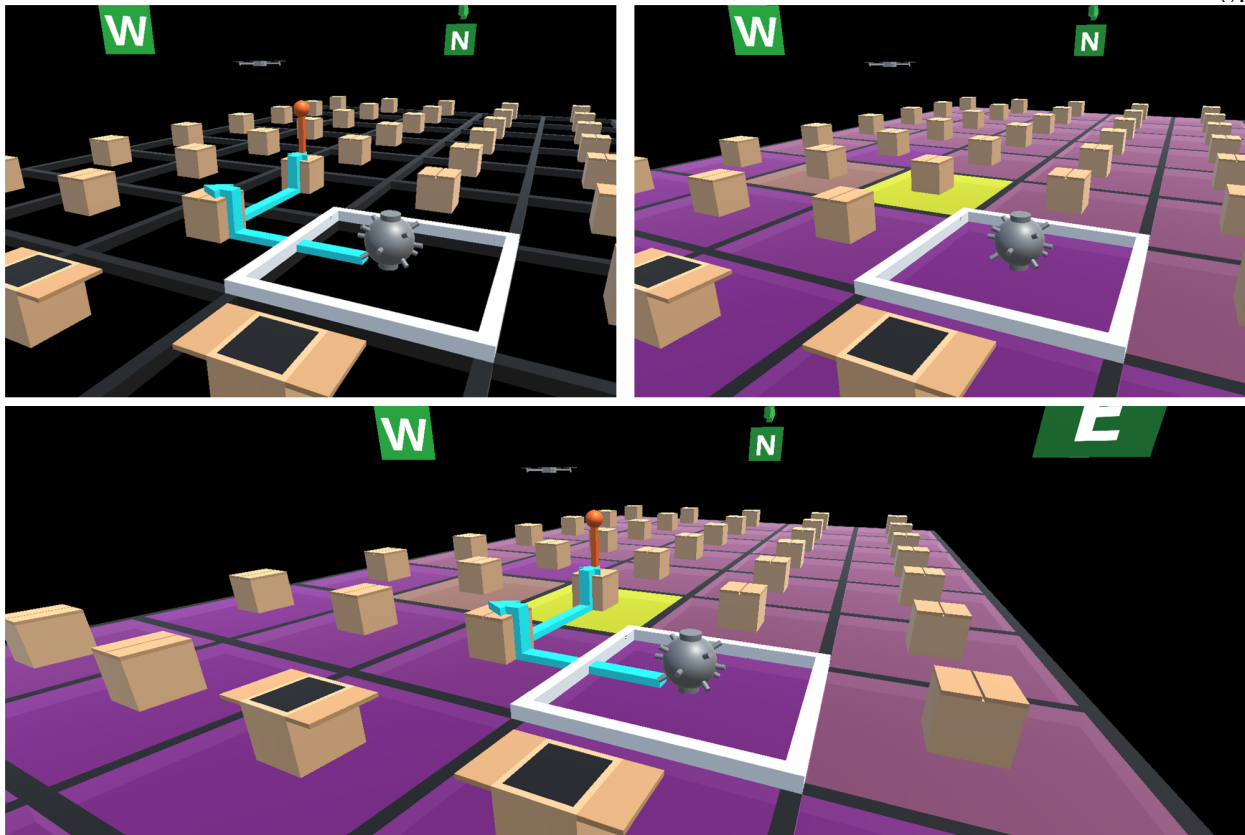


Figure 4.3: The three experimental conditions. A white square marks the user's current location where they have defused a mine. Top-left: 'arrow' condition, Top-right: 'heatmap' condition, Bottom: 'combined' condition.

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.

4.2.5.4 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.

4.2.5.5 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.

4.2.5.6 Measurement

We had 19 participants (12 males, 7 females) in our IRB-approved study, ranging in age from 18 to 37. 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 [120, 150], interpretability and decision-making [269, 120], and stress and workload (NASA-TLX) [110]. 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).

4.2.6 Results and Discussion

4.2.6.1 Analysis

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 certainly “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

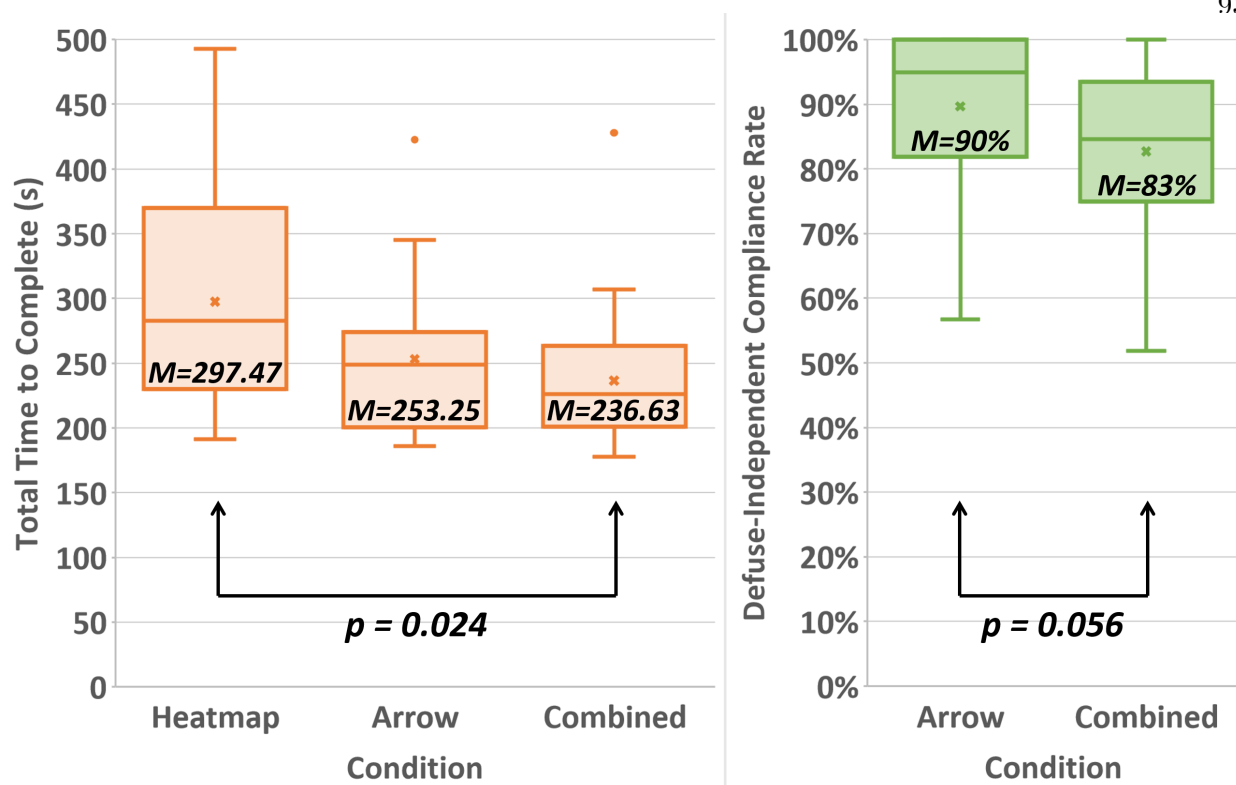


Figure 4.4: ‘Combined’ visualization achieves the Total Time performance benefits of ‘arrow’ while allowing for reduced rigidity in suggested action compliance.

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, $p = 0.024$).

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 par-

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.”*

4.2.6.2 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 [246, 72].

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) [130]. 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 [266, 59].

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.

4.3 Hierarchical Multi-Agent Reinforcement Learning with Explainable Decision Support for Human-Robot Teams

4.3.1 Practical Limitations of the MARS Framework

The MARS framework, as described in Chapter 4.2, represents a promising technique for providing explainable decision support to human teammates in environmental navigation and search tasks, thus better integrating humans into complex multi-agent planners. However, the real-world domains in which MARS is usable remain severely limited, due to issues of computational complexity and environmental diversity. MARS was evaluated in a discrete, regularized gridded environment overlaid onto a single large room, with a total of one human agent, one autonomous agent, and 45

states. Real-world domains in the navigation and search class (urban and wilderness search and rescue, passive environmental surveillance, explosive ordnance disposal, radiological device recovery, etc.) are often characterized by large, irregular environments with variable traversability. For such environments, the MARS algorithm as described in Section 4.2.3.3 would quickly run into bottlenecks on state and agent count, preventing it from providing real-time feedback at the level of granularity necessary to be useful for decision support.

This motivates an extension to the MARS framework capable of addressing this practical limitation, to serve as a demonstration of the MARS concept’s practical utility, and helping bridge the gap between the measured responses to autonomously-provided guidance from abstract domains to more realistic human-robot collaborative search domains.

4.3.1.1 Spatial Hierarchy in Search

For this extension, we aim to exploit the inherently hierarchical nature of search, and the hierarchical structure of human psychology surrounding search problems [103, 1]. Search as a class of problem typically involves progressing over time from considering a large initial search space to smaller and smaller regions as environmental uncertainty is collapsed and targets are narrowed in on. The goal of autonomous decision support in a search problem is not necessarily to guide a human teammate to a target, but rather to guide the human close enough to a target that they will quickly locate it on their own.

Take as an example the domain of avalanche rescue. Organized avalanche rescue teams often involve specialized sub-roles divided among members: they may include trained dogs attempting to locate a scent, humans sweeping the area attempting to localize where victims are roughly likely to be located, either visually or with the aid of an avalanche transponder beacon, and one or more humans methodically probing downward through the snow with long poles searching for a strike, attempting to identify possible shoveling locations [237]. These subtasks exist at differing levels of spatial granularity - from the coarse refinement of a large initial search area into much smaller regions likely to contain a victim, to the fine-grained tasks of probing through the snow and digging

within those regions to actually locate victims.

While aerial robotic teammates would not provide much value to the scent-tracking, probing, or shoveling aspects of avalanche rescue, they could dramatically improve the coarse localization phase of search. Drones equipped with cameras and avalanche transponder beacons could traverse the steep, treacherous, snow-covered terrain of the search space substantially quicker than their human teammates, sweeping a much larger area per unit time. In this setup, the drone teammates' responsibility would be to collapse environmental uncertainty and guide the human team to sub-regions within the environment in which their unique action sets (probing, digging, and rescue/recovery operations) would be most efficiently used. The coarse step of the hierarchical search would be largely the responsibility of robotic teammates, whereas the fine-grained step would be performed by humans.

Many tasks where robots and humans collaborate to search for targets, each with their own requirements for environment size and spatial resolution of search, can be modeled computationally using such a hierarchy. A continuous search space may be divided into a discrete set of states, each of a size appropriate for the final, fine-grain search phase, and those states may be grouped into larger, spatially contiguous regions. By allowing a multi-agent planner to reason over multiple levels of spatial hierarchy (states within a region, or regions within an environment), reinforcement learning solutions such as MARS become computationally practical, without sacrificing the ability to efficiently locate search targets in real-world, continuous environments.

With this in mind, it is useful to transform the discrete state space of the MARS framework into a hierarchically discrete one, such that the system is capable of reasoning, planning, and providing explainable guidance at different levels of spatial granularity, depending on the real-time requirements of the search problem in progress. This provides two likely benefits: first, the algorithm can theoretically be run and utilized in environments of arbitrary size, even on robot hardware of limited computing power, by properly leveraging hierarchical planning to limit the number of states considered at each time step, and even by extending the hierarchy recursively to three or more levels if required. Second, tying human guidance to this limited number of states will

improve the interpretability of guidance in environments with large state spaces: human teammates will not be oversaturated with environmental information, instead attending to the data most useful for influencing their immediate decisions, at an appropriate level of abstraction.

4.3.2 Hierarchical Min-Entropy Algorithm for Robot-Supplied Suggestions

At a high level, our novel algorithm functions similarly to MARS as described in [250]. We call this new algorithm 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 explainable 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.

Unlike the original MARS algorithm described in Section 4.2.3.3, all agents solve their own MDPs during each iteration, rather than encoding for joint actions between all robot agents. This is for reasons of flexibility and reduced computational complexity: solving multiple MDPs adds computation linearly with respect to the number of agents, while solving a single MDP with joint actions has computation time exponential with respect to the number of agents, quickly becoming infeasible for domains with more than a few robotic teammates. This design change comes at the cost of slightly suboptimal robot actions. Since the MDPs are updated and solved iteratively in real time, and are solved utilizing implicit information about the state of other agents, via the *pmf* term and the set of distance features D (Section 4.2.3.3), the suboptimality of these actions is unlikely to propagate beyond the time-scale of an individual iteration.

The H-MARS algorithm also crucially differs 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

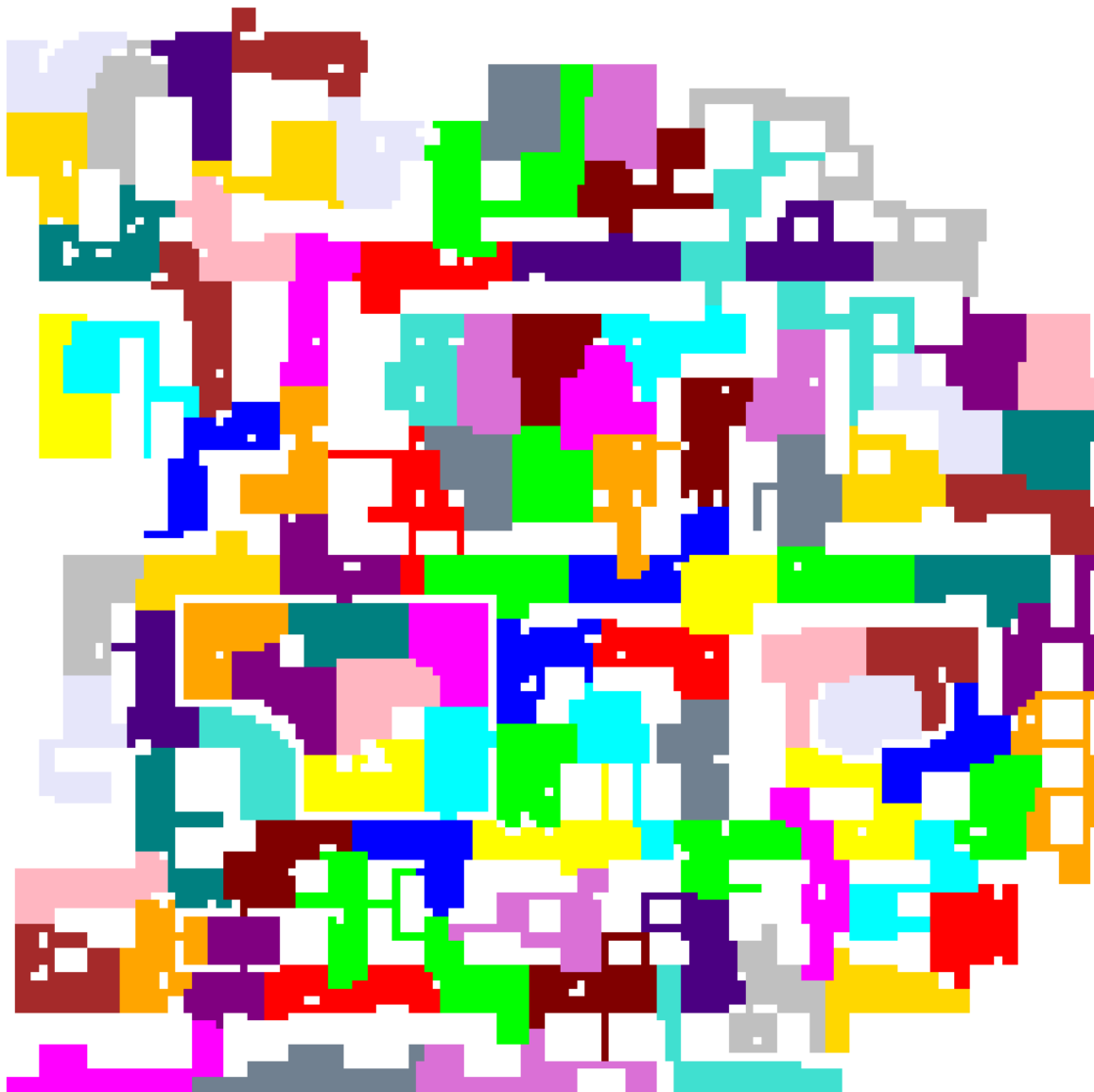


Figure 4.5: Results of graph partition on a 2-dimensional projection of a 3-dimensional experimental environment. In this example, the environment is divided into approximately 10,000 3 meter x 3 meter grid-squares, grouped into 100 regions of roughly 100 grid-squares each, shown as colored regions, with impassible obstacles rendered in white.

levels of spatial resolution. For example, an environment could be divided into regions, which are themselves divided into sub-regions, and so forth, until the final level of individual states is reached.

4.3.3 Hierarchical Environment Creation

As a preprocessing step, we first discretize a 2D representation of the continuous environment in which we wish to operate, dividing it into a grid of a desired spatial resolution for the final, fine-grained search phase. The traversability of each grid-square is determined. In practice, this would require the generation of a top-down map prior to human-robot collaboration, likely based on aerial or satellite imagery of the target environment. The concept of using the H-MARS algorithm on a map that has no prior, which must be dynamically generated through autonomous sensor readings, has not yet been explored.

These grid-squares are used as nodes to form a graph, with edges connecting adjacent, traversable nodes. We then use the METIS graph partition algorithm [131] over this graph, producing contiguous regions of reachable states. To optimize for computational efficiency while running the H-MARS algorithm in real-time, the number of regions produced should roughly equal the n th 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 [30]. The only aspect of computation that is unaffected by the choice of hierarchy is PMF normalization during every iteration, following the computation of updated PMF values within observed states. This step is of linear complexity with respect to the state count, and is an insignificant contributor to overall computation time, compared to solving each MDP.

An example of this n th root principle used in environment creation can be seen in Fig. 4.5, where a 2D traversability map is divided into approximately 10,000 low-level states, which are grouped into 100 regions of roughly 100 grid-squares each. This means that the H-MARS algorithm, for a 2-level hierarchical environment, is never tasked with considering more than 100 states during any iteration.

4.3.3.1 Algorithm Phases

The MARS algorithm is laid out in Algorithm 3. H-MARS operates much the same, with a key distinction: rather than maintaining static MDPs M_H and M_R for the human and robot teammates throughout execution, the components of each MDP, (S, A, T, R) are capable of being swapped between three variants during each loop of the algorithm. These MDP variants are tied to the three phases described below, representing different levels of hierarchy, each of which is run on a per-agent basis (meaning different agents can decide on actions derived from different phases within the same iteration of the algorithm).

Phase 0 (Local Window Search): The algorithm begins with an initial phase considering individual states within a limited distance horizon of each agent. This is to avoid edge cases that would arise by starting immediately with Phase 1, involving potentially high-reward actions taking agents to physically nearby states that arbitrarily happen to lie across a region boundary. By considering these actions first, we avoid the situation where they receive an oversized reward penalty, given in Phase 1 to represent the average time required to travel to a separate target region.

For each agent X , $MDP_X = (S_X, A_X, T_X, R_X)$:

- S_X : The world features within the state space comprise all individual grid-squares within a distance d of agent X 's current location.
- A_X : The action space consists of movement primitives (North, South, East, West, Stay).
- T_X : The transition function determines which actions are legal on a per-state basis (checking for adjacency of grid-squares, and discounting actions that would take the agent into obstacles or out of bounds).
- R_X : The *pmf* term of the reward function is on a per-state basis, with an individual probability value for every grid-square. The *penalty* term is tuned to represent the time required to take a single, atomic action.

Phase 1 (Inter-Regional Search): If a tuned reward threshold within Phase 1 is not passed, the algorithm moves to consider entire regions as single states, with the state-wise PMF used to calculate the expected cumulative number of targets to be found per region. The algorithm decides whether it is more advantageous to stay and search within the agent’s current region, or travel to a new, more target-rich region, considering the added movement penalty for taking the time to travel to that region, proportional to the distance between the current and target region’s centroids. If the algorithm decides an agent should move regions, it provides this as a recommendation if the agent is human, or commands actions pathing to the nearest edge of the new target region if the agent is autonomous. If the algorithm instead decides to stay and search within the current region, it progresses to Phase 2.

For each agent X , $MDP_X = (S_X, A_X, T_X, R_X)$:

- S_X : The world features within the state space comprise all regions within the environment.
- A_X : The action space consists of a ‘Stay’ action, representing a decision to progress to Phase 2 of search, along with actions representing movement to any adjacent region.
- T_X : The transition function determines which regions have adjacency with each other, thereby determining which actions are valid from every state.
- R_X : The *pmf* term of the reward function is on a per-region basis, with a cumulative probability value representing the expected total number of targets found in a region. The *penalty* term is tuned to represent the time required to travel from the start region to the goal region.

Phase 2 (Intra-Regional Search): The H-MARS algorithm now moves to consider the states within the individual target region as its state space, using the PMF value of states in the reward function for calculating optimal agent actions. In this phase, the state space, action space, and reward function are identical to the MARS algorithm described in Chapter 4.2.3.3 [250]. This

phase, being the lowest level of hierarchy in the H-MARS planner, possesses no minimum reward threshold: whichever action in this phase maximizes expected reward will always be selected.

For each agent X , $MDP_X = (S_X, A_X, T_X, R_X)$:

- S_X : The world features within the state space comprise all individual grid-squares within X 's current region.
- A_X : The action space consists of movement primitives (North, South, East, West, Stay).
- T_X : The transition function determines which actions are legal on a per-state basis (checking for adjacency of grid-squares, and discounting actions that would take the agent into obstacles or out of bounds).
- R_X : The *pmf* term of the reward function is on a per-state basis, with an individual probability value for every grid-square. The *penalty* term is tuned to represent the time required to take a single, atomic action.

Phases 0, 1, and 2 are repeated every time the global PMF updates in response to an accumulation of agent observations. The exact frequency of these PMF updates is adjustable on a domain-by-domain basis.

4.3.4 Hierarchical Guidance Design

We also propose modifying the visual guidance provided by MARS, both to better deal with large, continuous environments, and to leverage the hierarchical phases of H-MARS more effectively. Firstly, rather than rely solely on ground projection of heatmap data, as in [250], we add a minimap for displaying descriptive guidance, allowing human teammates to efficiently ingest spatial data in large, irregular environments with varying terrain heights and incomplete sight-lines. Aside from the change in descriptive guidance form factor, our primary insight is that visualizing an entire PMF in a large environment with numerous states can overwhelm users (Fig. 4.6 Left). We hypothesize that the excessive visual complexity in such guidance would lengthen decision-making



Figure 4.6: Left: Descriptive guidance from the MARS algorithm, coloring every state in an environment according to its probability of containing a goal (dark purple to bright yellow). Right: Descriptive guidance applied to regions rather than individual states. Heatmap coloring is graded to the expected number of goals within each region, rather than individual PMF values.

times and reduce the efficacy of guidance. Additionally, prior research in human navigation shows that humans tend to simplify and scaffold complex environments hierarchically [103, 222].

To address this, we propose a dual visualization strategy. During Phase 1 (Inter-Regional Search) of the H-MARS algorithm, descriptive guidance will consist of a heatmap projected onto the minimap, covering regions rather than individual states, and using color gradation to indicate the expected number of goals in each region (Fig. 4.7 Left). Prescriptive guidance will indicate the target region through a combination highlighting the region on the minimap and projecting an arrow emanating from the user and pointing towards the centroid of the target region.

During both Phase 0 (Local Window Search) and Phase 2 (Intra-Regional Search), descriptive guidance will focus solely on the PMF of individual grid-square states close to the user’s location (Fig. 4.7 Right). The minimap will substantially zoom in to achieve this effect, limiting the information being conveyed to that which is most relevant for immediate decision making, enhancing interpretability of guidance. Prescriptive guidance will highlight the target state and project an arrow emanating from the user and pointing towards the state. By capping the number of states human teammates need to focus on, regardless of algorithmic phase, we expect this new guidance regime to significantly enhance subjective measures of reported guidance quality, as well the speed

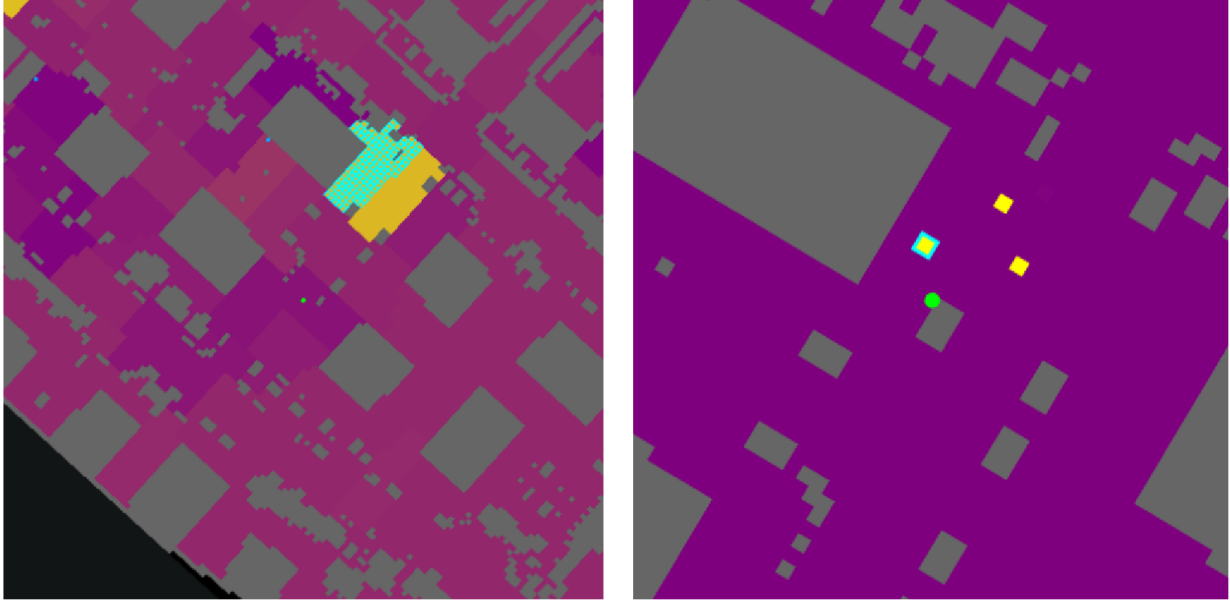


Figure 4.7: Left: Guidance provided by the H-MARS algorithm during Phase 1 (Inter-Regional Search). The human agent is centered in the 2D minimap, represented by a green dot. The drones (blue dots) are recommending the human travel to the high-reward region covered by the cyan lattice. Right: H-MARS guidance during Phase 2 (Intra-Regional Search). Now that the human has approached the target region, and the algorithm has transitioned to considering individual states, the minimap zooms in to show the PMF values of individual grid-squares. The drones are recommending the human travel to the closest of the trio of likely goal states.

of real-time decision-making.

4.3.5 Simulation Testbed

To serve both as demonstration of H-MARS’ usefulness in large, irregular environments, and as a platform for evaluating H-MARS’ algorithmic performance and the effects of its guidance, we present a 3-dimensional simulation testbed environment, involving a timed mine detection and defusing task. Though the overall task objectives are similar to the Minesweeper environment from Chapter 4.2, the environment possesses substantially more realism with respect to target domains. Unlike the discrete, turn-based ‘up, down, left, right’ action space from the Minesweeper domain, this testbed allows participants to move freely and continuously through a cluttered nighttime urban environment (Figs. 4.8 and 4.9). The human’s task is to visually locate and defuse as many hidden mines in the environment as they can within a time limit, a task made more difficult by

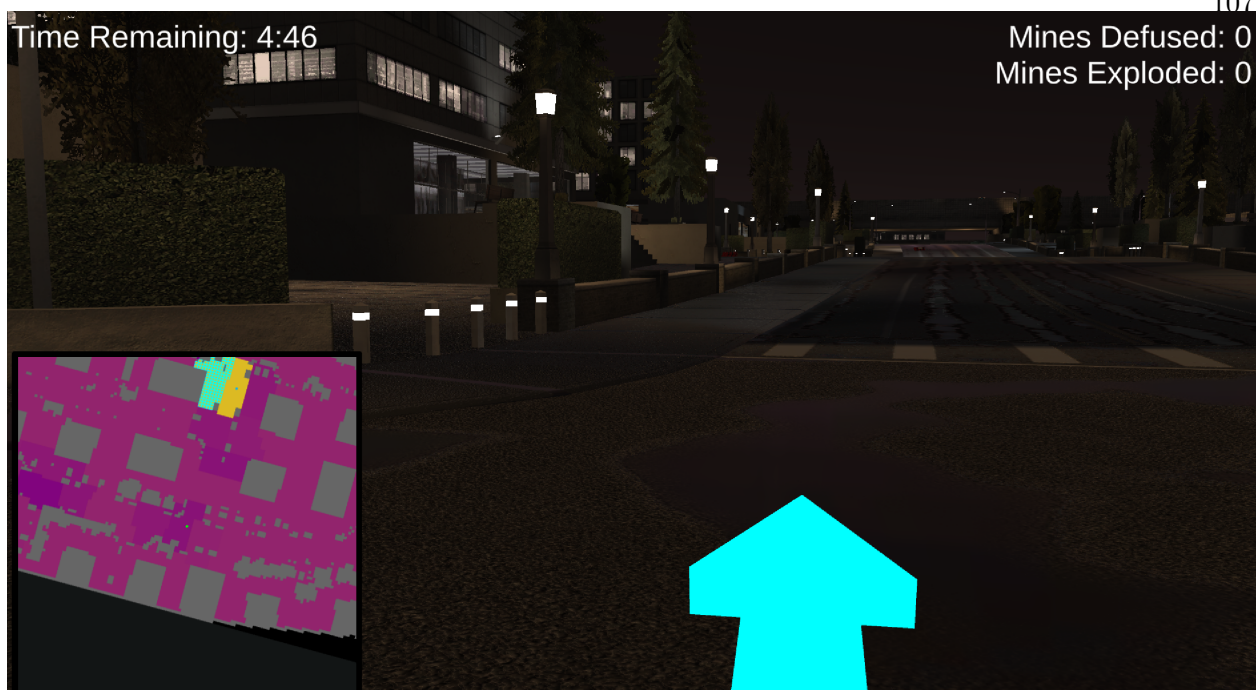


Figure 4.8: H-MARS simulation testbed near the beginning of a 5-minute gameplay round. The player’s drone teammates have found evidence of possible hidden mines in a pair of regions further down the street, appearing as bright yellow on the player’s minimap. The drones provide guidance according to Phase 1 of H-MARS (Inter-Regional Search), recommending the human travel to the region covered by the cyan lattice on the minimap and pointed at by the cyan arrow.

the unfavorable lighting conditions. The human is assisted by a fleet of drone teammates providing explainable guidance via H-MARS to accelerate the process. Like MARS, the drone teammates possess noisy mine detection sensor equipment, though the sensor observations now span over a radius, rather than an individual grid-square.

The testbed domain contains a total of 34,565 fine-grain states, 22,182 of which are traversable by agents. This is significantly more than the 45 states in the Minesweeper domain. Likewise, the total physical area of the testbed domain is approximately 311,000 square meters (or 77 acres), compared to roughly 100 square meters for Minesweeper. This domain size and irregularity is much more representative of the use cases (urban or wilderness search and rescue, explosive ordnance disposal, etc.) we envision for H-MARS.

To serve as a platform for evaluation, hidden mine locations are customizable, as is the human’s starting location, and the starting location and number of drone teammates. The frontend

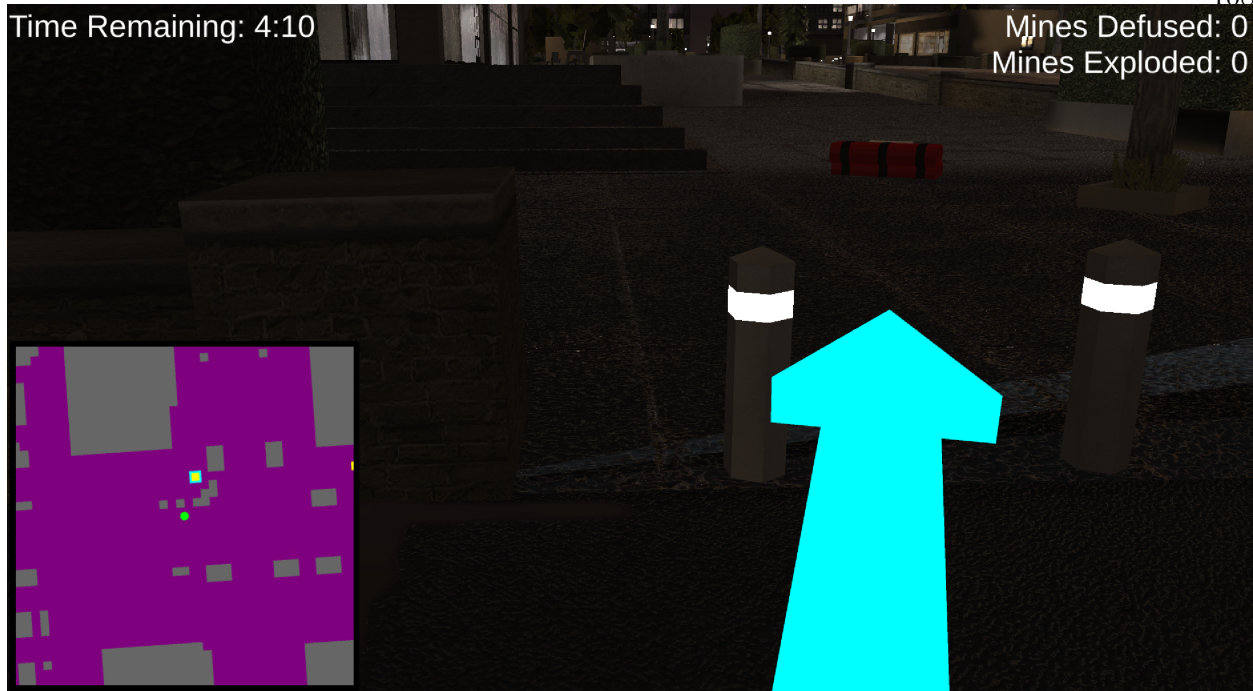


Figure 4.9: H-MARS simulation testbed later in the same gameplay round. The player has approached a region of interest, and the guidance shifts to Phase 2 (Intra-Regional Search) of H-MARS, zooming in the minimap, showcasing individual PMF values, and prescribing the target state (which visibly contains an undefused mine) to the player with a cyan square on the minimap and a cyan arrow.

for the testbed is implemented in Unity, with the H-MARS planner operating off a live-updating 2-dimensional representation of the environment in a backend Python implementation, synchronizing pertinent information (prescriptive and descriptive guidance, and all agent locations and observations) via UDP throughout the gameplay round.

Descriptive guidance is provided in the form of a minimap in the lower-left of the screen, which is zoomed out, overlaying a regional heatmap for guidance during Phase 1 of the H-MARS algorithm (Fig. 4.8), and zooms in for guidance during Phase 0 or Phase 2, showing heatmap colors for each individual grid square (Fig. 4.9). Prescriptive guidance consists of region or state highlighting on the minimap, along with an arrow projected from the player's location pointing towards the recommended target. When the user locates a mine, they must approach within a defuse radius, starting a 10 second timer within which they must hold the defuse key for 3 seconds to successfully defuse it, adding 1 to the player's score for the round. If the player fails to defuse

within that time limit, the mine is marked as unintentionally detonated, leading to a round score of 0.

4.3.6 Algorithmic Performance Evaluation

Using the testbed environment described in Section 4.3.5, we aim to evaluate the efficacy of the H-MARS algorithm against alternative approaches for multi-agent, multi-objective search. We generate seven environments of differing sizes, the largest consisting of the entire simulation testbed environment, and the remaining six representing progressively smaller subsets of the testbed environment, each decreasing in area by 50% until reaching the smallest environment, 1/64 the size of the largest. Every environment maintains the character of the larger testbed, representing a realistic urban setting with only a fraction of the states being traversable. Within each environment, mines are randomly scattered for every trial, numbered to cover roughly 0.3% of the traversable grid-squares. The seven environments are:

- **Environment 1:** 356 traversable states, 2 hidden mines
- **Environment 2:** 717 traversable states, 3 hidden mines
- **Environment 3:** 1527 traversable states, 5 hidden mines
- **Environment 4:** 2970 traversable states, 9 hidden mines
- **Environment 5:** 5949 traversable states, 18 hidden mines
- **Environment 6:** 12742 traversable states, 39 hidden mines
- **Environment 7:** 22182 traversable states, 67 hidden mines

For each trial, the objective is for a simulated human agent (who always complies with the guidance provided to them by their condition’s algorithm) to locate and defuse as many mines as possible within a time limit of 300 seconds, with the help of one or more robot teammates. The simulated human takes 1.5 seconds to traverse from grid-square to grid-square, while the aerial

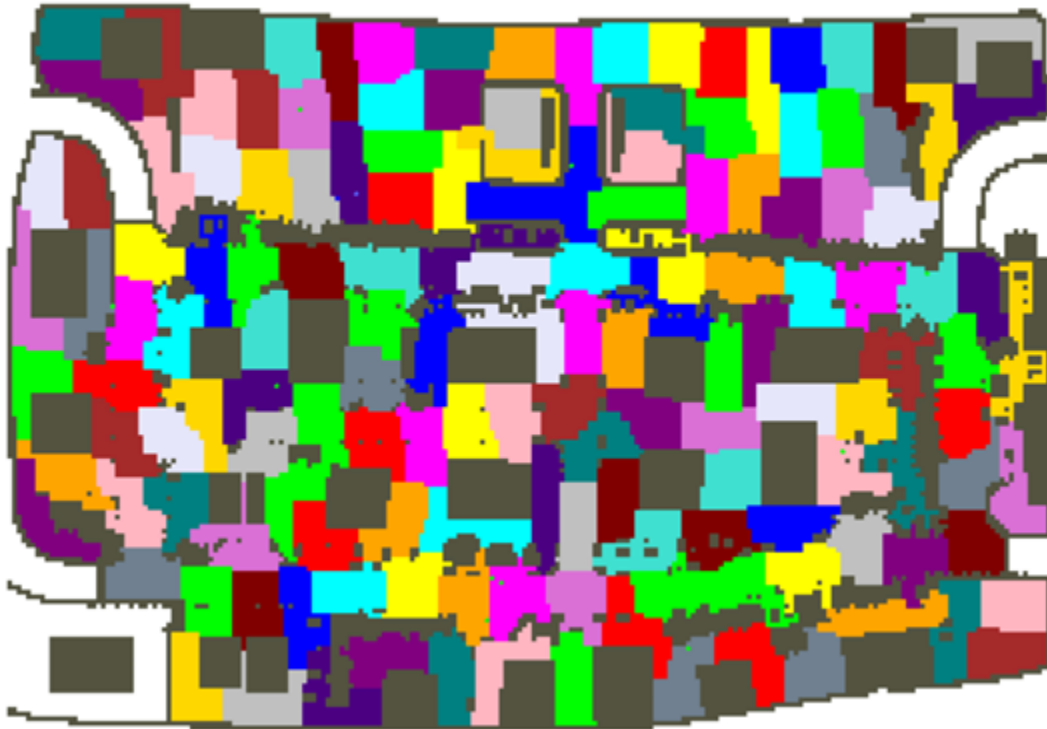


Figure 4.10: Top-down view of Environment 7, covering the entire simulation testbed environment. H-MARS regions are shown in various colors; impassable obstacles are shown in grey. Environments 1-6 are subsets of this map.

robots are faster, taking 0.3 seconds to traverse a single grid-square. The human will always use the most up-to-date guidance available to them: if they receive guidance within their 1.5 second travel window, they will take whichever action is newly prescribed next. If the algorithm is too slow to update, however, they will take another action according to their most recent guidance received, which may reduce the optimality of their next action, including idling in place if the algorithm has not recommended actions far enough in advance.

Four algorithms were tested on these seven environments:

- (1) **H-MARS**: The full H-MARS algorithm as described in Section 4.3.2, with a 2-level hierarchy and a region count roughly equal to the square root of the traversable states.
- (2) **H-MARS (A)**: An ablation of the H-MARS algorithm, removing Phase 0 (edge-case

checking) from the loop, thus removing a source of added complexity and computation. H-MARS (A) starts immediately at Phase 1 for each iteration.

- (3) **Non-RL**: A non-reinforcement learning technique: limited horizon multi-objective A*, as used in [160]. Each time step, every agent performs a search of its local area, identifies a goal state that best balances travel time and expected reward, and paths to it.
- (4) **MARS**: The MARS algorithm as described in [250]. Every state in the environment is considered on every iteration.

Each algorithm is run for 100 trials with randomized mine locations on each environment: 50 trials with 1 human agent and 1 robot agent, and 50 trials with 1 human agent and 5 robot agents. We hypothesize that H-MARS will demonstrate significantly better performance (in terms of percentage of targets found) than MARS, particularly as the number of states in the environment increases, and MARS becomes unable to provide real-time guidance due to computational barriers. We also hypothesize that H-MARS will perform significantly better than the H-MARS (A) ablation condition, showing the performance gains afforded by Phase 0 of the algorithm. Lastly, we hypothesize that the H-MARS algorithm will significantly outperform the Non-RL condition, showcasing how the added computation required to synchronize policies via RL leads to better decision-making capacity.

4.3.6.1 Results

After running 50 trials for each combination of algorithm, environment size, and agent combination, we tested our performance hypotheses using a one-way analysis of variance (ANOVA) over each environment/agent combination, with the algorithm 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 algorithms.

For the 1 human, 1 robot trials, the ANOVA revealed significant effects for the largest 5 out of 7 environments, with no significant differences in mean percentage of targets for Environment 1 and Environment 2, as all four algorithms were successfully able to locate all targets in a vast majority

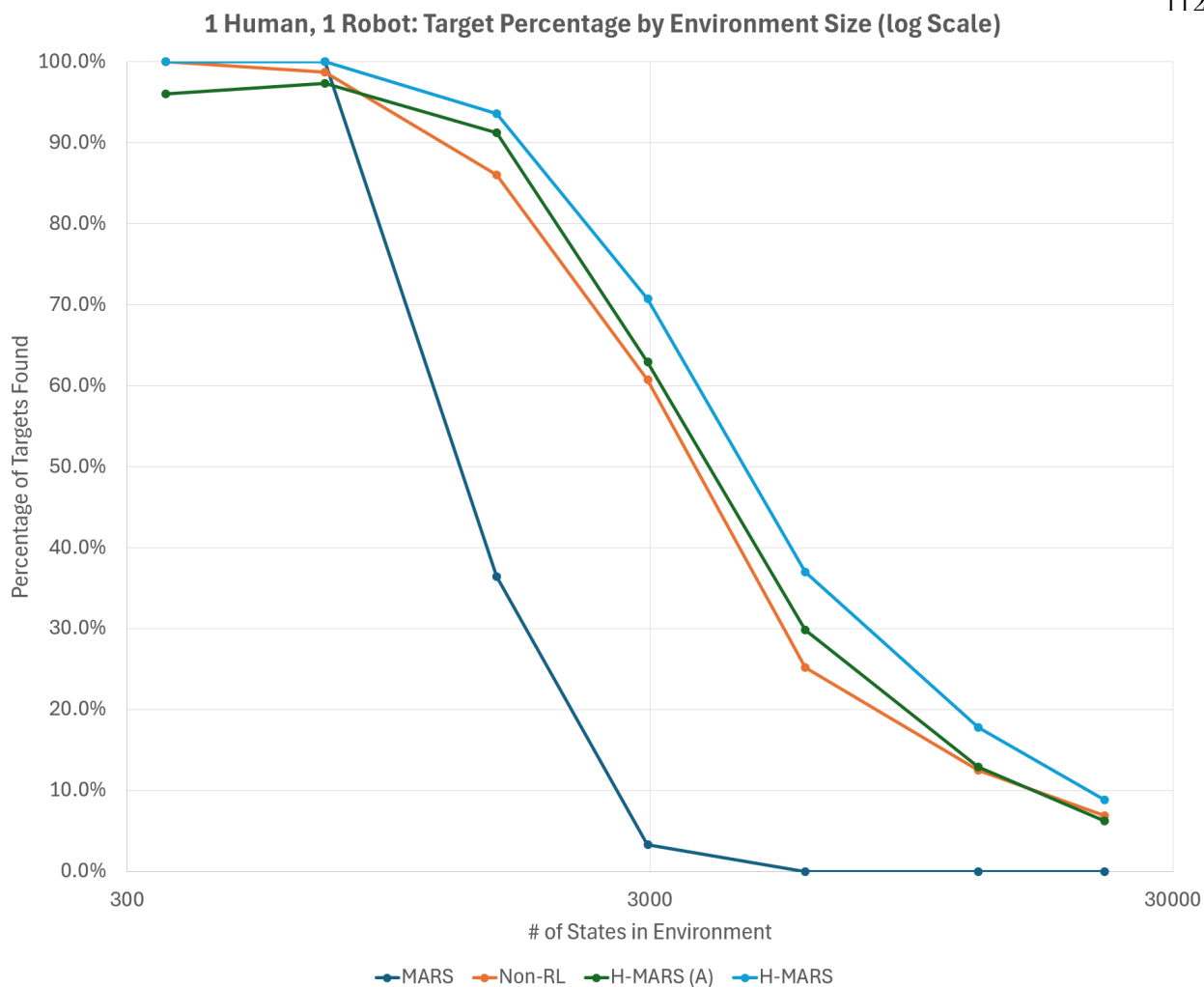


Figure 4.11: Percentage of targets found within 300 seconds, per algorithm, per environment size, using teams comprising of 1 human and 1 robot. The number of states per environment is plotted on a logarithmic scale.

of trials. The mean target percentages for each algorithm on each environment are reported in Table 4.1, as are the differing levels of significance revealed by Tukey's HSD, shown via lettered subscript (distributions with A are significantly higher performance than B, which are higher than C, etc.). The performance per algorithm is also plotted in Fig. 4.11.

Overall, the results conform to expectation. In the smallest environments, where the MARS algorithm is able to comfortably run in real time, performance is indistinguishable. Once the 1,000 state threshold is crossed, MARS begins to struggle to run in real time, showing a precipitous drop from 100.0% in Environment 2 to 36.4% in Environment 3 as the simulated human must rely on

Table 4.1: Percentage of targets found within 300 seconds, per algorithm, per environment size, using teams comprising of 1 human and 1 robot. Environments with ANOVA significance between algorithms are denoted with *. Post-hoc significance using Tukey’s HSD is shown using superscript letters. Individual means denoted by A are significantly higher than those denoted by B or C, and means denoted by B are significantly higher than C, with $p < 0.05$. The mean denoted A/B has significance over C, but no significance over B or under A. The highest of these post-hoc categories for environments with significance is colored green, and the lowest is colored red.

	356 States	717 States	1527 States*	2970 States*	5949 States*	12742 States*	22182 States*
MARS	100.0%	100.0%	36.4% ^B	3.3% ^C	0.0% ^C	0.0% ^C	0.0% ^C
Non-RL	100.0%	98.7%	86.0% ^A	60.7% ^B	25.2% ^B	12.5% ^B	6.9% ^B
H-MARS (A)	96.0%	97.3%	91.2% ^A	62.9% ^{A/B}	29.8% ^B	12.9% ^B	6.2% ^B
H-MARS	100.0%	100.0%	93.6% ^A	70.7% ^A	37.0% ^A	17.8% ^A	8.8% ^A

outdated guidance for a significant fraction of their time.

Towards the upper end of environment sizes (Environments 5, 6, and 7), a clear distinction arises between the highest performing H-MARS algorithm, followed by the middle-performing H-MARS (A) and Non-RL algorithms, which have no significant difference between them, and lastly the MARS algorithm, failing to find a target in any of the simulation runs. In the very largest environments (6 and 7), MARS is unable to run for more than a single iteration within the 300 second time limit of the rounds, due to the overwhelming state count.

For the trials with 1 human and a fleet of 5 robots, the ANOVA again revealed significant effects for Environments 3, 4, 5, 6, and 7, with no significant difference in the smallest two environments. The mean target percentages are reported in Table 4.2, with the same alphabetical scheme for reporting Tukey’s HSD significance. The performance per algorithm is plotted in Fig. 4.12.

In almost every case, the performance of each algorithm in each environment is improved over the 1 human, 1 robot case. This is expected: the added drones are able to multiply the overall information gain per unit time, thus collapsing environmental uncertainty at a faster rate. The primary exception to this is the MARS algorithm. While MARS found 36.4% of targets on average for Environment 3 and 3.3% of targets for Environment 4 in the 1 human, 1 robot setup, those rates dropped to 16.0% and 0.9% respectively for 1 human, 5 robots. This is due to the increased

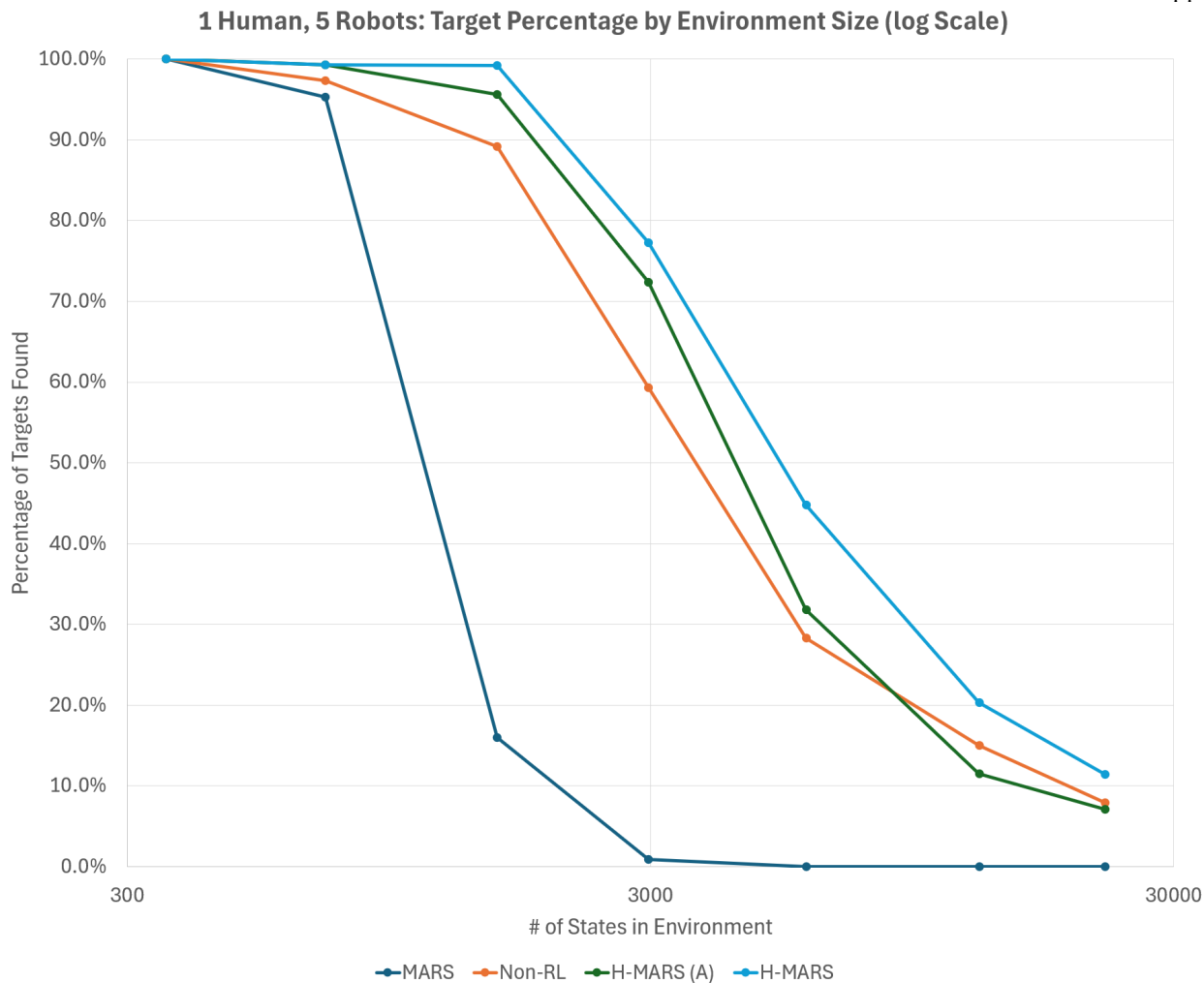


Figure 4.12: Percentage of targets found within 300 seconds, per algorithm, per environment size, using teams comprising of 1 human and 5 robots. The number of states per environment is plotted on a logarithmic scale.

computation load of deciding actions for 4 additional agents. For MARS, which already struggled to provide real-time guidance in these environments, any gains from improved uncertainty reduction were outweighed by the algorithm providing far fewer guidance updates to the simulated human during the 300 second round. The other algorithms, which run far more efficiently, were able to continue providing real-time guidance for all 6 agents, regardless of environment size.

The same pattern as the 1 human, 1 drone case is repeated in the largest environments for the 1 human, 5 robots setup, with H-MARS consistently outperforming all alternatives, and H-MARS (A) and Non-RL occupying a middle ground, where they outperform MARS but underperform

Table 4.2: Percentage of targets found within 300 seconds, per algorithm, per environment size, using teams comprising of 1 human and 5 robots. Environments with ANOVA significance between algorithms are denoted with *. Post-hoc significance using Tukey’s HSD is shown using superscript letters. Individual means denoted by A are significantly higher than those denoted by B, C, or D, means denoted by B are significantly higher than C or D, and means denoted by C are significantly higher than D, with $p < 0.05$. The mean denoted A/B has significance over C, but no significance over B or under A. The highest of these post-hoc categories for environments with significance is colored green, and the lowest is colored red.

	356 States	717 States	1527 States*	2970 States*	5949 States*	12742 States*	22182 States*
MARS	100.0%	95.3%	16.0% ^C	0.9% ^C	0.0% ^C	0.0% ^D	0.0% ^C
Non-RL	100.0%	97.3%	89.2% ^B	59.3% ^B	28.3% ^B	15.0% ^B	7.9% ^B
H-MARS (A)	100.0%	99.3%	95.6% ^{A/B}	72.4% ^A	31.8% ^B	11.5% ^C	7.1% ^B
H-MARS	100.0%	99.3%	99.2% ^A	77.3% ^A	44.8% ^A	20.3% ^A	11.4% ^A

H-MARS.

These results for both team compositions show that H-MARS is superior to MARS in performance for larger environments, showcasing the power of its hierarchical structure for handling numerous states efficiently. Even in smaller environments, H-MARS is always empirically the best performing, or tied for the best performing algorithm. H-MARS also outpaces H-MARS (A) in large environments, showing that Phase 0 of the H-MARS algorithm improves guidance quality over the span of interaction. Likewise, H-MARS outperforms the Non-RL algorithm, suggesting that multiagent reinforcement learning is able to better coordinate its human and robot agents to achieve improved overall team performance.

4.3.7 Guidance-Type Study Design

The results of the algorithmic evaluation show that human teammates derive more utility from complying with H-MARS guidance compared to alternative algorithms, especially as environment state count increases. However, the algorithmic evaluation assumes full compliance on the part of the human, an assumption which is unlikely to hold in reality, as the results from the Minesweeper evaluation of the MARS framework demonstrate [250]. From that human-subjects study, it it also

apparent that guidance type can have a significant effect on human compliance, as well as other behavioral measures such as thinking time, and subjective opinions on the guidance.

To explore these various aspects of human response to H-MARS guidance, we have designed a user study where participants play timed rounds of the mine detection task in the simulation testbed described in Section 4.3.5. The study will be within-subjects, with each participant playing three rounds of the game total, with the type of guidance provided by the drone teammates varying between rounds. This will allow participants to directly compare and contrast the candidate guidance types in a post-experimental questionnaire. To reduce learning effects between experimental rounds, participants will first play a short practice round to familiarize themselves with the game’s rules and controls.

The three guidance-type conditions are:

- (1) **No Heatmap:** The drones make real-time suggestions for where the participant should search by highlighting a target location on the minimap (prescriptive guidance). No heatmap (descriptive guidance) is given.
- (2) **Grid-Square Heatmap:** The drone provides prescriptive guidance as in ‘No Heatmap,’ while also showing an evolving state-wise heatmap, changing as the drones explore the environment, colored from dark purple (low probability of a mine) to bright yellow (high probability), as shown in Fig. 4.6 Left. This guidance is equivalent to the ‘Combined’ condition from the Minesweeper experiment in [250].
- (3) **Hierarchical Heatmap:** The drone provides prescriptive and descriptive guidance, as in ‘Grid Square Heatmap.’ However, the descriptive guidance is hierarchical, tied to the current level of hierarchy in the H-MARS planner, as described in Section 4.3.4. The PMF data switches from being displayed in a zoomed-out, regional heatmap (Fig. 4.7 Left) in Phase 1 of the H-MARS algorithm, and a zoomed-in, state-wise heatmap (Fig. 4.7 Right) in Phases 0 and 2.

The ordering of each condition will be randomized and counterbalanced between participants. As participants play three rounds of the game, mine locations will also be randomized between three environment seedings to minimize difficulty effects from a specific distribution of mine locations.

In addition to game performance and behavioral measures collected from gameplay (idle time, movement distance, average distance to recommended target), we will administer a brief survey following every round, consisting primarily of Likert-scale questions assessing workload, trust in the system, system usability, and system interpretability. Following completion of all rounds, a final post-experiment survey will be administered, consisting of a combination of direct comparison and open-ended questions.

We hypothesize that the two conditions containing descriptive guidance (‘Grid-Square Heatmap’ and ‘Hierarchical Heatmap’ will lead to improved subjective rankings and better task performance compared to the ‘No Heatmap’ condition). This would align with the findings from [250], comparing a combined guidance type with prescriptive guidance. Along similar lines, we expect the ‘No Heatmap’ condition to be associated with lesser idle time and behavior that indicates greater compliance (lower average distance to goal), again aligning with [250].

Among the two conditions containing descriptive guidance, however, we expect ‘Hierarchical Heatmap’ to perform the best on subjective ratings of guidance usefulness, interpretability, trust, and workload, as participants will find the hierarchical guidance much easier to understand and follow than the highly detailed state-wise PMF presented in ‘Grid-Square Heatmap.’ Likewise, we expect ‘Hierarchical Heatmap’ to outperform ‘Grid-Square Heatmap’ in objective performance, as well as in behavioral measures associated with compliance.

4.3.8 Conclusion

In this work, we presented H-MARS, a multi-agent reinforcement learning algorithm for coordinating robotic agent actions and providing explainable decision support to improve human performance on multi-objective search tasks in large, realistic environments. Through leveraging spatial hierarchy, H-MARS is capable of providing useful guidance in large, continuous state

spaces, operating at multiple levels of abstraction. We also designed a novel form of robot-provided guidance, utilizing the hierarchical nature of the algorithm to limit the environment-based visual explanations given at any time to those which are useful for the current phase of search, aimed at improving interpretability and achieving better mental model alignment.

We introduced a large, urban simulation environment to serve as a testbed for H-MARS, demonstrating the algorithm’s utility for real-world multi-objective search tasks, such as explosive ordnance disposal. Using this testbed, we evaluated the performance of the H-MARS algorithm against alternative techniques for multi-agent coordination with a simulated human agent, demonstrating that, assuming human compliance with guidance, H-MARS leads to improved performance, especially as the state count of the environment increases. Lastly we describe an evaluation design for measuring the human-centric effects of the novel H-MARS guidance, including subjective ratings of trust and interpretability, compliance, and resultant performance.

Chapter 5

Spatially-Grounded Justifications for Robotic Decision Support

5.1 Motivation

This chapter directly builds on the algorithms and interfaces described in Chapter 4, again focusing on providing explainable guidance in human-robot search tasks. Section 4.2.6.2 describes a set of takeaways derived from evaluating the MARS framework with human participants [250]. Among these, **T4** recounts that participants often expressed a desire for explanation whenever their recommended path changed suddenly or drastically. These path changes were the result of shifting probability distributions as the drone teammates gathered data and improved their guidance, but they were viewed as confusing and untrustworthy by participants, indicating a mismatch in expectation.

This inspires the work in this chapter, which introduces a method for identifying periods of likely expectation mismatch between human and robot teammates, thus triggering a spatially grounded natural language justification. After validating this trigger methodology, we test multiple candidate justification templates in a collaborative treasure hunt environment. These templates are created by varying the basis of a provided counterfactual justification between environment-centric and policy-centric explanations, as well as varying the scope of that justification between local and global contexts. Through a human-subjects study, we find that environment-centric justifications are rated as most trustworthy, helpful, and interpretable, but that policy-centric justifications lead to the highest rates of human compliance. We use these results to present a taxonomy of situations where each justification type is likely to be useful in human-robot teaming domains. This work was

presented at Robotics: Science and Systems (**RSS 2023**) [160].

This chapter is unique among the works presented in this thesis in two key aspects: first, it is the only work to consider the question of “when” communication should be delivered to make the best use of limited human attention, contrasting with the “always on” visuals of the interfaces in prior chapters. This requires the robot teammate to explicitly identify when human and robot models diverge, a task that is implicitly left to the human in the systems of prior chapters. Second, in this chapter, the spatially-grounded communication under discussion is not wholly reliant on visualization. In this work, each of the designed justification types have a visual component showcasing features in environmental context, along with a natural language/numerical component which references those features in a counterfactual style, thus attempting to justify a robot’s updated decision.

Much like the work in Chapter 4, this chapter involves explicit transfer of knowledge from a noisily competent robot teammate to a human teammate. Interestingly, the contrasting results between environment-centric and policy-centric justifications in this chapter’s user study hint at possible techniques for delivering autonomous justifications in domains with comparatively lower robot competence in a way that promotes more active decision-making effort from human teammates. In such domains, additionally providing methods for human-to-robot communication to help close the informational loop could lead to better teaming outcomes. However, this concept has yet to be explored and is left as a direction for future work.

5.2 Autonomous Justification for Enabling Explainable Decision Support in Human-Robot Teaming

5.2.1 Introduction

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 [270, 105, 29]. Various xAI techniques facilitate better transparency into collaborative robots’

choices, improving trust, interpretability, and user acceptance [36, 247, 68, 163]. However, if explanations are given at inopportune times with poor context, they can produce the opposite effect [118]. Furthermore, different explanation content can have differing effects on a human collaborator’s mental model, which can impact their behavior [35, 159]. 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 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 [229, 112]. 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 [250, 246, 36, 241]. 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 [250]. A justification (see Fig. 5.1 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**

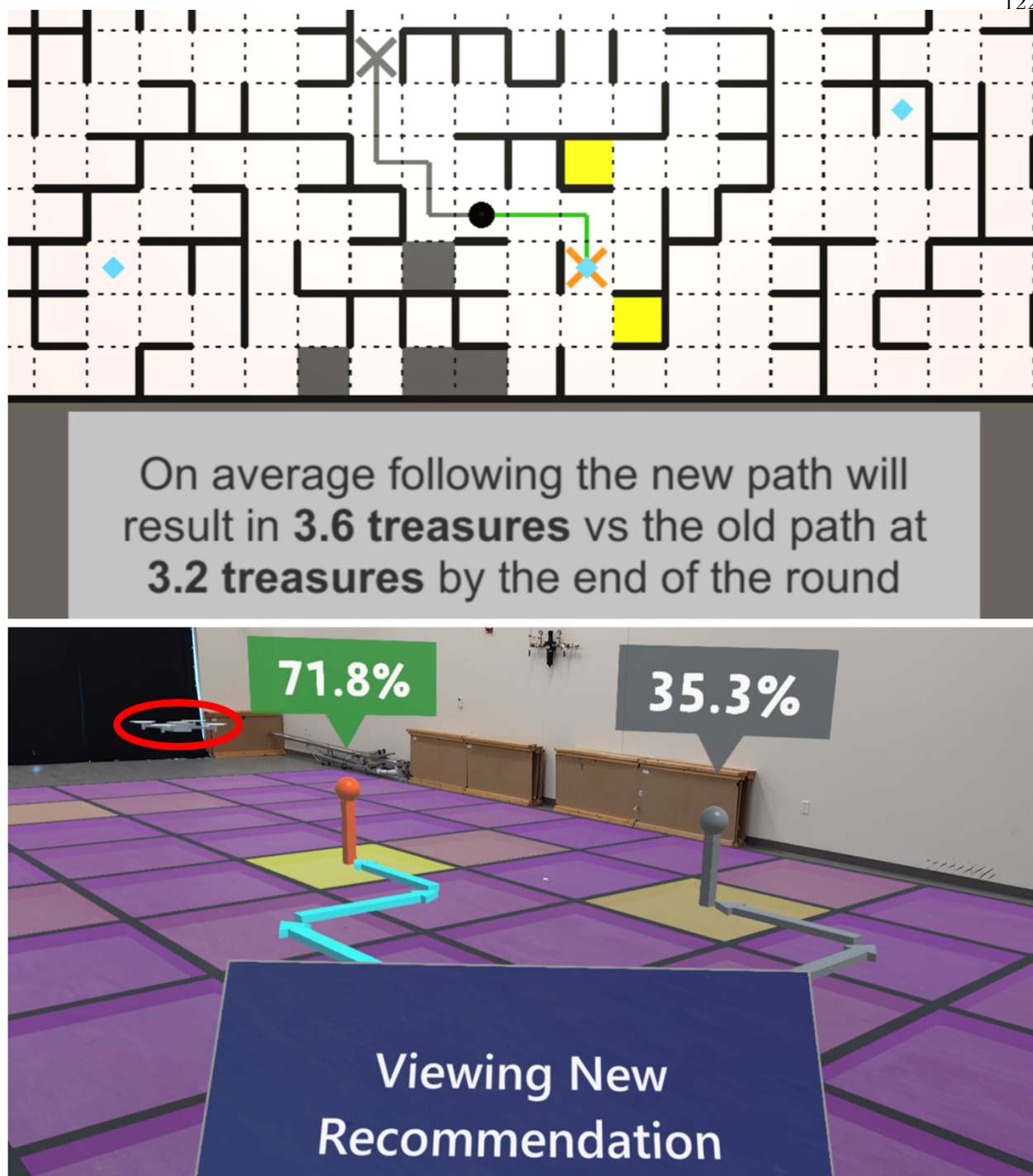


Figure 5.1: **Top:** a counterfactual policy-based justification provided by drones (blue diamonds) to the human in a collaborative 2D treasure hunting game. **Bottom:** a counterfactual environment-based justification showing the relative percentages of finding a target, provided by a drone (circled 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.

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.

5.2.2 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 [227, 249, 270]. This is particularly relevant in the context of model reconciliation, where mismatches in expectations can lead to catastrophic failures [36, 35]. 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 [114, 205, 270].

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 [283, 135]. Robots with this explanation-providing capability are generally perceived to be more helpful and transparent [246]. Conlon et al. [48] 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 [242]. 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 [161], model-agnostic methods like LIME [205], and visualizations like Grad-CAM [226].

Explainable behavior methods attempt to make the intentions of robotic agents clearer to humans by improving metrics like explicability [141], predictability [34], or legibility [70]. Research has demonstrated that people dislike inexplicable behavior from robots, rating it as frustrating, and leading to mistrust of the robot [270, 6]. Robot behavior that attempts to align itself with human expectations often must sacrifice optimality to achieve high explicability. In Tabrez et al. [250], 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. [52] 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 [72, 246]. In this work, we introduce and analyze different types of justifications aimed at addressing both of those goals.

5.2.3 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. [250], 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 $R_r : S_r \times A_r \times S_r \rightarrow \mathbb{R}$.
- 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 5.2.5.1). This relationship can be seen in Fig. 5.2.

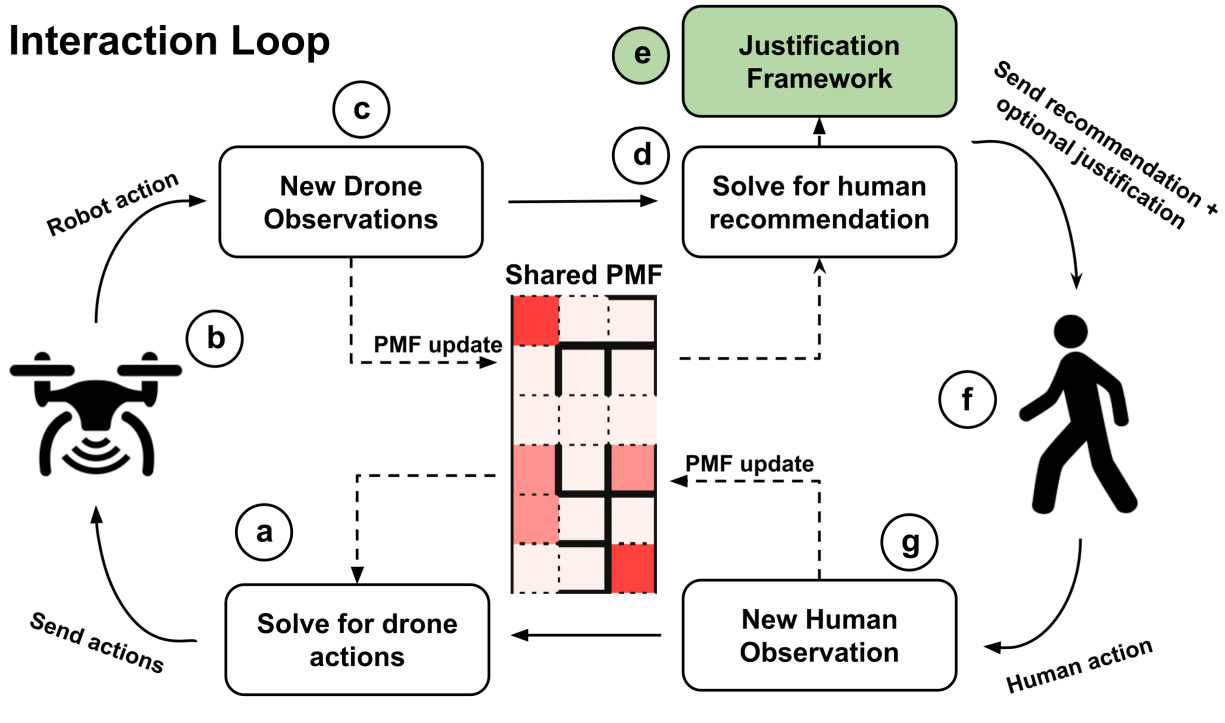


Figure 5.2: The loop describing the human-drone interaction with shared PMF in our domain. The Justification Framework, the primary contribution of this work, is highlighted in green.

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 [250]. Fig. 5.2 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 5.2.4 and 5.2.5. The human’s 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).

5.2.4 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.

5.2.4.1 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 [118]. 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 [67]. In most adaptive autonomy use cases, the second criterion is rarely met, especially in high uncertainty environments [210, 250]. Prior research has found that whenever there is a misalignment of expectations between human and autonomous teammates, explanations are expected to be provided [250, 35]. 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 [36]. Trust and reliance in these systems deteriorate when they lack the capability to justify their recommendations in the presence of such mismatches [52]. 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 [28]. Justifying too frequently can lead to habituation, as repeated explanations reduce user responsiveness to them [257, 100, 101, 126]. Thus, always justifying is also undesirable.

5.2.4.2 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 [121] 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 [132].

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 [28].

Justification Framework: Using the human MDP M_h described in Section 5.2.3, 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 ‘ $\widehat{\quad}$ ’ 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:

$$\mathcal{D} = |\mathbb{E}_{\pi_h^*,s}(R_h) - \mathbb{E}_{\widehat{\pi}_h^*,s}(\widehat{R}_h)| \quad (5.1)$$

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:

$$\mathcal{T} = \text{dist}(\widehat{\psi}_h - \psi_h) \quad (5.2)$$

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:

$$V(\mathcal{J}) = \max \begin{cases} \alpha * \mathcal{D} \\ \beta * \mathcal{T} \\ \gamma * \mathcal{D} + \kappa * \mathcal{T} \end{cases} \quad (5.3)$$

α , β , γ , and κ are tunable hyper-parameters. The first component of Eq. 5.3 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 be tuned separately per domain. A justification should only be triggered if the expected benefit to the user is higher than the justification cost.

$$V(\mathcal{J}) - C(\mathcal{J}) > 0 \quad (5.4)$$

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 [246, 212]. 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. 5.2 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 [250, 177]. With that in mind, we weight differences higher the closer they are to the human’s current location.

$$\mathcal{T} = \sum_{s' \in \psi_h \oplus \widehat{\psi}_h} \gamma^{d(s', s_h)} \quad (5.5)$$

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. 5.1, and tune the relevant hyperparameters in Eq. 5.3 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.

5.2.4.3 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. 5.1-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. 5.1-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 [56].

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 5.1, 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.

Table 5.1: Means and standard deviations of rated usefulness of justification timing (on scale of 1-5) per timing strategy.

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

5.2.5 Justification Framework: Content

In this section, we investigate the **content** of effective justification. Drawing from previous works in explainable AI [218, 205, 5], 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 ? [265]” 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 [96, 41, 252].

The following four proposed types of features used in a justification vary along a spectrum of interpretability and comprehension for its users [66].

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 [235], and tend to prioritize short-term rewards in their own reasoning (e.g., Stanford marshmallow experiment [177]). 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.

5.2.5.1 Framing Justification 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 [85, 280, 281]. 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

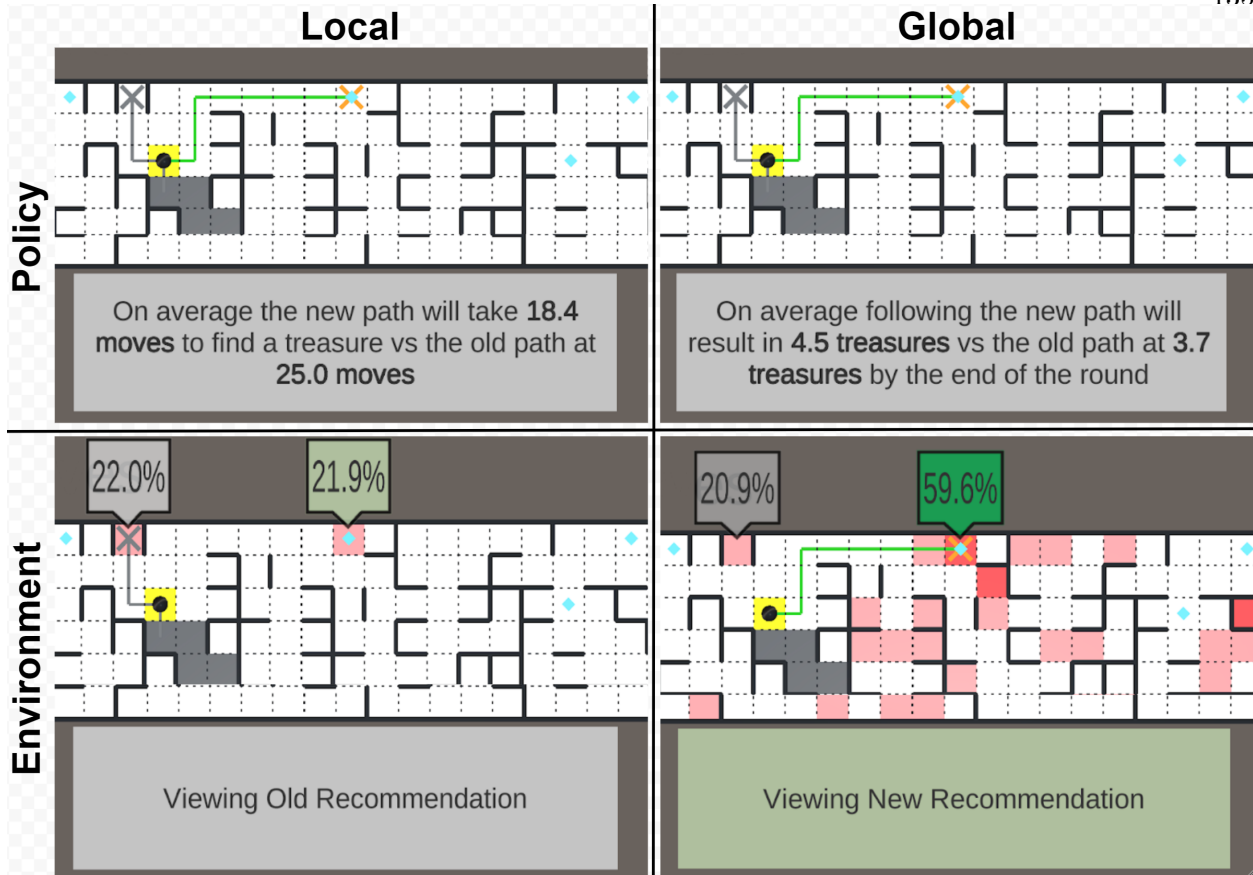


Figure 5.3: The four types of characterized justifications, given during the same gameplay scenario in the treasure hunt domain. Note that the percentages shown on the map in both environment-based justifications involve alternating visually between the old and new probabilities every 1.5 seconds. For simplicity, only the old probabilities are shown for ‘environment local’ and the new probabilities for ‘environment global’ in this figure.

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. 5.3 shows how these four justification types were mapped to our treasure-hunt domain.

5.2.5.2 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 [199].

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 [250].

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 jus-

tifications.

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.

5.2.6 Experimental Evaluation

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

5.2.6.1 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 5.2.3 (Fig. 5.1-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 5.2.4 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).

5.2.6.2 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

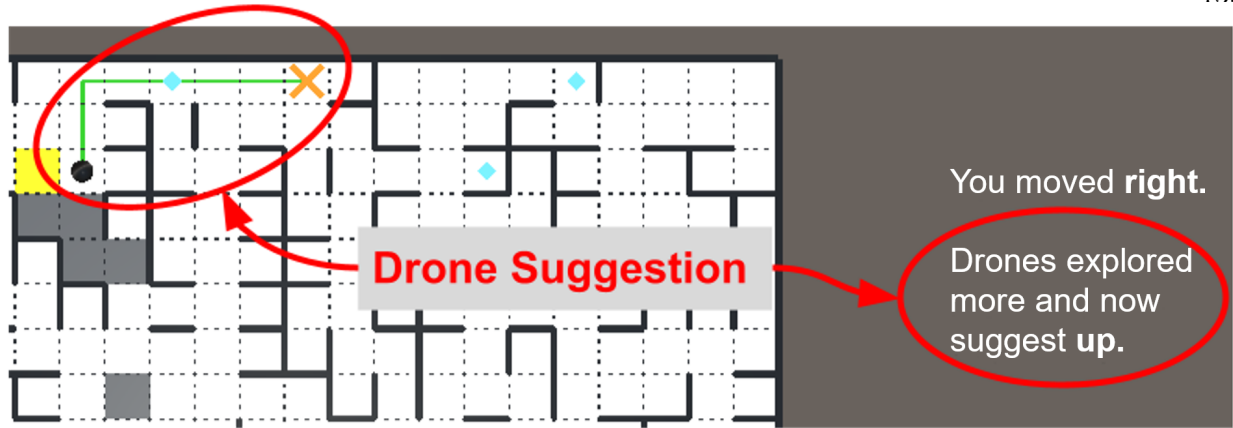


Figure 5.4: Drone guidance is shown as a path overlay and a textual representation of the next suggested move.

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. 5.4), 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. 5.3).

5.2.6.3 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.

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.

5.2.6.4 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

Table 5.2: 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 Interpretability (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)

metrics from the fields of robotics and explainable AI, including the Trust in Automation Survey [127], the Interpretability and Decision-Making Surveys for XAI metrics [270, 120, 231], Stress and Workload (NASA-TLX) [110], and Perceived Intelligence (Godspeed Questionnaire) [18]. 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 [119] which resulted in the scales presented in table 5.2.

5.2.7 Results

5.2.7.1 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 5.3 and 5.4 below.

Table 5.3: 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.

	Global Pol.	Local Pol.	Global Env.	Local Env.	None
Compliance Rate*	84.67% ^A	81.53%	70.65% ^B	75.48%	70.53% ^B
Compliance Rate (While Justifying)*	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 (While Justifying)*	1.74s ^B	1.66s ^B	2.49s	3.39s ^A	1.85s ^B

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

Table 5.4: 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.

	Policy Features	Env. Features	None
Compliance Rate*	83.14% ^A	73.00% ^B	70.53% ^B
Compliance Rate (While Justifying)*	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 (While Justifying)*	1.70s ^B	2.93s ^A	1.85s ^B

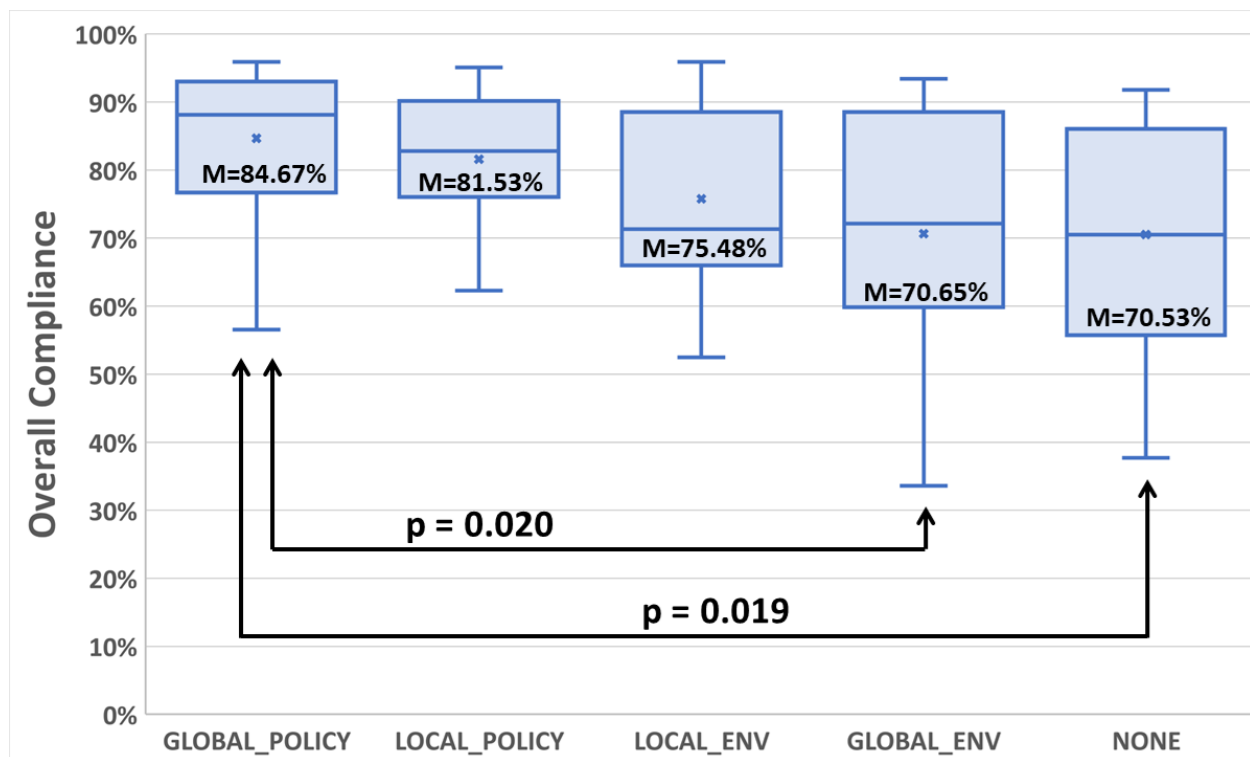


Figure 5.5: Compliance rate by condition, with means and post-hoc significance shown.

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' ($p = 0.047$).

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)**.

5.2.7.2 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-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.

Table 5.5: 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.

	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

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

Table 5.6: 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.

	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

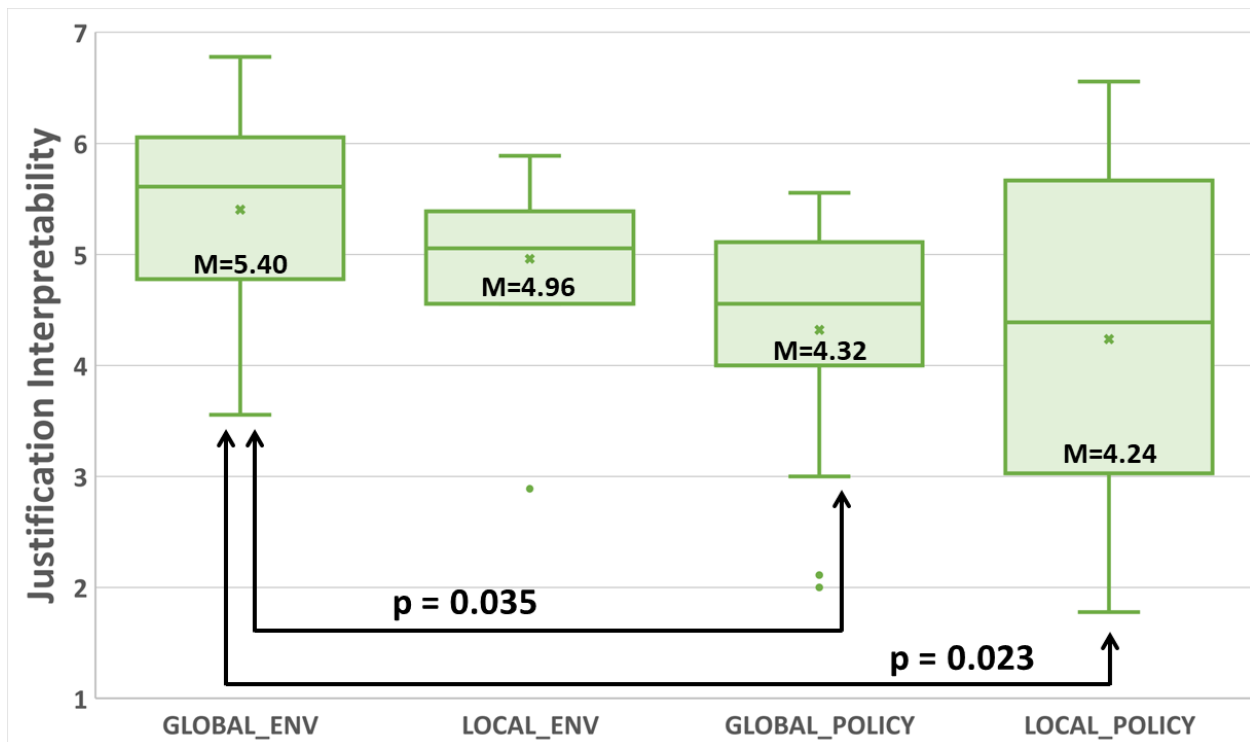


Figure 5.6: Rated interpretability of justifications by condition, with means and post-hoc significance shown.

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)**.

5.2.8 Recommendations & Potential Applications

5.2.8.1 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.

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 [130] where the human cedes effective control of decision-making entirely to the robot agent. In cases of low robot competence, this

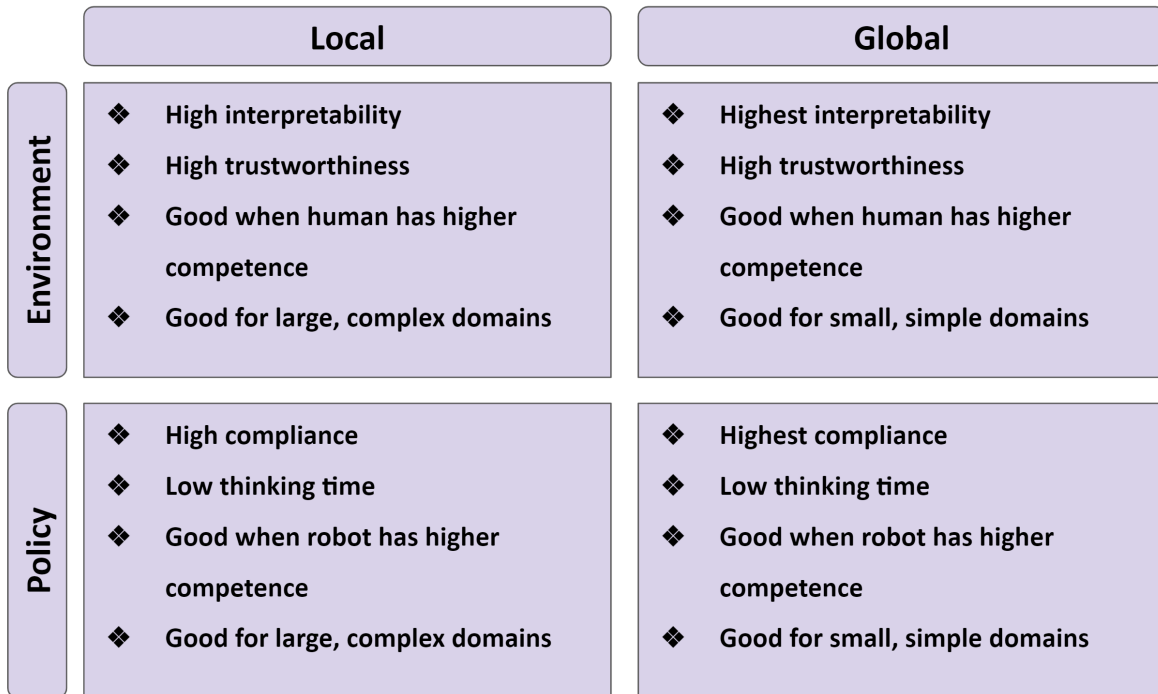


Figure 5.7: A taxonomy of the usefulness of each justification type.

would lead to a large number of Type I errors where users accept low-quality advice from the system [92, 65].

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 [58, 156]. Policy-based justification can also be suitable when the human needs to make snap decisions in time-critical situations.

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 [130]. 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 [250]. 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. 5.7.

5.2.8.2 Potential Application: AR-based Spatial Navigation

To illustrate the application of these synthesized justification design principles, we present a concept of how they might be implemented in a real-world decision support system embedded in an augmented reality (AR) interface (similar to Tabrez et al. [250]). 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. 5.7 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 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. 5.8 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. 5.8 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



Figure 5.8: Top: AR-based policy justification. Bottom: AR-based environment justification.

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. 5.8 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.

5.2.9 Conclusion

In this work, 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 synthesized 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 6

Conclusion and Future Directions

In this thesis, I presented a collection of systems, algorithms, and interfaces capable of enabling mental model alignment between human and robotic agents, thus enhancing the performance and fluency of human-robot teams. To accomplish this, each work leveraged spatially-grounded communication, using a combination of augmented reality-based visualization and natural language to transfer task and environment knowledge between agents. The works presented cover a wide range of scenarios, including collaboration with manipulator robots and mobile robots.

These works also showcase the ability of spatially-grounded communication to facilitate bidirectional alignment, with expert humans improving robot mental models and resultant performance in ARC-LfD, and expert robots improving human mental models and performance in the MARS series of algorithms. The various evaluations presented in this thesis serve to highlight the versatility of spatially-grounded communication techniques for aligning mental models between agents, and improving objective and subjective measures of team performance across a wide range of target human-robot domains.

6.1 Summary of Individual Contributions

In the prior chapters of this thesis, I introduced and evaluated:

- A novel augmented reality interface enabling humans to teach new skills to robots via learning from demonstration, visualize those learned skills in environmental context, and adapt those skills to changing environments and task setups.

- A novel technique for autonomously organizing shared workspaces, including the projection of “virtual obstacles” in augmented reality, such that human motions and goals are made more predictable, enhancing fluency of collaboration.
- A novel augmented reality interface for visually indicating floor ownership between human and robotic agents, negotiated using human and robot requests, thus enabling close-proximity collaboration.
- A novel multi-agent algorithm for multi-objective search tasks in uncertain environments, capable of simultaneously commanding robotic agents and providing explainable guidance to human teammates, delivered via augmented reality.
- A novel extension to the aforementioned algorithm, leveraging spatial hierarchy to reason, plan, and provide guidance at multiple levels of granularity according to the current phase of search, allowing for real-time use in large, irregular environments.
- A novel technique for autonomously generating and timing multi-modal justifications of robot-provided guidance during periods of mismatched expectation, enhancing human understanding of robot decision-making and improving compliance with guidance.

6.2 Directions for Future Work

In this final section, I describe a selection of topics for future exploration, related to and building upon the themes of thesis:

6.2.1 Cooperative Reward Design

Many of the works in this thesis involve human agents communicating to improve robot performance or robot agents communicating to improve human performance. In Chapter 2, the ARC-LfD interface facilitates human injection of constraint information into robot learning, capturing high-level task knowledge previously unknown to the robot and improving robot performance.

In Chapters 4 and 5, the MARS series of algorithms provide live and explainable guidance to human teammates, providing them with information and recommendations that allow them to reach task objectives quicker.

For a truly collaborative system, it would be advantageous to have both directions of improvement captured, rather than one or the other. Many potential domains for human-robot teaming will contain bidirectional information imbalances: robot teammates will possess task or environment knowledge unknown to the human, and vice-versa. By designing communication interfaces to facilitate this bidirectional exchange, allowing all agents to shape shared notions of environmental uncertainty and reward, we would enable iterative mental model refinement as tasks progress.

Take for example the MARS family of algorithms [250, 160, 251]. It's clear how robots provide benefits to a human teammate, collapsing environmental uncertainty as they explore and communicating what they've found back to the human. However, the system has no mechanism for humans to influence robot actions or search patterns. This means humans are unable to inject their own domain expertise or intuition into the collaborative problem. For real-world problems we are interested in (such as search and rescue, explosive ordnance disposal, or radiological device recovery), humans will indeed have a degree of training and experience that would be likely to give them knowledge difficult to model precisely in a robot planner. Robots could waste substantial amounts of time exploring areas of the environment that a human expert knows would be unproductive.

MARS uses a shared probability mass function (PMF) to represent evolving environmental uncertainty. While the AR and minimap interfaces presented in Chapter 4.2 and 4.3 respectively are capable of displaying this information to human teammates, such interfaces could also include mechanisms for humans to give spatially-anchored feedback to shape this shared reward notion. By developing interfaces that allow humans to directly influence the shared PMF, pushing the probability of target regions up or down, the robots will gain a new information source, extending beyond that which is provided by their own sensors.

One promising technique for accomplishing this lies in spatial sketching-type interfaces, like the interface presented by Ahmed et al. [2], allowing humans to inject 'soft data' into collaborative

tasks. Such an interface could be implemented in augmented reality, or utilize a standalone tablet. In either form factor, the human would visually denote regions of the PMF through sketching that they desire to have higher or lower probability. For example, if there is a region of a shared environment a human expert knows is highly unlikely to contain a target, they would reduce the probability mass in that region, incentivizing robot agents to explore other regions instead. The opposite action, raising a region's probability, would cause the provided autonomous decision support to be drawn towards that region. By allowing for cooperative reward design, environmental uncertainty could be collapsed quicker, leading to quicker convergence on targets: a beneficial result for time-sensitive domains.

6.2.2 Remedial Actions for Low Robot Confidence

In many domains, autonomous robotic agents will experience varying levels of competence and certainty throughout the duration of a task. This was plainly observed in the Minesweeper domain from Chapter 4.2 and the treasure hunt domain from Chapter 5.2. At the beginning of each experimental round, when the robot possessed no knowledge about the environment and the PMF was uniformly distributed, the guidance provided by the system was of limited usefulness. But as the robots gained confidence in the location of possible targets, the guidance became much more actionable, leading participants to higher scores when they complied with it.

In many cases, it is not obvious to human teammates what the comparative quality of robot policies or decision support is. In fact, unless a robot has an explicit mechanism for assessing self-confidence in its decision-making, it is not obvious to robot teammates either. The robot is simply choosing an action that maximizes expected reward. This raises an interesting research question: can a robot teammate improve performance by identifying whether itself or a human teammate is more likely to make better decisions at a specific point in interaction, adjusting its actions or guidance accordingly to shift decision-making authority?

Prior work has shown the validity of using Q-ledger methods (storing buffers of Q-function values in reinforcement learning problems to test for Q-function stability over time) as a proxy for

robot self-confidence [55]. This confidence can be used, in combination with a technique like value of information theory [121], to decide when it's valuable to consider remedial actions that involve a human teammate.

If the robot is generating a policy for itself, it can improve its own performance during periods of low confidence by soliciting human feedback, in degrees of involvement up to and including full handover of control to a human to act as a teleoperator. This feedback or teleoperation would continue until the robot has entered a higher-confidence state, with the value of human involvement no longer surpassing the cost of distracting the human teammate from their own tasks. For such a system soliciting human intervention, there are additional open research questions on how best to prime human teammates so they are best able to quickly establish situational awareness and provide quality guidance, thus reducing their overall workload.

If the robot is instead generating a policy for a human teammate (i.e., providing decision support), human performance can be improved during periods of low robot competence by introducing deliberate friction into the agent-to-agent communication [194]. The aim of such friction would be to pull the human out of a high-compliance, Type I style of thinking, encouraging them to instead spend more mental effort assessing the situation and scrutinizing the guidance, deciding for themselves what the best course of action is. By developing a framework deciding when and how humans should assist robots with variable degrees of confidence, we can expect not only to improve team performance, but also enhance safety, deferring to humans during out-of-distribution scenarios to avoid potentially dangerous robot actions.

6.2.3 Leveraging Psychology for Improved Human Modeling

There exists substantial amounts of prior work, across wide-ranging fields, that assumes humans behave with perfect rationality, modeling them as such. Boltzmann rationality improves on this rationality assumption by adding Gaussian noise around an optimal human action [192, 191]. However, this alone does not accurately capture how humans approach decision-making. In the real-world, humans come pre-installed with a wide variety of cognitive biases; mental heuristics

arising from evolution and measurable by experiment. These mental shortcuts enable us to make quick decisions in a complex world at the cost of rationality, a concept called ‘satisficing’ [234].

The work in this thesis which relies on modeling human decisions, namely for determining mental model divergence and initiating justifications to repair discrepancies in Chapter 5, does so by treating the human as a reinforcement learning agent who has a rational, well-formed policy, but an incomplete knowledge of reward. This human model is inspired by prior work by Tabrez et al. [246], and functions empirically well in the treasure hunt domain from Chapter 5, which is characterized by large quantities of robot knowledge which are unknowable by the human, except through deliberate communication. However, related research has found success in modeling humans with known patterns of suboptimal decision making, such as Kwon et al.’s use of cumulative prospect theory from behavioral economics to inform human modeling for HRI tasks [143].

Another dimension to consider when modeling human decision making is differences in personality. In many of the studies described in this thesis, participants displayed highly variable strategies, and variable compliance in the face of robot-provided guidance. Prior work in HRI has demonstrated links between participant personality and style of interaction [223, 224], suggesting that such differences could be in part the result of personality effects. If a human teammate’s personality was known ahead of time and a personality-aware human model was created, robot communication strategy could be adjusted between users, to improve compliance, understanding, or other desiderata. By leveraging psychological concepts such as suboptimality stemming from cognitive bias or personality differences, we can improve upon the one size fits all approach to human modeling, enhancing both team performance, and subjective measures of team function.

Bibliography

- [1] 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.
- [2] 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.
- [3] Baris Akgun, Maya Cakmak, Karl Jiang, and Andrea L Thomaz. Keyframe-based learning from demonstration. International Journal of Social Robotics, 4(4):343–355, 2012.
- [4] Baris Akgun, Kaushik Subramanian, and Andrea Lockerd Thomaz. Novel interaction strategies for learning from teleoperation. 2012 AAAI Fall Symposium: Robots Learning Interactively from Human Teachers, 12, 2012.
- [5] 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.
- [6] 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.
- [7] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. Robotics and Autonomous Systems, 57(5):469–483, 2009.
- [8] 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.
- [9] 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.

- [10] Christopher G Atkeson and Stefan Schaal. Robot learning from demonstration. In ICML, volume 97, pages 12–20. Citeseer, 1997.
- [11] 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.
- [12] Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. Action understanding as inverse planning. Cognition, 113(3):329–349, 2009.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] Shray Bansal, Rhys Newbury, Wesley Chan, Akansel Cosgun, Aimee Allen, Dana Kulić, Tom Drummond, and Charles Isbell. Supportive actions for manipulation in human-robot coworker teams. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 11261–11267. IEEE, 2020.
- [17] 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.
- [18] 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.
- [19] Felix Berkenkamp, Andreas Krause, and Angela P Schoellig. Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. arXiv preprint arXiv:1602.04450, 2016.
- [20] 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.
- [21] 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.
- [22] Serena Booth, Yilun Zhou, Ankit Shah, and Julie Shah. Bayes-trex: a bayesian sampling approach to model transparency by example. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 11423–11432, 2021.

- [23] 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.
- [24] 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.
- [25] Connor Brooks and Daniel Szafrir. Building second-order mental models for human-robot interaction. arXiv preprint arXiv:1909.06508, 2019.
- [26] Connor Brooks and Daniel Szafrir. Visualization of intended assistance for acceptance of shared control. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020.
- [27] Judith Bütepage, Hedvig Kjellström, and Danica Kragic. Anticipating many futures: On-line human motion prediction and generation for human-robot interaction. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 4563–4570, 2018.
- [28] Maya Cakmak and Andrea L Thomaz. Designing robot learners that ask good questions. In 2012 ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 17–24. IEEE, 2012.
- [29] 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.
- [30] Yair Censor. Pareto optimality in multiobjective problems. Applied Mathematics and Optimization, 4(1):41–59, 1977.
- [31] Sonia Mary Chacko and Vikram Kapila. An augmented reality interface for human-robot interaction in unconstrained environments. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3222–3228. IEEE, 2019.
- [32] 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.
- [33] Tathagata Chakraborti, Subbarao Kambhampati, Matthias Scheutz, and Yu Zhang. Ai challenges in human-robot cognitive teaming. arXiv preprint arXiv:1707.04775, 2017.
- [34] 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.
- [35] Tathagata Chakraborti, Sarath Sreedharan, Sachin Grover, and Subbarao Kambhampati. Plan explanations as model reconciliation—an empirical study. arXiv preprint arXiv:1802.01013, 2018.
- [36] Tathagata Chakraborti, Sarath Sreedharan, and Subbarao Kambhampati. The emerging landscape of explainable ai planning and decision making. arXiv preprint arXiv:2002.11697, 2020.

- [37] 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.
- [38] 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.
- [39] Kishan Chandan, Vidisha Kudalkar, Xiang Li, and Shiqi Zhang. Negotiation-based human-robot collaboration via augmented reality. 2019 AAAI Fall Symposium: AI for HRI, 2019.
- [40] Christine T Chang, Matthew B Luebbbers, 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.
- [41] 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.
- [42] Sung Ho Choi, Kyeong-Beom Park, Dong Hyeon Roh, Jae Yeol Lee, Mustafa Mohammed, Yalda Ghasemi, and Heejin Jeong. An integrated mixed reality system for safety-aware human-robot collaboration using deep learning and digital twin generation. Robotics and Computer-Integrated Manufacturing, 73:102258, 2022.
- [43] 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.
- [44] 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.
- [45] 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.
- [46] 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.
- [47] Mark Colley, Svenja Krauss, Mirjam Lanzer, and Enrico Rukzio. How should automated vehicles communicate critical situations? a comparative analysis of visualization concepts. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 5(3):1–23, 2021.
- [48] 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.
- [49] 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.

- [50] Nancy J Cooke, Eduardo Salas, Janis A Cannon-Bowers, and Renee J Stout. Measuring team knowledge. Human factors, 42(1):151–173, 2000.
- [51] Michael D Coovert, Tiffany Lee, Ivan Shindeev, and Yu Sun. Spatial augmented reality as a method for a mobile robot to communicate intended movement. Computers in Human Behavior, 34:241–248, 2014.
- [52] 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.
- [53] David Roxbee Cox and Nancy Reid. The theory of the design of experiments. CRC Press, 2000.
- [54] Kenneth James Williams Craik. The nature of explanation, volume 445. CUP Archive, 1952.
- [55] Breanne Crockett, Kyler Ruvane, Matthew B Luebbers, and Bradley Hayes. Effective human-in-the-loop control handover via confidence-aware autonomy. 2023.
- [56] 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.
- [57] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. Nature, 521(7553):503–507, 2015.
- [58] 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.
- [59] 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.
- [60] Bruce H Deatherage. Auditory and other sensory forms of information presentation. Human engineering guide to equipment design, pages 123–160, 1972.
- [61] Alessandro Delfanti and Bronwyn Frey. Humanly extended automation or the future of work seen through amazon patents. Science, Technology, & Human Values, 46(3):655–682, 2021.
- [62] Daniel Clement Dennett. The intentional stance. MIT press, 1989.
- [63] 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.
- [64] Maximilian Diehl, Alexander Plopski, Hirokazu Kato, and Karinne Ramirez-Amaro. Augmented reality interface to verify robot learning. In 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), pages 378–383. IEEE.

- [65] 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.
- [66] 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.
- [67] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608, 2017.
- [68] 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.
- [69] Anca D Dragan. Robot planning with mathematical models of human state and action. arXiv preprint arXiv:1705.04226, 2017.
- [70] 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.
- [71] Anca D Dragan and Siddhartha S Srinivasa. Formalizing assistive teleoperation. MIT Press, July, 2012.
- [72] 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.
- [73] Mica R Endsley. Measurement of situation awareness in dynamic systems. Human factors, 37(1):65–84, 1995.
- [74] Joseph F Engelberger. Robotics in practice: management and applications of industrial robots. Springer Science & Business Media, 2012.
- [75] Muhammad Fahad, Zhuo Chen, and Yi Guo. Learning how pedestrians navigate: A deep inverse reinforcement learning approach. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 819–826, 2018.
- [76] Caroline Fairhurst, Catherine E Hewitt, and David J Torgerson. Using pairwise randomisation to reduce the risk of bias. Research Methods in Medicine & Health Sciences, 1(1):2–6, 2020.
- [77] Jaime F Fisac, Andrea Bajcsy, Sylvia L Herbert, David Fridovich-Keil, Steven Wang, Claire J Tomlin, and Anca D Dragan. Probabilistically safe robot planning with confidence-based human predictions. arXiv preprint arXiv:1806.00109, 2018.
- [78] Danyel Fisher. Hotmap: Looking at geographic attention. IEEE transactions on visualization and computer graphics, 13(6):1184–1191, 2007.
- [79] Paul M Fitts. Human engineering for an effective air-navigation and traffic-control system. 1951.

- [80] Matthew Fontaine and Stefanos Nikolaidis. Differentiable quality diversity. Advances in Neural Information Processing Systems, 34:10040–10052, 2021.
- [81] Matthew C Fontaine, Ya-Chuan Hsu, Yulun Zhang, Bryon Tjanaka, and Stefanos Nikolaidis. On the importance of environments in human-robot coordination. arXiv preprint arXiv:2106.10853, 2021.
- [82] Matthew C Fontaine and Stefanos Nikolaidis. Evaluating human–robot interaction algorithms in shared autonomy via quality diversity scenario generation. ACM Transactions on Human-Robot Interaction (THRI), 11(3):1–30, 2022.
- [83] 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.
- [84] David Fridovich-Keil, Andrea Bajcsy, Jaime F Fisac, Sylvia L Herbert, Steven Wang, Anca D Dragan, and Claire J Tomlin. Confidence-aware motion prediction for real-time collision avoidance. The International Journal of Robotics Research, 39(2-3):250–265, 2020.
- [85] John R Frost. The theory of search: a simplified explanation. Soza Limited, 1997.
- [86] John R Frost and Lawrence D Stone. Review of search theory: Advances and applications to search and rescue decision support. 2001.
- [87] Jack Gale, John Karasinski, and Steve Hillenius. Playbook for uas: Ux of goal-oriented planning and execution. In Engineering Psychology and Cognitive Ergonomics: 15th International Conference, EPCE 2018, Held as Part of HCI International 2018, Las Vegas, NV, USA, July 15-20, 2018, Proceedings 15, pages 545–557. Springer, 2018.
- [88] György Gergely, Zoltán Nádasdy, Gergely Csibra, and Szilvia Bíró. Taking the intentional stance at 12 months of age. Cognition, 56(2):165–193, 1995.
- [89] Piotr J Gmytrasiewicz and Prashant Doshi. A framework for sequential planning in multi-agent settings. Journal of Artificial Intelligence Research, 24:49–79, 2005.
- [90] Piotr J Gmytrasiewicz and Edmund H Durfee. Rational coordination in multi-agent environments. Autonomous Agents and Multi-Agent Systems, 3(4):319–350, 2000.
- [91] Matthew Gombolay, Ronald Wilcox, and Julie Shah. Fast scheduling of multi-robot teams with temporospatial constraints. 2013.
- [92] Matthew Gombolay, Xi Jessie Yang, Bradley Hayes, Nicole Seo, Zixi Liu, Samir Wadhwanja, 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.
- [93] 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.
- [94] 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.

- [95] Shashank Govindaraj, Keshav Chintamani, Jeremi Gancet, Pierre Letier, Boris van Lierde, Yashodhan Nevatia, Geert De Cubber, Daniel Serrano, Miguel Esbri Palomares, Janusz Bedkowski, et al. The icarus project-command, control and intelligence (c2i). In 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), pages 1–4. IEEE, 2013.
- [96] Yash Goyal, Ziyang Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In International Conference on Machine Learning, pages 2376–2384. PMLR, 2019.
- [97] Scott A Green, Mark Billingham, XiaoQi Chen, and J Geoffrey Chase. Human-robot collaboration: A literature review and augmented reality approach in design. International Journal of Advanced Robotic Systems, 5(1):1, 2008.
- [98] 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.
- [99] Herbert P Grice. Logic and conversation. In Speech acts, pages 41–58. Brill, 1975.
- [100] 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.
- [101] 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.
- [102] 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.
- [103] 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.
- [104] Brian C Gunia, Sharon H Kim, and Kathleen M Sutcliffe. Trust in safety-critical contexts. The Routledge Companion to Trust, pages 423–437, 2018.
- [105] David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang. Xai—explainable artificial intelligence. Science robotics, 4(37):eaay7120, 2019.
- [106] Himanshu Gupta, Bradley Hayes, and Zachary Sunberg. Intention-aware navigation in crowds with extended-space pomdp planning. In Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems, pages 562–570, 2022.
- [107] 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.
- [108] Martin Hägele, Klas Nilsson, J Norberto Pires, and Rainer Bischoff. Industrial robotics. In Springer handbook of robotics, pages 1385–1422. Springer, 2016.

- [109] 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.
- [110] 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.
- [111] Bradley Hayes and Michael Moniz. Trustworthy human-centered automation through explainable ai and high-fidelity simulation. In Advances in Simulation and Digital Human Modeling: Proceedings of the AHFE 2020 Virtual Conferences on Human Factors and Simulation, and Digital Human Modeling and Applied Optimization, July 16-20, 2020, USA, pages 3–9. Springer, 2021.
- [112] Bradley Hayes and Brian Scassellati. Challenges in shared-environment human-robot collaboration. learning, 8(9), 2013.
- [113] 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.
- [114] Bradley Hayes and Julie A Shah. Improving robot controller transparency through autonomous policy explanation. In 2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 303–312. IEEE, 2017.
- [115] Hooman Hedayati, Michael Walker, and Daniel Szafrir. Improving collocated robot teleoperation with augmented reality. In 2018 ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 78–86, 2018.
- [116] Guy Hoffman. Evaluating fluency in human–robot collaboration. IEEE Transactions on Human-Machine Systems, 49(3):209–218, 2019.
- [117] Guy Hoffman and Cynthia Breazeal. Cost-based anticipatory action selection for human–robot fluency. IEEE transactions on robotics, 23(5):952–961, 2007.
- [118] 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.
- [119] 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.
- [120] Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. Metrics for explainable ai: Challenges and prospects. arXiv preprint arXiv:1812.04608, 2018.
- [121] Ronald A Howard. Information value theory. IEEE Transactions on systems science and cybernetics, 2(1):22–26, 1966.
- [122] 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.

- [123] Sandy H Huang, David Held, Pieter Abbeel, and Anca D Dragan. Enabling robots to communicate their objectives. Autonomous Robots, 43(2):309–326, 2019.
- [124] 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.
- [125] Ashesh Jain, Brian Wojcik, Thorsten Joachims, and Ashutosh Saxena. Learning trajectory preferences for manipulators via iterative improvement. In Advances in Neural Information Processing Systems (NeurIPS), pages 575–583, 2013.
- [126] 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.
- [127] 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.
- [128] 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.
- [129] 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.
- [130] Daniel Kahneman. Thinking, fast and slow. macmillan, 2011.
- [131] George Karypis and Vipin Kumar. Metis: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. 1997.
- [132] 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.
- [133] Sarah Keren, Luis Pineda, Avigdor Gal, Erez Karpas, and Shlomo Zilberstein. Equi-reward utility maximizing design in stochastic environments. HSDIP, 2017:19, 2017.
- [134] Kazuhiko Kobayashi, Koichi Nishiwaki, Shinji Uchiyama, Hiroyuki Yamamoto, Satoshi Kagami, and Takeo Kanade. Overlay what humanoid robot perceives and thinks to the real-world by mixed reality system. In 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, pages 275–276. IEEE, 2007.
- [135] Bing Cai Kok and Harold Soh. Trust in robots: Challenges and opportunities. Current Robotics Reports, 1:297–309, 2020.
- [136] 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.

- [137] Tomáš Kot and Petr Novák. Utilization of the oculus rift hmd in mobile robot teleoperation. In Applied Mechanics and Materials, volume 555, pages 199–208. Trans Tech Publ, 2014.
- [138] Dennis Krupke, Frank Steinicke, Paul Lubos, Yannick Jonetzko, Michael Görner, and Jianwei Zhang. Comparison of multimodal heading and pointing gestures for co-located mixed reality human-robot interaction. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1–9. IEEE, 2018.
- [139] Anagha Kulkarni, Sarath Sreedharan, Sarah Keren, Tathagata Chakraborti, David E Smith, and Subbarao Kambhampati. Designing environments conducive to interpretable robot behavior. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 10982–10989. IEEE, 2020.
- [140] 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.
- [141] 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.
- [142] 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.
- [143] 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. arXiv preprint arXiv:2001.04377, 2020.
- [144] 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.
- [145] 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.
- [146] Przemyslaw A. Lasota and Julie A. Shah. Analyzing the effects of human-aware motion planning on close-proximity human–robot collaboration. Human Factors, 57(1):21–33, 2015. PMID: 25790568.
- [147] Przemyslaw A. Lasota and Julie A. Shah. A multiple-predictor approach to human motion prediction. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 2300–2307, 2017.
- [148] 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.

- [149] John D Lee and Katrina A See. Trust in automation: Designing for appropriate reliance. Human factors, 46(1):50–80, 2004.
- [150] Michael Lewis, Katia Sycara, and Phillip Walker. The Role of Trust in Human-Robot Interaction, pages 135–159. Springer International Publishing, Cham, 2018.
- [151] Daniel Leyzberg, Aditi Ramachandran, and Brian Scassellati. The effect of personalization in longer-term robot tutoring. ACM Transactions on Human-Robot Interaction (THRI), 7(3):19, 2018.
- [152] 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.
- [153] Qinghua Li, Zhao Zhang, Yue You, Yaqi Mu, and Chao Feng. Data driven models for human motion prediction in human-robot collaboration. IEEE Access, 8:227690–227702, 2020.
- [154] Xiao Li, Wen Yi, Hung-Lin Chi, Xiangyu Wang, and Albert PC Chan. A critical review of virtual and augmented reality (vr/ar) applications in construction safety. Automation in Construction, 86:150–162, 2018.
- [155] Peter Lipton. Contrastive explanation. Royal Institute of Philosophy Supplements, 27:247–266, 1990.
- [156] 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.
- [157] 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.
- [158] Matthew B Luebbers, Connor Brooks, Carl L Mueller, Daniel Szafir, and Bradley Hayes. Arc-ld: Using augmented reality for interactive long-term robot skill maintenance via constrained learning from demonstration. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pages 3794–3800. IEEE, 2021.
- [159] 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, 2022.
- [160] 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.
- [161] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. Advances in neural information processing systems, 30, 2017.
- [162] Ren.C Luo and Licong Mai. Human intention inference and on-line human hand motion prediction for human-robot collaboration. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5958–5964, 2019.

- [163] 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.
- [164] Michal Luria, Jodi Forlizzi, and Jessica Hodgins. The effects of eye design on the perception of social robots. In 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pages 1032–1037. IEEE, 2018.
- [165] Udit Madan, Michael Ellsworth Bundy, David Daniel Glick, and John Elias Darrow. Augmented reality user interface facilitating fulfillment, April 25 2017. US Patent 9,632,313.
- [166] Jim Mainprice, Rafi Hayne, and Dmitry Berenson. Goal set inverse optimal control and iterative replanning for predicting human reaching motions in shared workspaces. IEEE Transactions on Robotics, 32(4):897–908, 2016.
- [167] Sotiris Makris, Panagiotis Karagiannis, Spyridon Koukas, and Aleksandros-Stereos Matthaiakis. Augmented reality system for operator support in human-robot collaborative assembly. CIRP Annals, 65(1):61–64, 2016.
- [168] Gustav Markkula and Mehmet Dogar. How accurate models of human behavior are needed for human-robot interaction? for automated driving? arXiv preprint arXiv:2202.06123, 2022.
- [169] 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.
- [170] Julieta Martinez, Michael J. Black, and Javier Romero. On human motion prediction using recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.
- [171] 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.
- [172] John P McIntire, Paul R Havig, and Eric E Geiselman. Stereoscopic 3d displays and human performance: A comprehensive review. Displays, 35(1):18–26, 2014.
- [173] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence, 2018.
- [174] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence, 267:1–38, 2019.
- [175] Tim Miller. Contrastive explanation: A structural-model approach. The Knowledge Engineering Review, 36:e14, 2021.
- [176] Marvin Minsky. A framework for representing knowledge. 1974.
- [177] Walter Mischel. The marshmallow test: Why self-control is the engine of success. Little, Brown New York, 2015.

- [178] 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.
- [179] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. arXiv preprint arXiv:1504.04909, 2015.
- [180] Carl Mueller, Jeff Venix, and Bradley Hayes. Robust robot learning from demonstration and skill repair using conceptual constraints. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 6029–6036. IEEE, 2018.
- [181] Meinard Müller. Dynamic time warping. Information retrieval for music and motion, pages 69–84, 2007.
- [182] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In Icml, volume 1, page 2, 2000.
- [183] Stefanos Nikolaidis, Przemyslaw Lasota, Gregory Rossano, Carlos Martinez, Thomas Fuhlbrügge, and Julie Shah. Human-robot collaboration in manufacturing: Quantitative evaluation of predictable, convergent joint action. In IEEE ISR 2013, pages 1–6. IEEE, 2013.
- [184] 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.
- [185] Stefanos Nikolaidis and Julie Shah. Human-robot cross-training: computational formulation, modeling and evaluation of a human team training strategy. In 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 33–40. IEEE, 2013.
- [186] 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.
- [187] Olle Nilsson and Antoine Cully. Policy gradient assisted map-elites. In Proceedings of the Genetic and Evolutionary Computation Conference, pages 866–875, 2021.
- [188] Albert A Nofi. Defining and measuring shared situational awareness. 2000.
- [189] 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.
- [190] Özgür Örün and Yavuz Akbulut. Effect of multitasking, physical environment and electroencephalography use on cognitive load and retention. Computers in Human Behavior, 92:216–229, 2019.
- [191] Takayuki Osogami and Makoto Otsuka. Restricted boltzmann machines modeling human choice. In Advances in Neural Information Processing Systems, pages 73–81, 2014.
- [192] Makoto Otsuka and Takayuki Osogami. A deep choice model. In Thirtieth AAAI Conference on Artificial Intelligence, 2016.

- [193] 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.
- [194] Joon Sung Park, Rick Barber, Alex Kirlik, and Karrie Karahalios. A slow algorithm improves users’ assessments of the algorithm’s accuracy. Proceedings of the ACM on Human-Computer Interaction, 3(CSCW):1–15, 2019.
- [195] 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, pages 15–23. SPIE, 2020.
- [196] 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.
- [197] Mark Pfeiffer, Ulrich Schwesinger, Hannes Sommer, Enric Galceran, and Roland Siegwart. Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2096–2101, 2016.
- [198] Nick Pidgeon, John Walls, Andrew Weyman, and Tom Horlick-Jones. Perceptions of and trust in the health and safety executive as a risk regulator. 2003.
- [199] Scott Plous. The psychology of judgment and decision making. Mcgraw-Hill Book Company, 1993.
- [200] David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? Behavioral and brain sciences, 1(4):515–526, 1978.
- [201] Claudia Pérez-D’Arpino and Julie A. Shah. Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 6175–6182, 2015.
- [202] C Perez Quintero, Sarah Li, Cole Shing, Wesley Chan, Sara Sheikholeslami, HF Machiel Van der Loos, and Elizabeth Croft. Robot programming through augmented trajectories. In VAM-HRI Workshop at The International Conference on Human Robot Interaction, 2018.
- [203] Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. Annual Review of Control, Robotics, and Autonomous Systems, 3, 2020.
- [204] 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.
- [205] 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.

- [206] 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.
- [207] Paul Robinette, Wenchen Li, Robert Allen, Ayanna M Howard, and Alan R Wagner. Overtrust of robots in emergency evacuation scenarios. In 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 101–108. IEEE, 2016.
- [208] Eric Rosen, David Whitney, Michael Fishman, Daniel Ullman, and Stefanie Tellex. Mixed reality as a bidirectional communication interface for human-robot interaction. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020.
- [209] 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.
- [210] Avi Rosenfeld and Ariella Richardson. Explainability in human-agent systems. Autonomous Agents and Multi-Agent Systems, 33:673–705, 2019.
- [211] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Darius M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. The International Journal of Robotics Research, 39(8):895–935, 2020.
- [212] Dorsa Sadigh, Anca D Dragan, Shankar Sastry, and Sanjit A Seshia. Active preference-based learning of reward functions. In Robotics: Science and Systems, 2017.
- [213] 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.
- [214] 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.
- [215] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, 26(1):43–49, 1978.
- [216] 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.
- [217] 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.
- [218] 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.
- [219] Brian Scassellati. Theory of mind for a humanoid robot. Autonomous Robots, 12(1):13–24, 2002.

- [220] Noam Scheiber. Inside an amazon warehouse, robots’ ways rub off on humans. The New York Times, 3, 2019.
- [221] 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.
- [222] Nestor Schmajuk and Horatiu Voicu. Exploration and navigation using hierarchical cognitive maps. Animal Spatial Cognition: Comparative, Neural, and Computational Approaches, MF Brown and RG Cook. p. Available: <http://www.pigeon.psy.tufts.edu/asc/Schmajuk/Default.htm>, 2006.
- [223] Mariah L Schrum, Erin Hedlund, and Matthew C Gombolay. Improving robot-centric learning from demonstration via personalized embeddings. arXiv preprint arXiv:2110.03134, 2021.
- [224] Mariah L Schrum, Emily Sumner, Matthew C Gombolay, and Andrew Best. Maveric: A data-driven approach to personalized autonomous driving. IEEE Transactions on Robotics, 2024.
- [225] 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.
- [226] 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.
- [227] 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.
- [228] Yi Shen, Gunther Reinhart, and Mitchell Miendger Tseng. A design approach for incorporating task coordination for human-robot-coexistence within assembly systems. In 2015 Annual IEEE Systems Conference (SysCon), pages 426–431. IEEE, 2015.
- [229] Thomas B Sheridan. Human–robot interaction: status and challenges. Human factors, 58(4):525–532, 2016.
- [230] 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.
- [231] 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, 39(7):1390–1404, 2023.
- [232] David Silvera-Tawil, DanaKai Bradford, and Christine Roberts-Yates. Talk to me: The role of human-robot interaction in improving verbal communication skills in students with autism or intellectual disability. In 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pages 1–6. IEEE, 2018.

- [233] Herbert A Simon. Rational choice and the structure of the environment. Psychological review, 63(2):129, 1956.
- [234] Herbert A Simon. Rational decision making in business organizations. The American economic review, 69(4):493–513, 1979.
- [235] Herbert A Simon. Bounded rationality. Utility and probability, pages 15–18, 1990.
- [236] Taufik Akbar Sitompul and Markus Wallmyr. Using augmented reality to improve productivity and safety for heavy machinery operators: State of the art. In Proceedings of the 17th International Conference on Virtual-Reality Continuum and Its Applications in Industry, pages 1–9, 2019.
- [237] Leopold Slotta-Bachmayr. How burial time of avalanche victims is influenced by rescue method: An analysis of search reports from the alps. Natural Hazards, 34:341–352, 2005.
- [238] Dennis Sprute, Klaus Tönnies, and Matthias König. Virtual borders: Accurate definition of a mobile robot’s workspace using augmented reality. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 8574–8581. IEEE, 2018.
- [239] Dennis Sprute, Klaus Tönnies, and Matthias König. A study on different user interfaces for teaching virtual borders to mobile robots. International Journal of Social Robotics, 11(3):373–388, 2019.
- [240] Dennis Sprute, Philipp Viertel, Klaus Tönnies, and Matthias König. Learning virtual borders through semantic scene understanding and augmented reality. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4607–4614. IEEE, 2019.
- [241] Sarath Sreedharan, Subbarao Kambhampati, et al. Balancing explicability and explanation in human-aware planning. In 2017 AAAI Fall Symposium Series, 2017.
- [242] Sarath Sreedharan, Anagha Kulkarni, and Subbarao Kambhampati. Balancing communication and behavior. In Explainable Human-AI Interaction: A Planning Perspective, pages 95–105. Springer, 2022.
- [243] Daniel Szafrir. Mediating human-robot interactions with virtual, augmented, and mixed reality. In 2019 International Conference on Human-Computer Interaction (HCI), pages 124–149. Springer, 2019.
- [244] Daniel Szafrir, Bilge Mutlu, and Terry Fong. Communicating directionality in flying robots. In Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, pages 19–26, 2015.
- [245] Daniel Szafrir and Danielle Albers Szafrir. Connecting human-robot interaction and data visualization. In Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction, pages 281–292, 2021.
- [246] 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.

- [247] 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.
- [248] 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.
- [249] Aaquib Tabrez, Matthew B Luebbers, and Bradley Hayes. A survey of mental modeling techniques in human-robot teaming. Current Robotics Reports, 1:259–267, 2020.
- [250] 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.
- [251] 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.
- [252] 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.
- [253] Dušan Tatić and Bojan Tešić. The application of augmented reality technologies for the improvement of occupational safety in an industrial environment. Computers in Industry, 85:1–10, 2017.
- [254] Stefanie Tellex, Ross Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. Asking for help using inverse semantics. 2014.
- [255] Richard H Thaler and Cass R Sunstein. Nudge: Improving decisions about health, wealth, and happiness. Penguin, 2009.
- [256] Andrea Thomaz, Guy Hoffman, and Maya Cakmak. Computational human-robot interaction. Foundations and Trends in Robotics, 4(2-3):105–223, 2016.
- [257] 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.
- [258] Yi-Shiuan Tung, Matthew B Luebbers, Alessandro Roncone, and Bradley Hayes. Workspace optimization techniques to improve prediction of human motion during human-robot collaboration. In Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, pages 743–751, 2024.
- [259] Amos Tversky and Daniel Kahneman. Judgment under uncertainty: Heuristics and biases. science, 185(4157):1124–1131, 1974.
- [260] 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.

- [261] Milos Vasic and Aude Billard. Safety issues in human-robot interactions. In 2013 IEEE International Conference on Robotics and Automation, pages 197–204, 2013.
- [262] Louise Veling and Conor McGinn. Qualitative research in hri: A review and taxonomy. International Journal of Social Robotics, 13:1689–1709, 2021.
- [263] Luca Viganò and Daniele Magazzeni. Explainable security. arXiv preprint arXiv:1807.04178, 2018.
- [264] Valeria Villani, Fabio Pini, Francesco Leali, and Cristian Secchi. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. Mechatronics, 55:248–266, 2018.
- [265] 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.
- [266] Alan R Wagner, Jason Borenstein, and Ayanna Howard. Overtrust in the robotic age. Communications of the ACM, 61(9):22–24, 2018.
- [267] Michael Walker, Hooman Hedayati, Jennifer Lee, and Daniel Szafr. Communicating robot motion intent with augmented reality. In Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, pages 316–324, 2018.
- [268] Michael E Walker, Hooman Hedayati, and Daniel Szafr. Robot teleoperation with augmented reality virtual surrogates. In 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 202–210. IEEE, 2019.
- [269] Sebastian Wallkötter, Silvia Tulli, Ginevra Castellano, Ana Paiva, and Mohamed Chetouani. Explainable agents through social cues: A review. arXiv preprint arXiv:2003.05251, 2020.
- [270] 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.
- [271] 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.
- [272] Jonathan Weisz, Peter K Allen, Alexander G Barszap, and Sanjay S Joshi. Assistive grasping with an augmented reality user interface. The International Journal of Robotics Research, 36(5-7):543–562, 2017.
- [273] Katherine S Welfare, Matthew R Hallowell, Julie A Shah, and Laurel D Riek. Consider the human work experience when integrating robotics in the workplace. In 2019 14th ACM/IEEE international conference on human-robot interaction (HRI), pages 75–84. IEEE, 2019.
- [274] Dominik Widmann and Yiannis Karayiannidis. Human motion prediction in human-robot handovers based on dynamic movement primitives. In 2018 European Control Conference (ECC), pages 2781–2787, 2018.
- [275] Yorick Wilks and Afzal Ballim. Multiple agents and the heuristic ascription of belief. Computing Research Laboratory, New Mexico State University, 1986.

- [276] Tom Williams, Matthew Bussing, Sebastian Cabrol, Elizabeth Boyle, and Nhan Tran. Mixed reality deictic gesture for multi-modal robot communication. In 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 191–201. IEEE, 2019.
- [277] 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.
- [278] John R Wilson and Andrew Rutherford. Mental models: Theory and application in human factors. Human Factors, 31(6):617–634, 1989.
- [279] 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, 2(1):56–81, 2013.
- [280] Michał Wysokiński, Robert Marcjan, and Jacek Dajda. Decision support software for search & rescue operations. Procedia Computer Science, 35:776–785, 2014.
- [281] 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.
- [282] Tomonori Yamamoto, Niki Abolhassani, Sung Jung, Allison M Okamura, and Timothy N Judkins. Augmented reality and haptic interfaces for robot-assisted surgery. The International Journal of Medical Robotics and Computer Assisted Surgery, 8(1):45–56, 2012.
- [283] 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.
- [284] 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.
- [285] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. arXiv preprint arXiv:2006.10214, 2020.
- [286] Haoqi Zhang, Yiling Chen, and David C Parkes. A general approach to environment design with one agent. In Twenty-First International Joint Conference on Artificial Intelligence. Citeseer, 2009.
- [287] Yulun Zhang, Matthew C. Fontaine, Varun Bhatt, Stefanos Nikolaidis, and Jiaoyang Li. Multi-robot coordination and layout design for automated warehousing. In Edith Elkind, editor, Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23, pages 5503–5511. International Joint Conferences on Artificial Intelligence Organization, 8 2023. Main Track.
- [288] 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.

- [289] 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.