UNIVERSITY OF CALIFORNIA

Los Angeles

Bidirectional Mental State Alignment for Human-Machine Collaboration

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Statistics

by

Xiaofeng Gao

2022

ABSTRACT OF THE DISSERTATION

Bidirectional Mental State Alignment for Human-Machine Collaboration

by

Xiaofeng Gao
Doctor of Philosophy in Statistics
University of California, Los Angeles, 2022
Professor Song-Chun Zhu, Chair

For machines working alongside with humans, it is necessary to understand humans' mental states, including desires, beliefs and intentions for better interactions. In addition, humans also need to understand machines' capabilities and limitations to trust and rely on machines approriately. Achieving such bidirectional mental state alignment is crucial to the success of human-machine collaboration. This dissertation addresses this core challenge from both directions, spanning the domains of embodied artificial intelligence, autonomous driving and human-robot interaction. In the first direction of machines understanding humans, I propose a virtual environment for embodied agents and human users to work on daily activities via simulation. To enable embodied agents to better understand and execute human commands, I propose a benchmark allowing them to actively ask questions to resolve language ambiguities. For autonomous driving systems to have an accurate mental model of drivers, I propose a novel protocol to evaluate the effects of human-machine interfaces on drivers' situational awareness in different traffic conditions. In the second direction, I study how robots can generate communicative actions to be better understood by humans. I propose i) an action parsing algorithm based on an And-Or graph representation to generate explanations of task plans, ii) a task and motion planning framework to calibrate robot reachable

workspace by expressive motions. My works culminate in building a computational framework for bidirectional value alignment, which is evaluated in the human-machine collaborative scout exploration game.

The dissertation of Xiaofeng Gao is approved.

Demetri Terzopoulos

Tao Gao

Ying Nian Wu

Song-Chun Zhu, Committee Chair

University of California, Los Angeles

2022

To my family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First, I'd like to thank my advisor Prof. Song-Chun Zhu for encouraging me to work on human-machine collaboration and explainable AI, and for supporting me financially on relevant projects. These are interdisciplinary fields, and Prof. Zhu's supervision broadened my horizon, allowing me to make connections between different domains to tackle challenging problems.

I'd also like to express my sincere gratitude to my great mentors at UCLA: Prof. Hongjing Lu, Prof. Ying Nian Wu, Prof. Tao Gao, Prof. Demetri Terzopoulos. Their discussions and classes inspired a lot of works presented in this dissertation.

It has been my honor to be in the VCLA family during the last five years. I'd like to express thanks to my collaborators and labmates in VCLA:

Dr. Tianmin Shu, for intruducing me to VCLA and for all the mentoring and collaborations.

Dr. Luyao Yuan, Dr. Mark Edmonds, Dr. Zilong Zheng, and Prof. Yixin Zhu, for the time we spent together on the scout exploration project.

Ran Gong, Yizhou Zhao, Shu Wang, Ziheng Zhou, Prof. Zhixiong Nan, Dr. Xu Xie, Dr. Lifeng Fan, Dr. Siyuan Huang and many others, for collaborations and discussions during various projects.

I'm fortunate enough to have two wonderful internships, which give me valuable research experience in industry and offer me a different perspective from academia. Thus I'd like to thank my mentors and collaborators for their support during my internships: Dr. Xingwei Wu, Dr. Teruhisa Misu, Dr. Kumar Akash, Samson Ho, Dr. Qiaozi Gao, Dr. Kaixiang Lin, Govind Thattai and Prof. Gaurav Sukhatme.

Finally, I'd like to dedicate this dissertation to my family for their support throughout my doctoral studies.

VITA

2021	Applied Scientist Intern, Amazon
2021	Research Intern, Honda Research Institute USA
2019	Ph.D. Candidate in Statistics, UCLA
2018	Graduate Teaching Assistant, Department of Statistics, UCLA
2017-2022	Graduate Research Assistant, Department of Statistics, UCLA
2017	B.Eng. in Electronic Engineering, Fudan University

PUBLICATIONS

(* indicates equal contribution)

L. Yuan*, **X. Gao***, Z. Zheng*, M. Edmonds, Y. Wu, F. Rossano, H. Lu, Y. Zhu, S.-C. Zhu. Two-way street: In-situ Bidirectional Human-Robot Value Alignment with Communicative Learning. Submitted to Science Robotics, 2022.

X. Gao, Q. Gao, R. Gong, K. Lin, G. Thattai, G. Sukhatme. DialFRED: Dialogue-Enabled Agents for Embodied Instruction Following. Submitted to IEEE Robotics and Automation Letters (RA-L), 2022.

X. Gao, X. Wu, S. Ho, T. Misu, K. Akash. Effects of Augmented-Reality-Based Assisting Interfaces on Drivers' Object-Wise Situational Awareness in Highly Autonomous Vehicles. IEEE Intelligent Vehicles Symposium (IV), 2022.

X. Gao, L. Yuan, T. Shu, H. Lu and S.-C. Zhu. Show Me What You Can Do: Capability Calibration on Reachable Workspace for Human-Robot Collaboration. IEEE Robotics and Automation Letters (RA-L), 2022.

Z. Nan, J. Jiang, **X. Gao**, S. Zhou, W. Zuo, W. Ping, N. Zheng. Predicting Task-driven Attention via Integrating Bottom-up Stimulus and Top-down Guidance. IEEE Transactions on Image Processing (T-IP), 2021.

X. Gao*, R. Gong*, Y. Zhao, S. Wang, T. Shu, and S.-C. Zhu. Joint Mind Modeling for Explanation Generation in Complex Human-Robot Collaborative Tasks. IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), 2020.

X. Gao, R. Gong, T. Shu, X. Xie, S. Wang, and S.-C. Zhu. VRKitchen: an Interactive 3D Environment for Learning Real Life Cooking Tasks. International Conference on Machine Learning (ICML) Workshop on Reinforcement Learning for Real Life, 2019.

T. Shu, **X. Gao**, M. S. Ryoo, and S.-C. Zhu. Learning Social Affordance Grammar from Videos: Transferring Human Interactions to Human-Robot Interactions. IEEE International Conference on Robotics and Automation (ICRA), 2017.

CHAPTER 1

Introduction

With recent advances in aritificial intelligence (AI), intelligent machines have been developed to help us extensively in our daily lives. Advanced Driver Assistance Systems (ADAS) can maintain the car's speed and distance relative to other vehicles, and help drivers with parking. Intelligent robot vacuums can clean our houses without colliding with obstacles. Facial recognition systems allow us to make quick and secured payments without cash, credit card or mobile phones. Indeed, these applications demonstrate the acheivements of making machines perceive and change the physical environment. Nonetheless, for building machines that can better interact with humans, one important missing piece of research is related to mutual understanding between humans and machines.

Understanding human minds, in addition to the physical world, enables machines to have human-like interactions with people. It is well known that humans understand that others may have mental states different from oneself. This capability of attributing unique mental state to each individual is denoted as theory of mind (ToM) [PW78]. Such mental states, including beliefs, desires and intentions, are crucial for humans to comprehend, predict and influence the behaviors of other people. If we ever want to make intelligent machines that can infer what their human partners may know, predict their future behaviors, and establish efficient communications, understanding human mental states should be one crucial step.

One of the largest limitations in this direction is the lack of human data and standardized benchmarks. My works contributes to this area by creating virtual environments in simulation for humans and robots to work on daily activities. I propose a benchmark allowing embodied agents

to actively ask questions to humans to resolve language ambiguities for instruction following. In addition, I propose a novel protocol to study the effects of highlighting on Augmented-Reality-based user interfaces on drivers' situational awareness.

An equally important topic is for humans to understand machines. As machines are becoming increasingly complicated and autonomous, understanding how they make decisions as well as their capabilities and limitations enable humans to make better use of them. Failures to do so inevitably lead to misuse or disuse of the machine, inappropriate trust, and sometimes even catastrophic results. The situations have been exacerbated by recent popularity of deep learning models, which seem to be black boxes to most non-expert users.

Previous works in this area, categorized as explainable AI (XAI), focus on producing interpretable visualizations for data-driven machine learning models mainly used in perception [KWG18, AWZ20]. We believe that a goal-directed AI agent working with humans should be viewed as a system with multiple functionalities, including perception, planning, cognition and control. We argue that i) the target of human understanding should be more than the perception module, ii) the communication between the agent and human users should more diverse, and iii) efficient communication also requires understanding of the human mental states. Thus our works contribute to this area by generating communicative actions via cognitive modeling through a wide range of modalities suitable for the embodiment of the machine, ranging from assisting user interface for autonomous vehicles, expressive motion demonstrations for robots, and questions and task plan visualizations for embodied virtual agents.

In summary, this dissertation proposes simulation environment, benchmark, user study protocol and computational framework to address the challenge of human-machine mental state alignment, spanning the domains of embodied AI, autonomous driving and human-robot interaction. The detailed contributions of each chapter are outlined below.

In Chapter 2, we propose VRKitchen, a virtual kitchen environment for simulating cooking activities with rich object state changes and compositional goals. We build interfaces for both AI agents and human users to control the embodied agent. We also propose a new challenge including

two sets of task for evaluating the agent's ability for long-term task planning and fine-grained object manipulation respectively.

In Chapter 3, we propose DialFRED, an embodied instruction following benchmark allowing an agent to actively ask questions to the human user and use the information to better complete the task. The benchmark consists of 25 types of tasks and 53K human-annotated task-relevant questions and answers. We also propose a model based on a questioner-performer framework showing the effectiveness of adding dialogue for improving the performance of embodied instruction following.

In Chapter 4, we propose a novel protocol for evaluating the effects of a Augmented-Reality (AR) based user interface that highlights potential hazardous objects on drivers' situational awareness for objects with different locations, types and traffic densities. We further conduct a simulator experiment using the protocol and carefully analyzed the effect of highlighting in different conditions. The results show that highlighting has mixed effects on drivers' situational awareness, depending on object characteristics and traffic density.

In Chapter 5, we design a real-time collaborative cooking game to study human-robot collaborations. We propose an action-parsing algorithm based on an And-Or graph representation to infer human mental states from their actions. We further propose an explanation generation framework based on the inferred mental states for giving humans hints and explaining what the robot is doing. We demonstrate that our approach improves user perception of the robot and leads to more effective collaboration.

In Chapter 6, we propose REMP, a novel motion planning algorithm enabling the human to understand the robot's reaching capabilities. The algorithm is based on modeling perceived robot capability as a human's belief over robot's reachable workspace, and integrates the belief update process into motion planning via cost functions in trajectory optimiztion. We demonstrate that REMP can significantly increase human's reachability estimation accuracy, as well as the performance of the subsequent collaboration task.

In Chapter 7, we propose a human-machine teaming system instantiated as a collaborative scout exploration game, requiring the machine to both extract useful information from human's feedback to infer human's values and explain what they plan to do based on its current value estimation. We demonstrate that our iterative teacher-aware learning and explanation generation framework is able to achieve bidirectional value alignment in an in-situ manner.

CHAPTER 2

Building Interactive Environments for Embodied Agent Learning

2.1 Introduction

Fortunately, humans have built AI systems that can accurately detect and recognize objects [KH12, HGD17], generate vivid natural images [BDS18], and beat human Go champions [SSS17]. However, a truly intelligent machine agent should be able to solve a large set of complex tasks in the physical world by adapting itself to unseen surroundings and planning a long sequence of actions to reach the desired goals, which is still beyond the the capacity of current machine models. This gives rise to the need of advancing research on task-oriented learning. In particular, we are interested in the following three task-oriented learning problems for the present work.

Learning visual representation of a dynamic environment. In the process of solving a task in a dynamic environment, the appearance of the same object may change dramatically as a result of actions [ILA15, FR13, LWZ17]. To capture such variation in object appearance, the agent is required to have a better visual representation of the environment dynamics. For example, the agent should recognize the tomato even if it is cut into pieces and put into container. To acquire such visual knowledge, it is important for an agent to learn from physical interactions and reason over the underlying causality of object state changes. There have been work on implementing interaction-based learning in lab environments [LGF16,ACM15,HKB15], but the limited scenarios greatly restrict scalability and reproducitibility of prior work. Instead, we believe that building a simulation platform is a good alternative since i) performance of different algorithms can be easily

evaluated and benchmarked, ii) a large set of diverse and realistic environments and tasks can be designed and customized.

Learning to generate long-term plans for complex tasks. A complex task is often composed of various sub-tasks, each of which has its own sub-goal [BAR44]. Thus the agent needs to take a long sequence of actions to finish the task. The large number of possible actions in the sample space and the extremely sparse rewards make it difficult to steer the policy to the right direction. Recently, many researchers have focused on learning hierarchical policies [SP02, AKL16, SXS18] in simple domains. In this work, we provide a realistic environment where the agent can learn to compose long-term plans for daily life tasks that humans encounter in the real world.

Learning from human demonstrations to bootstrap agents' models. Training an agent from scratch is extremely difficult in complex environments. To bootstrap the training, it is common to let an agent to imitate human experts by watching human demonstrations [NR00,ZMB08,GGC16]. Previous work has shown that learning from demonstrations (or imitation learning) significantly improves the learning efficiency and achieves a higher performance than reinforcement learning does [ZGK17, HVP17]. However, it is expensive and time consuming to collect diverse human demonstrations with high qualities. We believe that virtual reality games can provide us with an ideal medium to crowd source demonstrations from a broad range of users [AD08].

In this work, we focus on simulating cooking activities and two sets of cooking tasks (using common tools and preparing dishes) in a virtual kitchen environment, VRKitchen. We illustrate how this system can address the emerged needs for the three task-oriented learning problems in an example shown in Figure 2.1, where an agent makes a sandwich in one of the kitchens created in our system.

- The environment allows the agent to interact with different *tools* and *ingredients* and simulates a variety of object changes. E.g., the bread changes its color when it is being heated in the oven, and the tomato turns into slices after it is cut. The agent's interactions with the physical world when performing cooking tasks will result in large variations and temporal

changes in objects' appearance and physical properties, which calls for a task-oriented visual representation.

- To make a sandwich, the agent needs to perform a long sequence of actions, including taking ingredients from a fridge, putting cheese and ham on the bread, toasting the bread, adding some sliced tomato and putting some sauce on the bread. To quickly and successfully reach the final goal, it is necessary to equip the agent with the ability to conduct long-term planning.
- We build two interfaces to allow an AI algorithm as well as a human user to control the embodied agent respectively, thus humans can give demonstrations using VR devices at any places in the world, and the AI algorithms can learn from these demonstrations and perform the same tasks in the same virtual environments.

In summary, our main contributions are:

- A configurable virtual kitchen environment in a photo-realistic 3D physical simulation which enables a wide range of cooking tasks with rich object state changes and compositional goals;
- A toolkit including a VR-based user interface for collecting human demonstrations, and a Python API for training and testing different AI algorithms in the virtual environments.
- Proposing a new challenge – VR chef challenge, to provide standardized evaluation for benchmarking different approaches in terms of their learning efficiency in complex 3D environments.
- A new human demonstration dataset of various cooking tasks – UCLA VR chef dataset.

2.2 Related Work

Simulation platforms. Traditionally, visual representations are learned from static datasets. Either containing prerecorded videos [RAA12] or images [JWS09], most of them fail to capture the

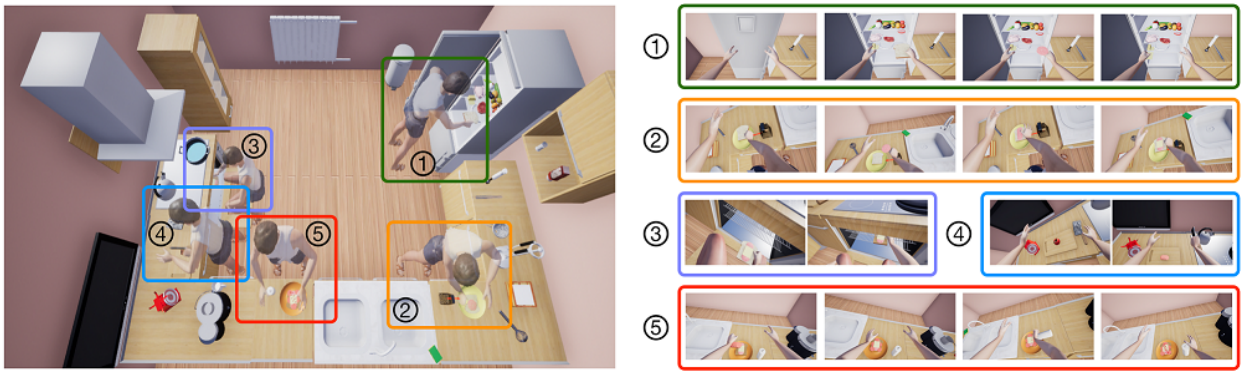


Figure 2.1: A sample sequence of an agent making a *sandwich*. Rectangles on the left graph represents five necessary sub-tasks, including (1) taking ingredients from fridge, (2) putting ham and cheese on the bread, (3) use the oven, (4) cut tomato and (5) add some sauce. Each rectangle on the right graph indicates atomic actions required to finish a sub-task.

Env.	Large-scale	Physics	Realistic	State	Manip	Avatar	Demo
Malmo [JHH16]	√			√			
DeepMind Lab [BLT16]							
VizDoom [KWR17]							
MINOS [SCD17]	√		√				
HoME [BPA17]	√	√	√				
Gibson [XZH18]	√	√	√			√	
House3D [WWG18]	√	√	√				
AI2-THOR [KMH17a]		√	√	√	√		
VirtualHome [PRB18]			√	√	√	√	
SURREAL [FZZ18]		√			√		√
VRKitchen (ours)		√	√	√	√	√	√

Table 2.1: Comparison with other 3D virtual environments. Large-scale: a large number of scenes. Physics: physics simulation. Realistic: photo-realistic rendering. State: changeable object states. Manip: enabling object interactions and manipulations. Avatar: humanoid virtual agents. Demo: user interface to collect human demonstrations.

dynamics in viewpoint and object state during human activities, in spite of their large scale.

To address this issue, there has been a growing trend to develop 3D virtual platforms for training embodied agents in dynamic environments. Typical systems include 3D game environments [KWR17, BLT16, JHH16], and robot control platforms [TET12, CB16, FZZ18, PAR18]. While these systems offer physics simulation and 3D rendering, they fail to provide realistic envi-

ronments and daily tasks humans face in the real world.

More recently, based on 3D scene datasets such as Matterport3D [CDF18] and SUNCG [SYZ17], there are have been several systems simulating more realistic indoor environments [BPA17,WWG18, SCD17, MHL17, XZH18] for visual navigation tasks and basic object interactions such as pushing and moving funitures [KMH17a]. While the environments in these systems are indeed more realistic and scalable compared to previous systems, they still can not simulate complex object manipulation that are common in our daily life. [PRB18] took a step forward and has created a dataset of common household activities with a larger set of agent actions including pick-up, switch on/off, sit and stand-up. However, this system was only designed for generating data for video understanding. In contrast, our system emphasizes training and evaluating agents on virtual cooking tasks, which involves fine-grained object manipulation on the level of object parts (e.g., grasping the handle of a knife), and flexible interfaces for allowing both human users and AI algorithms to perform tasks. Our system also simulates the animation of object state changes (such as the process of cutting a fruit) and the gestures of humanoid avatars (such as reaching for an object) instead of only showing pre-conditions and post-effects as in [KMH17a]. A detailed comparison between our system and other virtual environments is summarized in Table 2.1.

Imitation learning. Learning from demonstration or imitation learning is proven to be an effective approach to train machine agents efficiently [AN04a, SS08, RGB10]. Collecting diverse expert demonstrations with 3D ground-truth information in real world is extremely difficult. We believe the VR interface in our system can greatly simplify and scale up the demonstration collection.

VR for AI. VR provides a convenient way to evaluate AI algorithms in tasks where interaction or human involvement is necessary. Researches have been conducted on many relevant domains, including physical intuition learning [LGF16], human-robot interaction [LRM17, GRO17], learning motor control from human demonstrations [HKB15, KNM01, BCC01]. Researchers have also used VR to collect data and train computer vision models. To this end, several plugins for game engines have been released, such as UETorch [LGF16] and UnrealCV [QY16]. To date, such plugins only offer APIs to control game state and record data, requiring additional packages to train

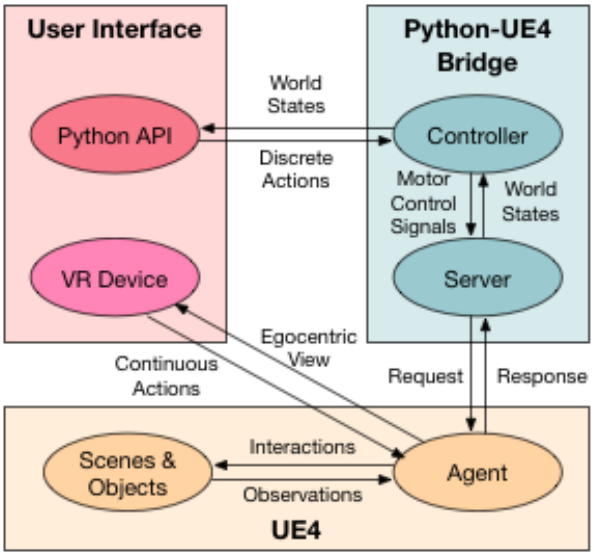


Figure 2.2: Architecture of VRKitchen. Users can either directly teleoperate the agent using VR device or send commands to the agent by Python API.

virtual agents.

2.3 VRKitchen Environment

Our goal is to enable better learning of autonomous agents for tasks with compositional goals and rich object state changes. To this end, we have designed VRKitchen, an interactive virtual kitchen environment which provides a testbed for training and evaluating various learning and planning algorithms in a variety of cooking tasks. With the help of virtual reality device, human users serve as teachers for the agents by providing demonstrations in the virtual environment.

2.3.1 Architecture Overview

Figure 2.2 gives an overview of the architecture of VRKitchen. In particular, our system consists of three modules: (1) the physics engine and photo-realistic rendering module consists of several humanoid agents and kitchen scenes, each has a number of ingredients and tools necessary for performing cooking activities; (2) a user interface module which allows users or algorithms to

(**A**) Female 1 (**B**) Female 2 (**C**) Male 1 (**D**) Male 2

Figure 2.3: Four humanoid avatars designed using MakeHuman [The].

Figure 2.4: Sample kitchen scenes available in VRKitchen. Scenes have a variety of appearance and layouts.

perform tasks by virtual reality device or Python API; (3) a Python-UE4 bridge, which transfers high level commands to motor control signals and sends them to the agent.

2.3.2 Physics Engine and Photo-realistic Rendering

As a popular game engine, Unreal Engine 4 (UE4) provides physics simulation and photo-realistic rendering which are vital for creating a realistic environment. On top of that, we design humanoid agents, scenes, object state changes, and fine-grained actions as follows.

Humanoid agents. Agents in VRKitchen have human-like appearances (shown in Figure 2.3) and detailed embodiment representations. The animation of the agent can be broken into different states, e.g. *walking, idle*. Each agent is surrounded by a capsule for collision detection: when it's *walking*, it would fail to navigate to a new location if it collides with any objects in the scene.

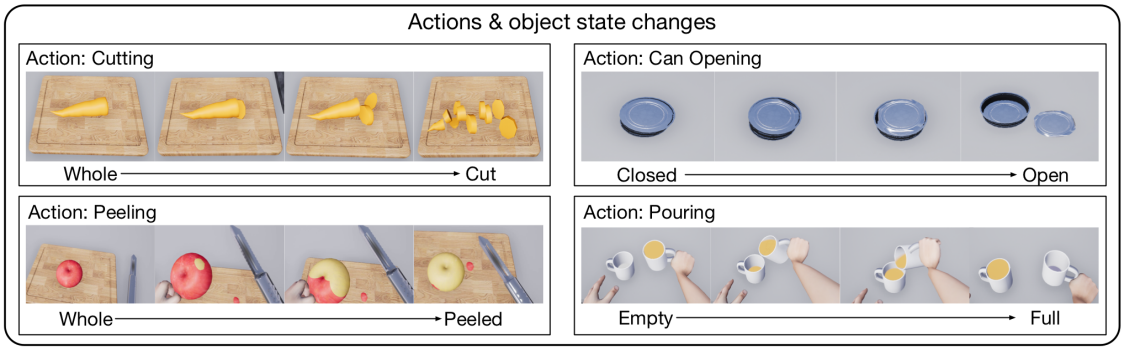


Figure 2.5: Sample actions and object state changes by making use of different tools in VRKitchen.

When it is *idle*, the agent can freely interact with objects within certain range of its body.

Scenes. VRKitchen consists of 16 fully interactive kitchen scenes as shown in Figure 2.4. Agents can interact with most of the objects in the scenes, including various kinds of *tools, receptacles* and *ingredients*. Each kitchen is designed and created manually based on common household setting. 3D models of furnitures and appliances in kitchens are first obtained from the SUNCG dataset [SYZ17]. Some of the models are decomposed to create necessary object interactions, e.g. we reassemble doors and cabinets to create effects for opening and closing the door. After we have basic furnitures and appliances in the scene, we then add cooking *ingredients* and *tools*. Instead of sampling their locations randomly, we place the objects according to their utility, e.g. *tools* are placed on the cabinets while perishable *ingredients* such as fruits and vegetables are available in the fridge. On average, there are 55 interactive objects in a scene.

Object state changes. One key factor of VRKitchen is the ability to simulate state changes for objects. Instead of showing only pre-conditions and post effects of actions, VRKitchen simulates the continuous geometric and topological changes of objects caused by actions. This leads to a great number of available cooking activities, such as roasting, peeling, scooping, pouring, blending, juicing, etc. Overall, there are 18 cooking activities available in VRKitchen. Figure 2.5 shows some examples of object interactions and state changes.

Fine-grained actions. In previous platforms [KMH17a, BPA17], objects are typically treated as a whole. However, in real world, humans apply different actions to different parts of objects.

E.g. to get some coffee from a coffee machine, a human may first press the power button to open the machine, and press the brew button afterwards to brew coffee. Thus we design the objects in our system in a compositional way, i.e., an object has multiple components, each of which has its own affordance. This extends the typical action space in prior systems to a much larger set of fine-grained actions and enables the agents to learn object-related causality and commonsense.

2.3.3 User Interface

With a detailed human embodiment representation, multiple levels of human-object-interactions are available. In particular, there are two ways for users to provide such demonstrations:

(1) Users can directly control the agent's head and hands. During teleoperation, actions are recorded using a set of off-the-shelf VR device, in our case, an Oculus Rift head-mounted display (HMD) and a pair of Oculus Touch controllers. Two Oculus constellation sensors are used to track the transforms of the headset and controllers in 3D spaces. We then apply the data to a human avatar in the virtual environment: the avatar's head and hand movements correspond to the human user's, while other parts of its body are animated through a built-in Inverse Kinematics solver (Forward And Backward Reaching Inverse Kinematics, or FABRIK). Human users are free to navigate the space using the Thumbsticks and grab objects using the Trigger button on the controller. Figure 2.6 gives an example of collecting demonstrations for continuous actions.

(2) The Python API offers a way to obtain discrete action sequences from users. In particular, it provides world states and receives discrete action sequences. The world state is comprised of the locations and current states of nearby objects and a RGB/depth image of agent's first person view. Figure 2.8 and Figure 2.9 show examples of recorded human demonstrations for tasks *pizza* and *roast meat* from a third person view.

2.3.4 Python-UE4 Bridge

The Python-UE4 bridge contains a communication module and a controller. The Python server communicates with the game engine to receive data from the environment and send requests to the agent. It is connected to the engine through sockets. To perform an action, the server sends a command to UE4 and waits for a response. A client in the game engine parses the command and applies the corresponding animations to the agent. A payload containing states of nearby objects, agent's first person camera view (in terms of RGB, depth and object instance segmentations) and other task-relevant information are sent back to the Python server. The process repeats until terminal state is reached.

The controller enables both low level motor controls and high level commands. Low level controls change local translation and rotation of agent's body, heads and hands, while other body parts are animated using FABRIK. High level commands, which performs atomic actions such as taking or placing an object, are further implemented by taking advantage of the low level controller. To cut a carrot with a knife, for example, the high level controller iteratively updates the hand location until the knife reaches the carrot.

2.3.5 Performance

We run VRKitchen on a computer with Intel(R) Core(TM) i7-7700K processor @ 4.50GHz and NVIDIA Titan X (Pascal) graphics card. A typical interaction, including sending command, executing the action, rendering frame and getting response, takes about 0.066 seconds (15 actions per second) for a single thread. The resolutions for RGB, depth and object segmentation images are by default 84×84.

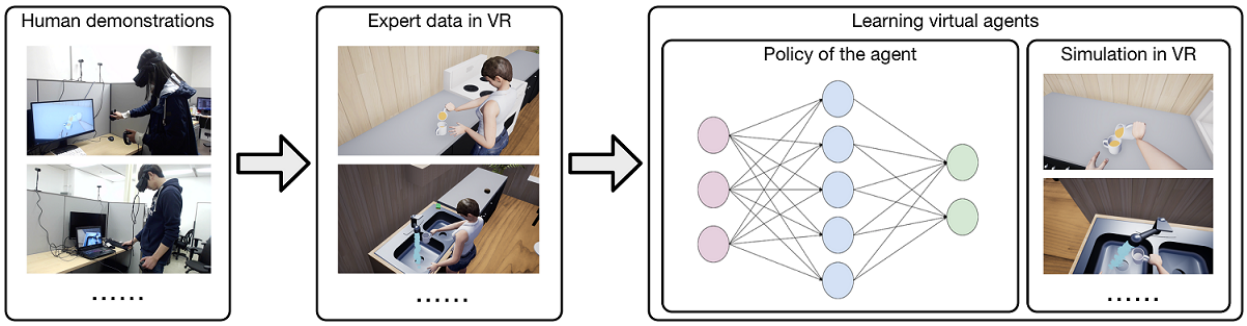


Figure 2.6: Users can provide demonstrations by doing tasks in VRKitchen. These data can be taken to initialize virtual agent's policy, which will be improved through interactions with the virtual environment.

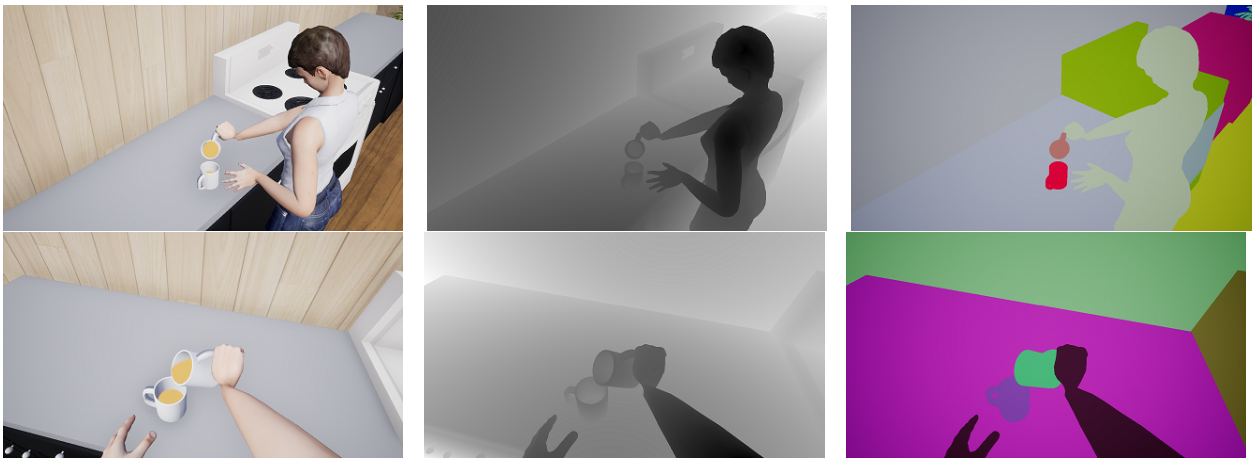
2.4 VR Chef Challenge

In this paper, we propose the VR chef challenge consisting of two sets of cooking tasks: (a) tool use, where learning motor control is the main challenge; and (b) preparing dishes, where compositional goals are involved and there are hidden task dependencies (e.g., ingredients need to be prepared in a certain order). The first set of tasks requires an agent to continuously control its hands to make use of a tool. In the second set of tasks, agents must perform a series of atomic actions in the right order to achieve the final goal.

2.4.1 Tool Use

Based on available actions and state changes in the environment (shown in Figure 2.5), we have designed 5 tool use tasks: *cutting, peeling, can-opening, pouring* and *getting water*. These tasks are common in cooking and require accurate control of agent's hand to change the state of an object. Agents would get rewards once it takes the correct tool and each time states of objects being changed. Definitions for these task are displayed as following.

- Cutting: cut a carrot into four pieces with a knife. The agent gets reward from getting the knife and each cutting.
- Peeling: peel a kiwi with a peeler. The agent receives reward from getting the peeler and



(**A**) RGB (**B**) Depth (**C**) Segmentation

Figure 2.7: Multi-modal data is available in the virtual environment. Figures in the first row show RGB, depth and semantic segmentations from a third person perspective. Figures in the second row are from the agent's first person view.

each peeled skin. Note that the skin will be peeled only if the peeler touches it within a certain range of rotation. The task finishes if enough pieces of skins are peeled.

- Can-opening: open a can with a can opener. Around the lid, there are four sides. One side of the lid will break if it overlaps with the blade. Agents receive reward from taking the opener and breaking each side of the lid.
- Pouring: take a cup full of water and pour water into a empty cup. The agent is rewarded for taking the full cup and each additional amount of water added into the empty cup. The task is considered done only if the cup is filled over fifty percent.
- Getting water: take an empty cup and get water from a running tap. The agent is rewarded for taking the cup and each additional amount of water added into it. The task is considered done only if the cup is filled over fifty percent.

In each episode, the agent can control the translation and rotation of avatar's right hand for 50 steps. The continuous action space is defined as a tuple $(\Delta x, \Delta y, \Delta z, \Delta\phi, \Delta\theta, \Delta\psi, \gamma)$, where (x, y, z) is the right hand 3D location and (ϕ, θ, ψ) is the 3D rotation in terms of Euler angle. If the grab strength γ is bigger than a threshold (0.1 in our case), objects within a certain range of avatar's

Figure 2.8: An example of human demonstrations for making a *pizza*.

Figure 2.9: An example of human demonstrations for making *roast meat*.

hand will be attached to a socket. Physics simulations are enabled on all the objects. For objects attached to agent's hand, physics simulation is disabled.

2.4.2 Preparing Dishes

Visual task planning require agents to take advantage of a sequence of atomic actions to reach a certain goal. Many challenges arise in this domain, including making long explorations and visual understanding of the surroundings. In VRKitchen, we design all atomic actions and object state changes available in several dish preparing tasks. Using these atomic actions, the agent can interact with the environments until a predefined goal is reached. Figure 2.10 shows some examples of dishes.

2.4.2.1 Atomic Actions

Each atomic action listed below can be viewed as a composition of a verb (action) and a noun (object). Objects can be grouped into three types: *tools, ingredients and receptacles*. (1) *Ingredients* are small objects needed to make a certain dish. We assume that the agent can hold at most one

ingredient at a time. (2) For *receptacles*, we follow the definition in [KMH17a]. They are defined as stationary objects which can hold things. Certain *receptacles* are called *containers* which can be closed and agents can not interact with the objects within them until they are open. (3) *Tools* can be used to change the states of certain *ingredients*. Atomic actions and object affordance are defined in a following way:

- `Take` {*ingredient*}: take an *ingredient* from a nearby *receptacle*;
- `Put into` {*receptacle*}: put a held *ingredient* into a nearby *receptacle*;
- `Use` {*tool*}: use a *tool* to change the state of a *ingredient* in a nearby *receptacle*;
- `Navigate` {*tool, receptacle*}: move to a *tool* or *receptacle*;
- `Toggle` {*container*}: change state of a *container* in front of the agent.
- `Turn`: rotating the agent's facing direction by 90 degrees.

Note that actions including `Take, put into, use`, and `toggle` would fail if the agent is not near the target object.

2.4.2.2 *Ingredient* Sets and States

Meanwhile, there are seven sets of *ingredients*, including *fruit, meat, vegetable, cold-cut, cheese, sauce, bread* and *dough*. Each set contains a number of *ingredients* as variants: for example, *cold-cut* can be ham, turkey or salami. One *ingredient* may have up to four types of state changes: *cut, peeled, cooked* and *juiced*. We manually define affordance for each set of *ingredients*: e.g. *fruit* and *vegetable* like oranges and tomatoes can be juiced (using a juicer) while bread and meat can not. *Tools* include grater, juicer, knife, oven, sauce-bottle, stove and *receptacles* are fridge, plate, cut-board, pot and cup.

Task	Goal states	Target location
Fruit juice	fruit1: cut, juiced; fruit2: cut, juiced	cup
Roast meat	fruit: cut, juiced, cooked; meat: cooked	pot
Stew	veg: cut, cooked; meat: cooked	pot
Pizza	veg: cut, cooked; cold-cut: cooked; cheese: cooked; sauce: cooked; dough: cooked	plate
Sandwich	veg: cut; sauce; cold-cut: cooked; cheese: cooked; bread: cooked	plate

Table 2.2: The goals for five available dishes. In each task, the agent should change required *ingredients* to the goal states and move them to a target location.

2.4.2.3 Goals

Based on the atomic actions defined in 2.4.2.1, agents can prepare five dishes: *fruit juice, stew, roast meat, sandwich* and *pizza*. Goals of each tasks are compositionally defined upon (1) goals states of several sets of ingredients and (2) target locations: to fulfill a task, all required *ingredients* should meet the goal states and be placed in a target location. For example, to fulfill the task *fruit juice*, two *fruits* should be cut, juiced and `put into` the same cup. Here, the target locations are one or several kinds of *containers*. Table 2.2 defines the goal states and target locations of all tasks.

2.5 Experiment

We train agents in our environments using several popular deep reinforcement learning algorithms to provide benchmarks of proposed tasks.

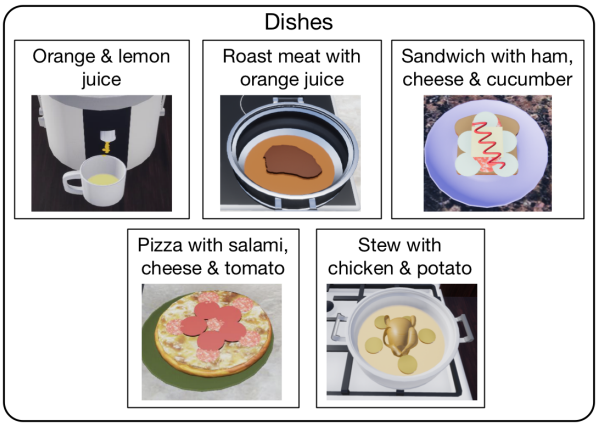


Figure 2.10: Examples of dishes made in VRKitchen. Note that different *ingredients* leads to different variants of a dish. For example, mixing orange and kiwi juice together would make *orange & kiwi juice*.

2.5.1 Experiment 1: Using Tools

2.5.1.1 Experiment Setup

In this experiment, we are learning motor controls for an agent to use different tools. In particular, five tasks (defined in 2.4.1) are available, including (a) cutting a carrot; (b) peeling a kiwi; (c) opening a can; (d) pouring water from one cup to another; (e) getting water from the tap. Successful policies should first learn to take the *tool* and then perform a set of transformations and rotations on the hand, correspond to the task and surroundings.

2.5.1.2 Results and Analysis

For five tool use tasks, we conduct experiments using three deep reinforcement learning algorithms: A2C [MBM16], DDPG [LHP15], PPO [SWD17]. The inputs are the 84×84 raw pixels coming from agent's first person view. We run each algorithm for 10000 episodes, each of which terminates if the goal state is reached or the episode exceeds 1000 steps.

Figure 2.11 summarizes the results of our experiments. We see that because of the large state space, agents trained using RL algorithms rarely succeed in most of the five tasks.

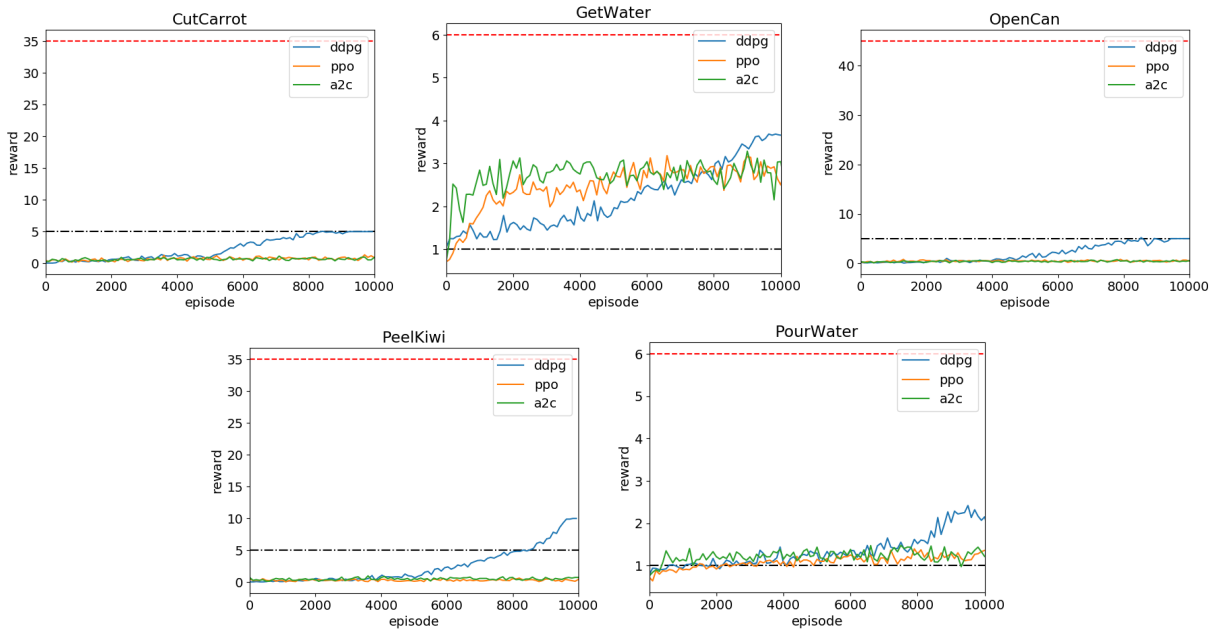


Figure 2.11: Experiment results for five tool use tasks. Black horizontal lines show rewards agents get from taking the tools, and the red lines indicate the rewards of completing the whole tasks. Each curve shows the average reward an agent receives using one of three different RL algorithms.

2.5.2 Experiment 2: Preparing Dishes

2.5.2.1 Experiment Setup

In this experiment, we study visual planning tasks, which require the agent to emit a sequence of atomic actions to meet several sub-goals. In general, successful plans should first go to locations near some *ingredients*, take them and change their states by making use of some *tools*. Particularly, tasks have three levels of difficulty:

1. Easy: First, `Navigate` to a *receptacle* R_1 and `take` an *ingredient* I_1. After that, `Navigate` to a *tool* T_1 with I_1 and `use` T_1. An example would be making orange juice: the agent should first go to the fridge and take an orange. Then it should take the orange to the juicer and use it. This task requires the agent to reason about the causal effects of its actions.

2. Medium: In addition to the "Easy" task, this task requires the agent to `take` from the *receptacle* R_1 a different *ingredient* I_2. The task ends when the agent `puts` I_1 and I_2 into a

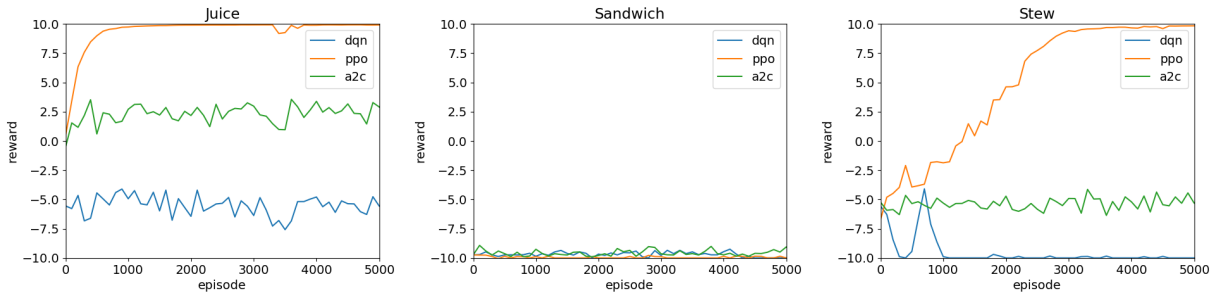


Figure 2.12: Experiment results for three dish preparing tasks. Each curve shows the average reward an agent receives using one of RL algorithms.

new `receptacle` R_2. A sample task is making beef stew: the agent should first go to the fridge and take an tomato and beef. Then it should bring the tomato to the knife and use it. Finally, the agent should put both beef and tomato into a pot. This task requires identifying various `tools`, `receptacles` and `ingredients`.

3. Hard: Compared to the "Medium" tasks, more objects are involved in hard tasks. Moreover, a longer sequence of actions is required to reach the goal state. Making sandwich is one example: *ingredients* involved are bread, tomato, ham and cheese, and an optimal policy takes about 29 steps to reach the goal states.

Atomic actions are defined as `take, put into, use, navigate, toggle, turn` (detailed definition in 2.4.2.1).

2.5.2.2 Results and Analysis

We evaluate the performance of three deep reinforcement learning algorithms (A2C [MBM16], DQN [MKS15] and PPO [SWD17]) on dish preparing tasks. We run each algorithm for 5000 episodes. We consider an episode fails if it exceeds 1000 steps.

Figure 2.12 shows the experiment results. For easy tasks (*juice*), it takes less than 1000 episodes for the best algorithm to find near-optimal solution. For medium-level tasks (*stew*), PPO [SWD17] is still able to converge after 3000 episodes. None of three RL algorithms can successfully guide

the agent in hard tasks.

2.6 Conclusion

We have designed a virtual reality system, VRKitchen, which offers physical simulation, photo-realistic rendering of multiple kitchen environments, a large set of fine-grained object manipulations, and embodied agents with human-like appearances and gestures. We have implemented toolkits for training and testing AI agents as well as for collecting human demonstrations in our system. By utilizing our system, we have proposed VR chef challenge with two sets of cooking tasks and benchmarked the performance of several popular deep reinforcement learning approaches on these tasks. We are also able to compile a video dataset of human demonstrations of the cooking tasks using the user interface in the system. In the future, we plan to enrich the simulation in our system and conduct a more thorough evaluation of current task-oriented learning approaches, including visual representation learning, world model learning, reinforcement learning, imitation learning, visual task planning, etc.

CHAPTER 3

Dialogue-Enabled Agents for Embodied Instruction Following

3.1 Introduction

Robot assistants need to understand natural language and interact with the environment. To help build language-driven embodied agents, various tasks and benchmarks have been proposed [AWT18, STG20], where the agent is given an instruction, following which it is supposed to execute the appropriate corresponding sequence of actions including navigation and object manipulation. Even with natural language instructions, such tasks are often overwhelming for the agent *on its own* due to two major challenges: 1) resolving ambiguities in natural language and grounding instructions to actions in a rich environment, and 2) planning for long-horizon action sequences and recovering from possible failures.

Humans, faced with inadequate information for a task, seek assistance from others. Similarly, embodied agents should be able to actively ask questions to humans, and utilize the verbal response to overcome challenges in understanding intent and task execution. For example, to deal with ambiguity in human instruction, clarifications are often necessary. As shown in Figure 3.1, the instruction "pick up the knife," is ambiguous when there are two knives in front of the robot – knowing the color of the intended knife helps the agent ground the instruction to its environment.

We present **DialFRED**, an embodied instruction following benchmark allowing an agent to 1) actively ask questions to the human user, and 2) use the information in the response to better complete the task. DialFRED is built by augmenting ALFRED [STG20], an existing benchmark that pairs demonstrations of common household tasks with instructions. ALFRED language instruc-

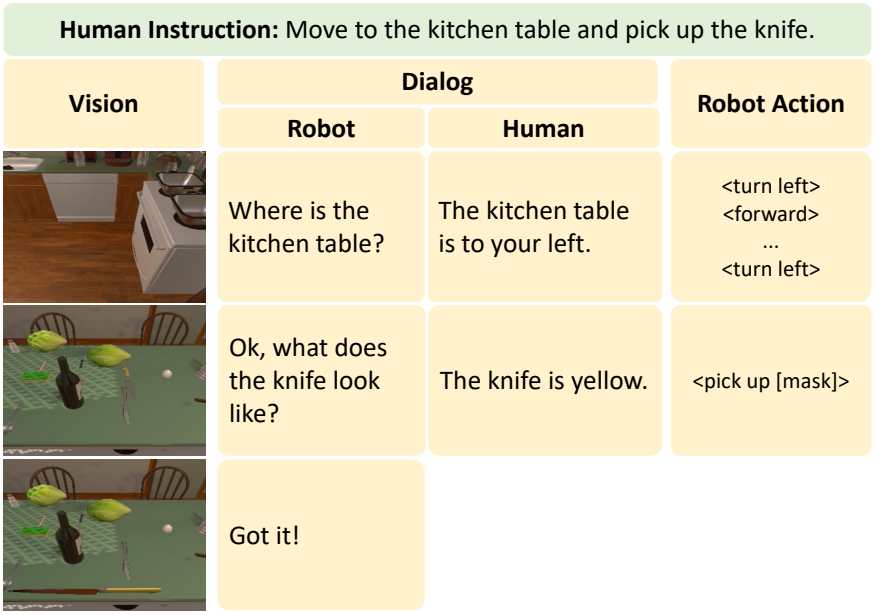


Figure 3.1: Example dialogue between a robot and a human user during task completion. The robot raises questions to obtain additional information (e.g., when the target location is not clear) and to resolve ambiguities (e.g., when facing two knives on the table).

tions are given as high level goals, e.g., *Move a knife to the sink*, and a sequence of step-by-step instructions (sub-goals), e.g., *Move forward to the center table*, *Pick up the knife*, *Walk to the sink*, *Put the knife in the sink*. ALFRED only contains 7 types of high level goals and 8 types of sub-goals. Existing work [BPF21, MCR21] has exploited patterns in ALFRED task structures, and shown that models can acheive state-of-the-art performance by classifying the task type from high-level task instructions alone, even without using step-by-step instructions. To mitigate this issue and ensure the necessity of instruction following, we build DialFRED by augmenting ALFRED for an increased number of task types. In addition, DialFRED facilitates agent-human dialogue by providing human-annotated task-relevant questions and answers.

Contributions. To enable the development and evaluation of dialogue-enabled agents in complex manipulation and navigation tasks, DialFRED consists of a) **25** types of sub-goal level tasks, compared to 8 sub-goals originally available in ALFRED; b) **53K** human-annotated task-relevant questions and answers; and c) models for a questioner-performer framework showing that adding dialogue helps to significantly improve the instruction following performance. We make Dial-

FRED publicly available and encourage researchers from related robotics disciplines to propose and evaluate their solutions to dialog-enabled embodied agents.

3.2 Related Work

Embodied Question Answering and Instruction Following. Multiple embodied AI environments are based on creating agents that learn to complete challenging tasks by interacting with their environments [CDF17, KMH17b, PRB18, GGS19, XSL20, BGK20, GSA20]. Building on these, tasks and benchmarks that require an interactive agent to extract information from the environment to answer specific questions have been proposed [DDG18, GKR18]. The other line of work that uses these environments focuses on creating agents to interpret natural language instructions and perform tasks in the environment [AWT18]. ALFRED [STG20], a recently proposed benchmark along this direction, requires the agent to complete complex household tasks by following natural language instructions. Dialogue-enabled agents in navigation or manipulation tasks have recently been proposed [TMC20, PTS21] – these focus on action prediction from dialogue history, and do not emphasize the agent's ability to ask task-appropriate questions. In this paper, we take a further step in dialogue-enabled agents by presenting a benchmark for the agent to actively ask questions and learn from the answers to better finish the task.

Task-Oriented Dialogue. In task-oriented dialogue, agents rely on skills beyond language modeling (e.g., processing multi-modal sensory data, querying knowledge bases, reasoning based on observations and knowledge [GGL18, CGS18]). Towards building robust dialogue systems, both data-driven [MSW16, HMW20] and reinforcement learning approaches [SGM16, PLL17] have been studied. Studies have shown that the ability to ask for help from humans is crucial for agent failure recovery [TKL14]. For visual language navigation, multiple works study when and how to ask for help [NDB19, ND19, RBT20, CSE20]. Household tasks however, pose greater challenges to agents compared to navigation-only tasks, due to longer action sequences, compositional task structures and irreversible object state changes. Our benchmark focuses on these complex

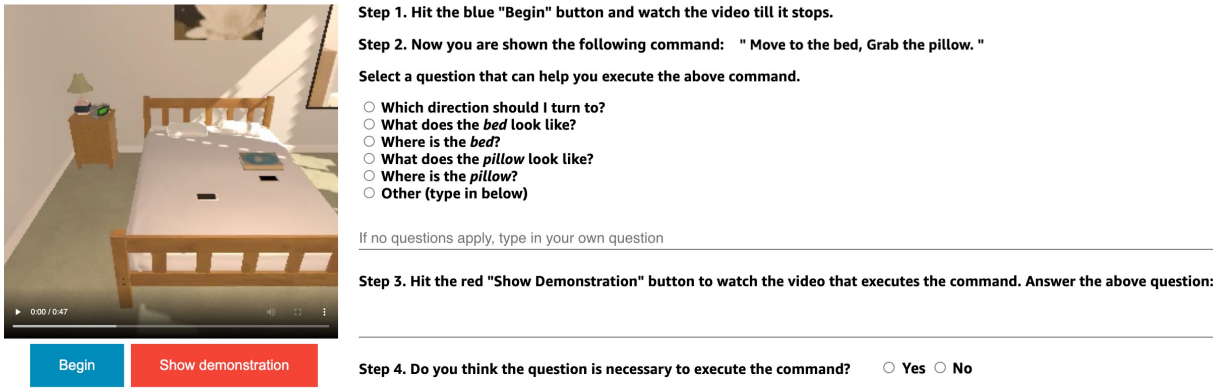


Figure 3.2: The annotation interface for hybrid data collection. The worker first clicks the "begin" button to watch a video clip showing the initial states of the environment. Given the instruction, the worker selects a question to help perform the task. Next, the worker clicks the "show demonstration" button to watch the expert demonstration on how to complete the task. The worker then answers their own question based on what they have learned from the videos. Finally, workers choose whether they think the questions and answers are necessary to help the agent carry out the command.

household tasks requiring both navigation and manipulation.

3.3 Task and Dataset

DialFRED requires an agent to follow natural language instructions and perform navigation and object manipulation to finish a household task in a virtual environment. We further enable the agent to ask questions, and use the extra information in the responses to better complete the task.

Each task instance in DialFRED is a tuple of the initial environment state, the target environment state and an instruction. The agent's goal is to perform a sequence of actions to change the environment states to the target. Given a natural language instruction, the agent can choose to ask questions to the human, or to execute physical actions in its environment based on the information in the original instructions together with the questions and answers. Physical actions include all 5 navigation actions (e.g., `Turn Left`) and all 7 manipulation actions (e.g., `Pickup`). These actions can change environment states, and some of the changes are irreversible (e.g. `Slice`). Instead of always emitting a physical action, a dialog-enabled agent may emit a question. To stan-

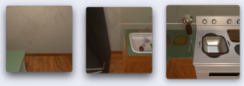
Tasks	Move & Pick	Pick & Move	Pick, Move & Slice	Move & Open
Instructions	Grab the fork.	Take the tissuebox to the floorlamp.	Cut the tomato with a knife.	Open the microwave.
Expert demonstrations				
Questions	Where is the fork?	Where is the floorlamp?	What does the knife look like?	Which direction should I turn to?
Oracle answers	The fork is behind you on the countertop.	The floorlamp is to your front right.	The knife is silver and made of metal.	You should turn right.
Human answers	It is on the kitchen island behind you.	It is in the corner of the room by the window.	It is long and silver with a blade.	Turn right and you will see the microwave.

Figure 3.3: Examples in our QA dataset. We show instructions and questions asked by humans, and answers provided by both the oracle and humans. Compared to step-by-step instructions, our augmented instructions are concise and general, thus requiring the agent to understand its current state to generate the correct action sequences.

dardize this benchmark, we provide an oracle that can answer a set of predefined types of questions (a crude 'simulated' human user). The oracle has access to the ground-truth states of the virtual environment, allowing it to provide accurate information regarding objects and tasks.

3.3.1 Hybrid data collection

We collect human annotations on Amazon Mechanical Turk for questions and answers. Each instance in the dataset is a tuple of question type $q \in Q$ asking about a specific property of an object $o \in O$ (o could be empty for questions not related to objects) and a human answer $a \in A$ for the question at the beginning of the task. Figure 3.2 displays the data collection interface. The hybrid data collection (HDC) process is as follows:

1. Each annotator watches a 10 second video clip, which displays the state of the environment right before the task. Annotators also see the original language instructions for the task.

2. The annotator then selects one pertinent question (from several predefined questions) that they think may help the agent complete the task. The annotator may also type in a question of their own choosing if none of the provided questions are a good fit.

3. The annotator watches a second video clip – this time of an expert agent performing the task.

4. The annotator answers their own question and provides feedback (yes/no) on whether asking the question was necessary in the given scenario.

To generate the predefined question choices for the annotator to choose from, we consider three types of questions Q, related to the location and appearance of the query object o that needs to be interacted with to finish the task, and the relative direction between the agent's current position and the target position to guide the navigation:

1. Location: where is o?

2. Appearance: what does o look like?

3. Direction: which direction should I turn to?

Given a natural language instruction (e.g. *Put the egg in the microwave*), we parse it and extract all the nouns (e.g. *egg* and *microwave*). We insert each noun into one of the question templates to generate questions (e.g. *Where is the egg?* and *What does the microwave look like?*).

We collected human questions and answers for 29,376 sub-goals (e.g., *Take the knife to the counter*). Two annotators provide questions and answers for each sub-goal and they reach a modest level of agreement (Fleiss' $\kappa = 0.13$) in terms of question selection. To ensure the quality of the dataset, we first remove invalid annotations when the annotation time is less than 15 seconds. We further ask trained annotators to rate annotations.

A worker is compensated \$0.25 for each HIT; the dataset collection cost is $\sim$ \$10K. The dataset is gathered over 112 rooms and 80 types of objects. Each human answer contains 6.73 words on average. A lexical complexity analysis on the human answers [Lu12] shows that the number of different words (NDW) in the answers are 7915. The lexical sophistication (the proportion of words not in the 2000 most frequent words in the American National Corpus) is 49%. Example human questions and answers are shown in Figure 3.3.

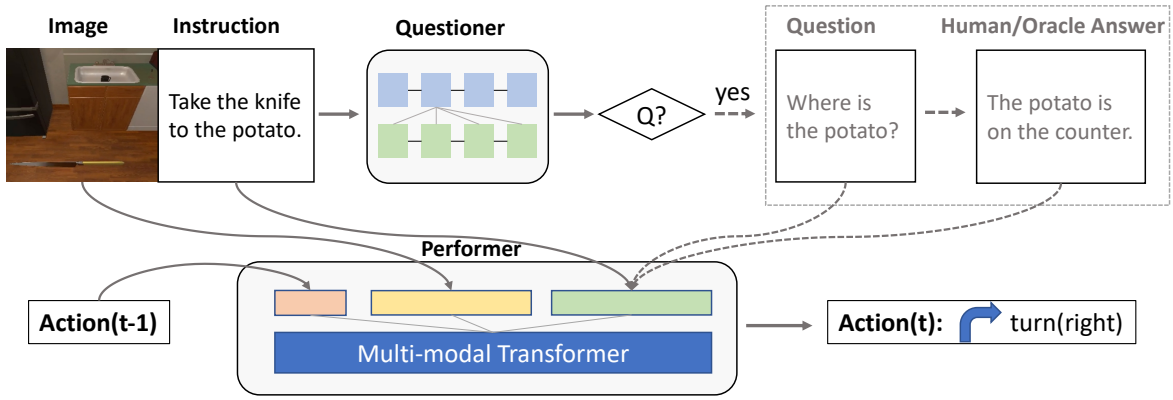


Figure 3.4: The questioner-performer architecture. The questioner generates questions based on the first person image of the agent and the task instruction. The oracle answers the question based on the scene metadata. The performer takes the image, the instruction, and question and answers as input to predict actions.

3.3.2 Generating answers

In addition to the human answers, we build an oracle that provides templated answers which can be easier for the agent to understand. In Figure 3.3, we show answers generated by the oracle for some task examples. To create the oracle, we take advantage of the ground-truth states of objects and the agent in the simulated environment: (i) to answer object location questions, we compute the direction of the object relative to the agent, and the receptacle that contains the object; (ii) to answer object appearance questions, we focus on its color and material. Object material is extracted from scene metadata in the underlying simulation environment (AI2-Thor). For object color, we extract object pixel RGB values from images, and map them to color names; (iii) for direction questions, we check the agent's location at the end of the task in the ground-truth action sequences, and compare it to the agent's initial location. Based on these metadata, we use language templates to generate answers. Some example templates include:

- Location: The *o* is to your [*direction*] in/on the [*container*].
- Appearance: The *o* is [*color*] and made of [*material*].
- Direction: You should turn [*direction*] / You don't need to move.

3.3.3 Data augmentation on ALFRED

Each task in ALFRED has a goal, split into multiple sub-goals. Each sub-goal requires the agent to manipulate some objects or move to a target location. Two types of language instructions are given. The high-level task instruction (e.g., *Move a knife to the sink*) describes the overall goal. The step-by-step instructions (e.g., *Pick up the knife*) guide the agent to complete each sub-goal. ALFRED exhibits clear patterns in both high-level task structures and sub-goal action sequences: tasks of the same type require very similar sub-goal sequences to complete them, and sub-goals of the same type (especially manipulation sub-goals) have almost fixed action sequences. The limited variety in tasks and sub-goals precludes instructions understanding – allowing models that directly classify the task type from high-level task instructions alone to perform well, without even using the step-by-step instructions [BPF21, MCR21].

To get rid of the strong patterns in both high-level task structures and sub-goal actions in ALFRED, DialFRED uses data augmentation to increase the number of task types, and focuses on instruction following at the sub-goal level to encourage learning from instructions for each task. We also introduce augmentations on the original ALFRED sub-goal instructions to add ambiguities in language so that the sequence of actions cannot be fully determined by only focusing on the instruction – reasoning based on knowledge of the environment state is needed. We show examples for DialFRED tasks and instructions in Figure 3.3.

Sub-goal augmentations. In ALFRED there are 8 only sub-goal types (*go to, pick up, put, cool, heat, clean, slice, toggle*). The action sequences required to finish these are almost fixed. For example, *cool object* always corresponds to: *open fridge, put object into fridge, close fridge, open fridge, take out object.* To increase task variations, we augment the sub-goals in two ways. First, we split an original sub-goal into multiple low level actions, each action corresponding to a new sub-goal. For example, the sub-goal *clean object* is split into three sub-goals: *put the object in the sink*, *turn on the faucet* and *turn off the faucet.* Second, we merge multiple sub-goals into a new sub-goal. For example, sub-goals *go to the fridge* and *open the fridge* are merged into a new sub-

goal which requires the agent to first go to the fridge and open it. Using these operations, we arrive at 25 sub-goal types in our augmented dataset. In our experiments, we evaluate agent performance on these sub-goals; henceforth referred to as **tasks**. To standardize the benchmark, we divide the sub-goal task instances into training and validation folds. We further divide the validation fold into *seen* and *unseen* splits depending on whether the environment presents in the training fold. This results in 34,253 tasks in the training fold, 1,296 tasks in the validation seen fold and 1,363 tasks in the validation unseen fold.

Instruction augmentation. To generate instructions for DialFRED tasks, we create instruction templates for each low level action. For tasks created from split sub-goals, we directly use the template as instruction. For tasks created from combined sub-goals, we concatenate the instructions of low level actions within the sub-goal. In addition to the step-by-step instructions describing low level actions, we generate new instructions that only describe one major action in the task. For example, for the task *go to the microwave and open the microwave* (Figure 3.3), a human may only describe the main action "open the microwave" in the instruction. The agent cannot determine the action sequence solely based on the instruction. It needs to have an understanding of its position in the room to decide which action to take next. We belive these instructions match human commands in real-world scenrios and are more challenging than the original step-by-step instructions.

3.4 Method

Our baseline for the **DialFRED** benchmark has two key components, a questioner and a performer (Figure 3.4). The questioner asks questions based on the task instruction and agent observations. The performer predicts a sequence of actions to execute in the environment based on the original instruction and the questions and answers. A good questioner knows both when to ask a question and what question(s) to ask, so that the task can be better completed by the performer. We first train the questioner using human-annotated data, i.e., give the model a good starting point by mimicking human judgement. To improve the coordination between the questioner and the performer, we fine-

tune the questioner with reinforcement learning. The questions and answers (QAs) together form a dialogue between two entities: the agent (represented by the performer and questioner model) and the human (represented by the instruction and the answers).

3.4.1 Architecture

Questioner. Our questioner model (Figure 3.5), is based on a sequence-to-sequence architecture, inspired by [RBT20]. It uses an LSTM layer [HS97] to encoder the instruction and a ResNet layer [HZR15] to encoder the visual observation. The instructions are embedded as 256 dimensional vectors and passed through an LSTM to produce context vectors and a final hidden state. The hidden state is used to initialize the LSTM decoder. At each time step the decoder is updated with the previous question token w_{i-1} and the image ResNet feature I. The hidden state is used to attend over the language and predict the next question token w_i. We pre-train the questioner using questions selected by Turkers (Section 3.3.1) on the training split. Based on Turkers feedback on whether asking the question is necessary, the questioner can also choose not to ask a question by generating a "none" token.

Performer. Our performer is based on the Episodic Transformer [PSS21], an attention-based multi-layer transformer model that encodes the full history of the instruction and QAs, visual observations and action history to predict future actions. To enable the model to handle all possible QAs from the questioner and oracle, we pre-train it on the training split by providing it with instructions and all possible questions, including a combination of question types and answers. Model parameters are optimized by minimizing the cross-entropy between predicted and expert actions.

3.4.2 Questioner fine-tuning using RL

The goal of the questioner is to ask necessary questions to help the performer finish the task. Thus we fine tune the questioner using reinforcement learning to learn *when* to ask a question and

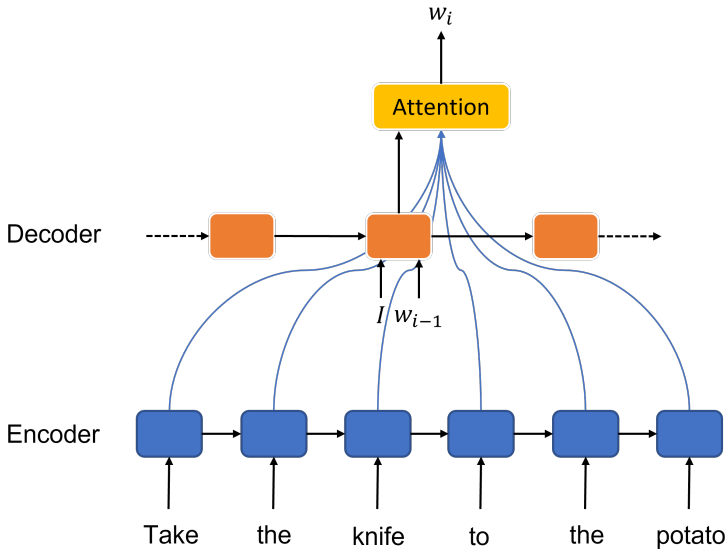


Figure 3.5: The Questioner model. Given the instruction and current image feature I, our Seq2seq model generates question tokens $w_{1:i}$.

what question(s) to ask on the validation seen split. The learning system for question asking is modeled as a Markov Decision Process, specified by a tuple $< S, A, T, R >$, where S is the state space (including number of steps and the current progress of task completion), $A = Q \times O$ is the action space of all possible questions (in our cases a combination of question types Q and target object O), $T(s'|s,a)$ is the transition function encoding how the performer can advance the task given the question and its corresponding answers, and $R(s,a)$ is the reward function encoding the reward for each (s,a) pair.

Asking questions has costs: a balance must be struck between the number of questions and performance gain. The reward function addresses this trade-off. Some questions generated by the questioner cannot be answered by the oracle, e.g. the appearance of task-irrelevant objects. Thus we add a penalty for invalid questions. We adopt the following reward structure: reward for task completion $r_{suc} = 1.0$, penalty for each step $r_{step} = -0.01$, penalty per question asked $r_q = -0.05$, penalty per invalid question $r_{invalid} = -0.1$.

Our seq2seq questioner model can be viewed as a policy network $\pi_\theta(s,a) = p(a|s;\theta)$ mapping each state vector s to a stochastic questioning policy. The optimal value of θ is found by mini-

#	Expt setting	Seen SR	Unseen SR	Seen PWSR	Unseen PWSR	NQ
1	Instruction only	25.4	18.3	18.4	11.4	0
2	All QAs	43.4	32.0	31.2	19.9	3.24
3	Random QA	39.9	27.9	28.4	17.3	0.81
4	Random MC	46.6	29.5	**35.5**	18.7	0.52
5	RL begin	47.3	32.7	33.5	20.1	0.37
6	RL anytime	**47.8**	**33.6**	34.2	**20.4**	0.71

Table 3.1: Performance of the baselines. Seen SR and unseen SR represent the success rate on valid seen and valid unseen splits. PWSR is the path weighted success rate. NQ is the number of questions asked by the questioner. The best results are highlighted in boldface.

mizing the actor-critic loss [SB18] using stochastic gradient decent. The performer model is not updated during questioner fine tuning.

Expt setting	Unseen SR	Unseen PWSR	NQ	Loc Perc.	App Perc.	Dir Perc.
Human	-	-	0.66	0.72	0.22	0.06
RL anytime	33.6	20.4	0.71	0.65	0.14	0.21
RL Loc perturb	32.2	19.8	0.66	0.47	0.15	0.38
RL App perturb	32.4	19.9	0.40	0.92	0.04	0.04
RL Dir perturb	33.0	20.4	0.61	0.51	0.45	0.04
Random	28.4	17.3	0.81	0.36	0.31	0.33
Random Loc perturb	26.0	16.0	0.81	0.36	0.31	0.33
Random App perturb	26.1	16.1	0.81	0.36	0.31	0.33
Random Dir perturb	26.2	15.8	0.81	0.36	0.31	0.33

Table 3.2: Ablation study by perturbing the oracle. We start from two settings: a questioner that has been fine-tuned for asking questions at any time and a questioner that asks a random question at the beginning. We perturb the oracle by not providing answers for one question type 50% of the time. Loc Perc, App Perc and Dir Perc represent the percentage questions about object locations, object appearance and directions respectively.

3.4.3 Heuristic-based questioner

Inspired by [CSE20], we implemented a questioner based on model confusion. The idea is that if the performer is not confident, the output action distribution would have high entropy; model confusion could be a good heuristic to know when to ask a question. We sort action probabilities in decreasing order; an agent is confused if the minimum difference between the top two actions is

Experiment setting	Seen SR	Unseen SR	Seen PWSR	Unseen PWSR	NQ
RL anytime (Fixed 1)	**51.9**	**34.7**	**36.3**	**21.2**	21.39
RL anytime (Fixed 5)	47.8	33.6	34.2	20.4	0.71
RL anytime (Fixed 10)	46.3	32.4	32.7	19.9	0.36
RL anytime (MC)	47.1	33.0	33.3	20.4	0.31

Table 3.3: Effect of question timing. We manipulate the number of steps the performer rolls out before the questioner can ask the next question. For *(Fixed 1)*, we modify the rewards ($r_{invalid} = -0.01, r_q = -0.002$) to promote question asking. For *(MC)*, the questioner asks questions based on the performer model confusion.

less than a threshold ε throughout the action sequences:

$$\min_t(p^t_{sorted}[0] - p^t_{sorted}[1]) < \varepsilon \tag{3.1}$$

The threshold is used to control the degree of confusion for asking questions. In practice, we set the confusion threshold $\varepsilon = 0.5$ in the experiment.

3.5 Experiments

We evaluate the baseline models on our dataset. We terminate the episode when it exceeds 1000 steps or has more than 10 failed actions. For evaluation, we focus on the success rate of tasks.

3.5.1 Evaluation metrics

We evaluate model performance using success rate and path weighted success rate. To understand the trade-off between the number of questions asked and task performance, we measure the number of questions throughout the task.

3.5.2 Baselines

We implemented 6 baselines and evaluated their performance on our augmented dataset (Table 3.1). In all baselines we use the Episodic Transformer model as the performer. In baselines 2–6, the performer is trained using imitation learning on expert action sequences with language instructions and QAs. In baselines 5–6, the questioner is pre-trained on human dialogues in the training split, and fine-tuned using reinforcement learning on the valid seen split. Baseline details are itemized below:

1. In this baseline, no questions are allowed, the performer is trained to predict the action sequences based on the language instruction and visual observations.

2. In addition to the instruction, the performer gets all valid QAs at the beginning of the task, including a combination of all three question types (i.e. location, appearance and direction) and objects mentioned in the instruction.

3. Questions are sampled randomly based on type: 25% for each of the 3 types and 25% for no question. Given a selected question type, questions and answers for all relevant objects are given to the performer in addition to the instructions.

4. When the action sequences generated by the performer satisfy the model confusion criterion (Equation (3.1)), the performer is randomly provided with a valid QA as input.

5. The questioner is fine-tuned using reinforcement learning to learn whether to ask a question and what question to ask at the beginning of a task. The performer uses the QA to generate the action sequence to finish the task.

6. Similar to baseline 5, the questioner is fine-tuned using reinforcement learning, but now it can ask questions in the middle of the task. Given the instruction and previous QAs, we roll out the performer for 5 steps, following which the questioner is allowed to generate new questions and thus get new answers.

3.5.3 Results

We display (Table 3.1) the task performance (success rate) on validation seen and unseen splits for all baseline models. Comparing the results of baselines 1–3, we see that adding QAs to the instructions improves task performance on both splits. Comparing the results of baselines 2–5, we see that the fine-tuned questioner achieves the best performance, with a smaller number of questions. Comparing the results of two reinforcement learning baselines (5,6), we see that enabling the agent to ask questions in the middle of the task improves performance at the cost of more questions and answers. In addition, the *random MC* baseline achieves reasonable performance on the valid seen split, but not on the valid unseen split. Since the performer is pre-trained on the training split, which has the same scene as the valid seen split, and is given random combinations of different types of QAs as input, it is not surprising that the *random MC* baseline does not generalize well to unseen environments.

3.5.4 Ablation Study

Perturbed oracle. We perform an ablation study (Table 3.2) by perturbing the oracle. For each question type, we limit the oracle to provide answers for 50% of the asked questions. The original fine-tuned questioner (*RL anytime*) has a similar number of questions (and distribution) as the human data it is pre-trained on – most questions are about object locations. With the perturbation, the model asks slightly fewer questions. We observe that the fine-tuned questioner model adapts to the deficient oracle. For example, after perturbation on object location answers, the questioner asks 28% fewer location questions and 81% more direction questions instead. Looking at model performance after perturbation, we find that perturbing the location answers reduces the performance the most. This result matches the large proportion of location questions asked by both humans and the learning-tuned model, indicating that location answers are probably most useful for task completion. We perform the same perturbations on the oracle for the *random* questioner. Comparing the performance of models before and after perturbation, we see that the drop in SR caused

by the perturbations for the learning-tuned questioners are smaller than the drops for the *random* questioners. The difference can be explained by the learning-tuned questioner's ability to adjust the proportion of questions to adapt to the deficient oracle.

Question timing. To understand how question timing affects performance, we change the number of action steps the performer executes before allowing the questioner to ask a question. We add a setting, *RL anytime (MC)*, which allows the questioner to ask questions based on model confusion (Equation (3.1)) during fine-tuning. The results (Table 3.3) show that reducing the number of steps between questions leads to slightly better performance, but it also requires the oracle to answer significantly more questions. Comparing the results of model confusion with fixed timing, we find that it achieves reasonable performance at relatively low cost.

3.6 Conclusion

We presented **DialFRED**, a dialogue-enabled embodied instruction following benchmark that allows an agent to actively ask questions while interacting with the environment to finish a household task. DialFRED is generated by augmenting ALFRED to increase task and language variations. It includes an oracle to answer questions and a human-annotated dataset with 53K task-relevant questions and answers – a potential resource to model how humans ask and answer task-oriented questions. To tackle DialFRED, we propose a questioner-performer baseline (and variants) wherein the questioner is pre-trained with the human-annotated data and fine-tuned with reinforcement learning. Experimental results show that asking the right questions leads to significantly improved task performance. Extending existing embodied instruction following benchmarks with dialogue is a promising avenue of research towards truly interactive embodied agents. Along these lines, we posit that the general framework of oracle-guided reinforcement training and the hybrid data annotation method we employ may be useful to "dialogue-enable" other embodied instruction following tasks.

CHAPTER 4

Evaluating the Effects of Assisting Interfaces on Drivers' Situational Awareness in Autonomous Vehicles

4.1 Introduction

Autonomous vehicles (AVs) have the potential to revolutionize the transportation industry. Despite the rapid development of the autonomous driving (AD) system, fully automated cars are still not available on public roads. Currently, some vehicles on the market are equipped with advanced driver assistance systems (ADAS) that allow partially automated driving, or SAE Level 2 (L2) automation [int18]. While drivers can briefly enjoy feet-free and hands-free driving under certain driving situations at this level of automation, they are still required to monitor the traffic conditions and prepare for sudden maneuvers and possible takeover requests. As a result, it is crucial to maintain the driver's situational awareness (SA) when interacting with the AD system and avoid the out-of-the-loop problem [End88a].

With the goal of improving drivers' SA and trust, researchers investigated various ways of communication to convey internal information to the drivers. The challenge is that showing additional information to the drivers can increase their cognitive load and cause distractions [Hel14]. Showing too much information not only prevents drivers from paying attention to the most critical information during driving [AC18], but is also against the main motivation of developing the AD system, i.e. reducing driver workload. Therefore, we believe that a smart user interface (UI) should be able to strike a balance between the amount of information provided and the driver's limited attention and cognitive load.

Figure 4.1: Our driving simulator is composed of a steering wheel and two pedals mounted on a cockpit, and three 55-inch displays showing the front and side views of the virtual environment.

Results from previous studies showed that highlighting hazardous objects via augmented reality (AR) based UI is a promising way to increase drivers' SA [LLR18, CER21]. Nevertheless, those works mainly focused on evaluating the effects of highlighting on SA **across all objects**. We believe a smart UI should optimize the highlighting for each object to maintain a proper workload and help the driver be aware of the potential hazards that are prone to be ignored. Therefore, we need to understand the effects of highlighting on **each specific object**, depending on the object characteristics.

Specifically, in this work, we distinguish objects by three properties 1) locations (relative to the driver), 2) types (i.e. pedestrian or vehicle), and 3) traffic densities and evaluate the effect of highlighting considering those factors.

We implemented object highlighting via a UI on a driving simulator based on Unreal Engine 4 (UE4), and conducted an in-person study (N=20) on the simulator to investigate the effect of highlighting on drivers' attention allocation and SA for each object in an urban environment.

The main contributions of this paper are:

- We implemented an AR-based UI in a driving simulator to inform drivers of the AD's perception capabilities by highlighting hazardous objects. We focused on urban intersections because of their complex traffic conditions, which can be demanding for the drivers to monitor.

Figure 4.2: This is a forward event intersection with high traffic density, corresponding to the event in Figure 4.6**A**. We highlight objects using bounding boxes on the user interface: red for pedestrians and blue for cars. In addition, we also display the ego vehicle's current speed and heading direction with yellow texts and arrows in the middle. During the study, this concatenated screenshot is separately shown on three displays to simulate the field-of-view of a driver in the real world (see Figure 4.1).

Figure 4.3: To evaluate drivers' SA, we pause the simulation and hide all road users. On top of the background road scene of the intersection, we display several regions and ask users to choose which regions were occupied by pedestrians or vehicles.

- We designed and conducted a simulator experiment to evaluate the impact of highlighting on drivers' object-wise SA and attention allocation. Specifically, we designed a novel Situational Awareness Global Assessment Technique (SAGAT) protocol with temporal variations to measure the same driver's SA changes before and after the highlighting in two identical intersections to better understand the effect of highlighting on SA. Objects' locations and movements at intersections are discretized based on spatial distance and eccentricity.

- We carefully analyzed the effect of highlighting with the AR interface in different conditions, including a combination of object types, object locations and traffic densities.

The results of our study suggest that the effects of highlighting on perception-level SA highly depend on object properties and traffic densities. We believe the results pave the way for a smart

UI that can selectively highlight objects to improve SA for drivers of AVs, leading to more safety in driving and monitoring partially autonomous vehicles.

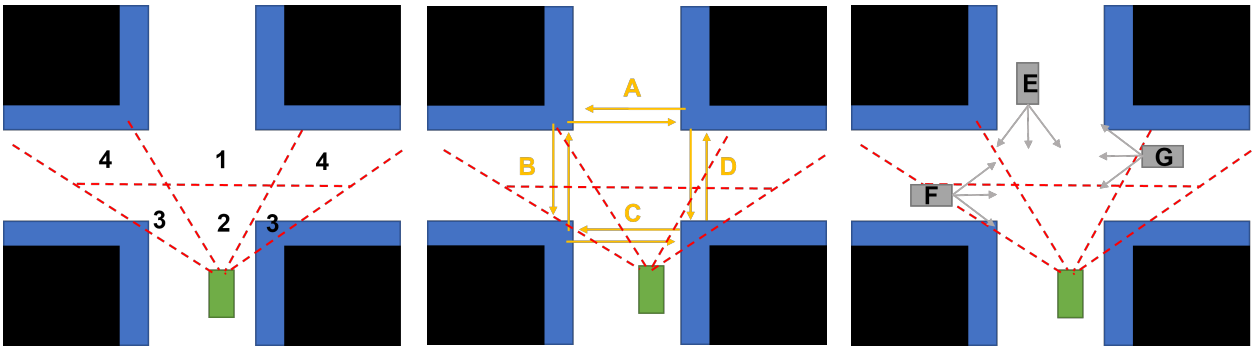
4.2 Related Work

4.2.1 Situational Awareness Measurements

SA can be generally understood as knowing what is going on around you [End88a]. Over the years, various methods have been proposed to measure SA. They can be categorized into objective measurements (e.g. SPAM [DHT98] and DAZE [SMJ17]) and subjective measurements (e.g. SART [Tay90] and SARS [WH94]). SAGAT is a widely-known technique to measure SA objectively [End88b]. During a SAGAT session, the display is frozen at selected times and participants are asked to answer questions to measure SA. An advantage of SAGAT is that participants are unable to prepare for the questions in advance, thus minimizing the possibility of attention bias. Studies suggested that SAGAT is a technique with a high degree of reliability [EB94] and validity [End90]. Apart from direct measurements, SA can also be inferred with indirect measurements, including eye gaze behaviors and takeovers [BL10, YKD18, ZMM21]. For a more comprehensive review of situational awareness measurements, we refer readers to this survey [SSW06].

4.2.2 Ways to Improve Driving Situational Awareness

Driver's SA at SAE L2 or L3 automated driving has been widely studied for years. Previous works have shown that driver's SA can be influenced by a wide range of factors, such as age [Bol01], driving experience [WSB16] and working memory [JH10, HHB14]. Since driver's SA plays an important role in driving safety, different methods have been proposed to enhance SA during driving. Recently, the idea of assisting human-machine interface (HMI) has generated much interest [MKT18]. Studies that examined the effects of AR windshield display (WSD) interface found significant effects on driver's SA, when highlighting potential driving threats [LLR18] or



(**A**) Area Discretizations based on spatial distance and eccentricity relative to the ego vehicle shown in green.

(**B**) Pedestrian movements. Yellows arrows show target pedestrians' all possible movements across the intersection.

(**C**) Vehicles movements. Gray rectangles are target vehicles' initial locations and arrows show possible moving directions.

Figure 4.4: We first discretize an intersection into 4 areas based on spatial distance and eccentricity relative to the green ego vehicle. According to area discretizations, we then discretize pedestrians' and vehicles' all possible movements near the intersection. For example, pedestrian A crossing the top of the intersection is considered moving in area 1, while car F going straight on the left will be moving in areas 2 and 3.

common traffic objects (e.g. cars, pedestrians, and traffic signs) [CER21, WWH20]. The SA improvement was observed at both perception level [TJ19] and comprehension and projection levels [PTF16]. In these works, the focus is to evaluate the effect of HMI on the driver's average SA across all traffic objects between experimental groups. Thus the driver's SA is measured within each treatment group without distinguishing between objects [PTF16, LLR18, CER21]. Our work takes one further step in this direction: 1) we focus on evaluating the effect of HMI on driver's SA for each object, distinguished by their locations and types, and 2) propose a novel SAGAT protocol with temporal variations by measuring the same participant's SA before and after the treatment to better analyze its effects toward a specific object.

4.3 Method

In this section, we introduce our simulation study. We start with participant and apparatus in Section 4.3.1 and talk about the AD system and AR cues in Section 4.3.2. We discuss how we

discretize the object locations in Section 4.3.3 and the details of the scenarios in Section 4.3.4. Then we describe how we measure attention allocation and situational awareness of the driver in Section 4.3.5. Finally, we go through the whole procedure of our study in Section 4.3.6.

4.3.1 Participants and Apparatus

A total of 20 participants (12 males, and 8 females) from the San Francisco Bay area completed the study. Their age ranged from 20 to 49 years old. To be eligible for the study, each participant was required to have had a valid license for more than two years and drive more than 5,000 miles (8,047 km) per year.

As shown in Figure 4.1, we used a medium-fidelity driving simulator built with AirSim [SDL18], a plug-in for UE4, to conduct all driving sessions. Also, Tobii Pro Glasses 3[1] were used to collect the participant's eye-tracking data.

4.3.2 SAE L2 AD System and AR Assisting Cues

The Wizard of Oz method is used to simulate realistic AD driving. To more realistically emulate a functioning AD system, we had an expert driver drive the ego-vehicle through the premeditated route in the simulated environment. The drive was recorded by saving all pedal and steering inputs to an AD file, which then provided realistic autonomous car behavior to participants. To ensure consistency of the AD driving behavior, all driving data were recorded from a single expert driver. The driving behaviors were further reviewed by two researchers, and routes were practiced to ensure consistency. During the driving, the participant was requested to indicate their take-over intention by pressing the brake pedal.

The AR cues were developed and assigned to highlight specific road users within intersections. As shown in Figure 4.2, AR graphics used for highlighting were consistent in shape, i.e. a series of 3D bounding boxes that formed a cubic region surrounding a specific type of road users (blue

[1] https://www.tobiipro.com/glasses3

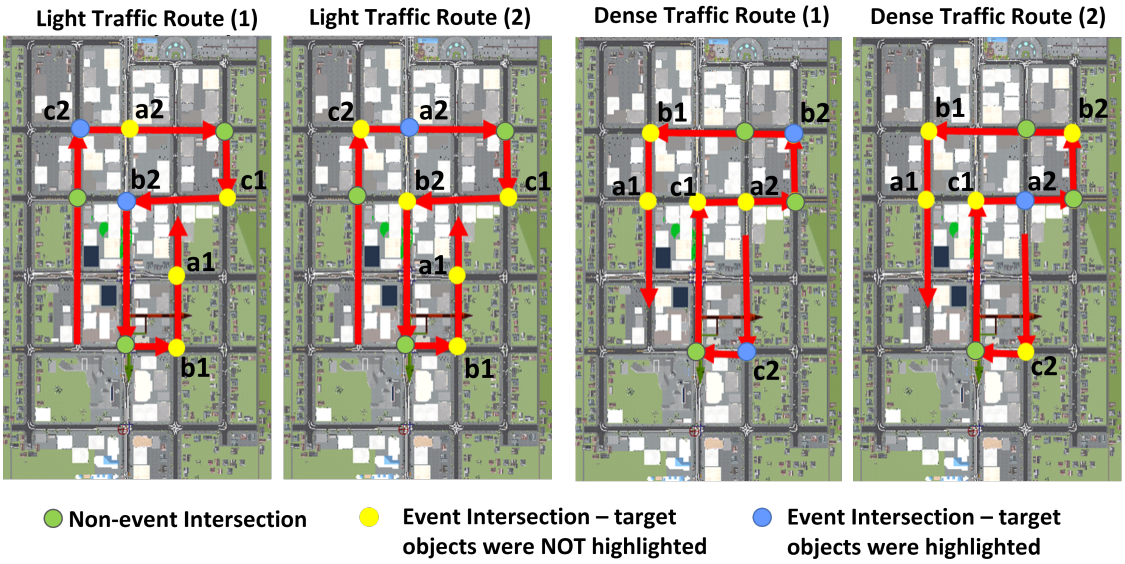
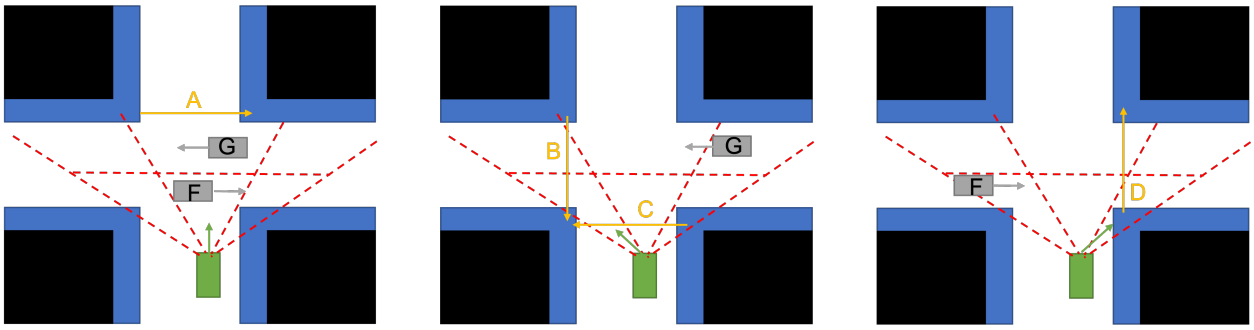


Figure 4.5: Drives and intersections. Light traffic route 1 (LT1) and light traffic route 2 (LT2) are of low traffic density, while dense traffic route 1 (DT1) and dense traffic route 2 (DT2) are of high traffic density. Blue dots represent event intersections where the target objects are highlighted. Yellow dots represent event intersections where the target objects are not highlighted. Green dots are non-event intersections where we ask dummy SAGAT questions to reduce the learning effect of SA in event intersections. In intersections a1, b1 and c1, SAGAT questions are asked before the treatment (highlighting or not highlighting). In intersections a2, b2 and c2, SAGAT questions are asked after the treatment.

for vehicles and red for pedestrians).

4.3.3 Object Location Discretization

The human visual field can be commonly divided into three major regions: foveal, parafoveal, and peripheral. The foveal region extends out to an angle of 1 degree and the parafoveal region from 1 to 5 degrees [NL80, QCF19]. Those two together are commonly referred to as the central vision, and the peripheral region encompasses the remainder of the visual field. Many researchers have noted that, as a result of the inhomogeneity of the visual system, attention allocation and awareness are strongly affected by the target eccentricity and spatial distance. Detecting a target far away in the peripheral as opposed to nearby and central vision requires longer search times and more eye movements [CEC95].



(**A**) Forward event intersection (**B**) Left event intersection (**C**) Right event intersection

Figure 4.6: We display the locations and heading directions of target objects in three types of event intersections. For clarity, distractor objects are not shown here. The green rectangle is the ego vehicle. Gray rectangles are other vehicles' locations and gray arrows show their moving directions. Yellows arrows show pedestrians' movements across the intersection.

Considering both an object's spatial distance (i.e. near or far) and eccentricity (i.e. center or marginal) relative to the driver in the ego vehicle, we categorized the object positions in an intersection into four types of areas: the top center area (area 1), the bottom center area (area 2), the bottom left and bottom right areas (area 3) and the top left and top right areas (area 4), as shown in Figure 4.4**A**. Since the pedestrians and cars move across different areas, their movements were also discretized (Figure 4.4**B**, Figure 4.4**C**).

4.3.4 Driving Scenario and Events

The driving scenario is an urban environment in daylight conditions with a posted speed limit of 25 mph. Events are triggered when the ego vehicle comes near event intersections. During an event, the ego vehicle first stops before the intersection due to a stop sign or a flashing red traffic light. The vehicle then waits until the other road users have passed the intersection following the traffic rule. While the vehicle is waiting, the driver is asked to continuously monitor the surroundings and take over the control if the AD system has made an unexpected or dangerous move. The cars and pedestrians across intersections are consistent in appearance. Intersections are of similar sizes ($L = 15.3 \pm 2.0$ m, $W = 14.4 \pm 1.2$ m).

We inserted a SAGAT pause while the ego-car is waiting for other traffic at an intersection. The participant is asked to answer the positions of objects, including cars and pedestrians (Level 1 SA). We are particularly interested to study the SA of some of the objects that can potentially collide with the ego vehicle (the future trajectory of the object will intersect with the ego-car's future trajectory). We carefully design the event timing so that only these objects are located in certain regions at the SAGAT pauses. We refer to them as target objects and other objects as distractor objects. During the SAGAT pause, the simulation is frozen and other situational objects (i.e. pedestrians, vehicles and traffic lights) are hidden in the simulator. Meanwhile, several regions would be displayed on the blank scene, as shown in Figure 4.3. The driver is asked to speak about all the regions where he/she believes there were pedestrians and/or vehicles. In particular, we have designed three types of events that correspond to three heading directions (i.e. forward, turning left and turning right) of the ego vehicle.

Two driving routes with opposite directions and different traffic densities were designed in this experiment (See Figure 4.5). We change the traffic density of routes, by adding or removing distractor objects from event intersections. The average number of total objects, including both distractors and target objects, is 10 for dense traffic (DT) route and 5 for light traffic (LT) route drives in event intersections. Based on each route, two drives with different event designs were developed (LT1 & LT2 and DT1 & DT2). Figure 4.6 illustrates the design of each type of event and target objects in the event:

- Forward intersections (a1 and a2 in Figure 4.5). For intersections where the ego vehicle is heading straight, the target objects are pedestrian A, vehicles F and G (Figure 4.6**A**). The SAGAT pause occurs while pedestrian A is crossing the intersection on the top (at area 1) and vehicles F and G are going through the intersection in the middle (at area 1 and area 2 respectively).

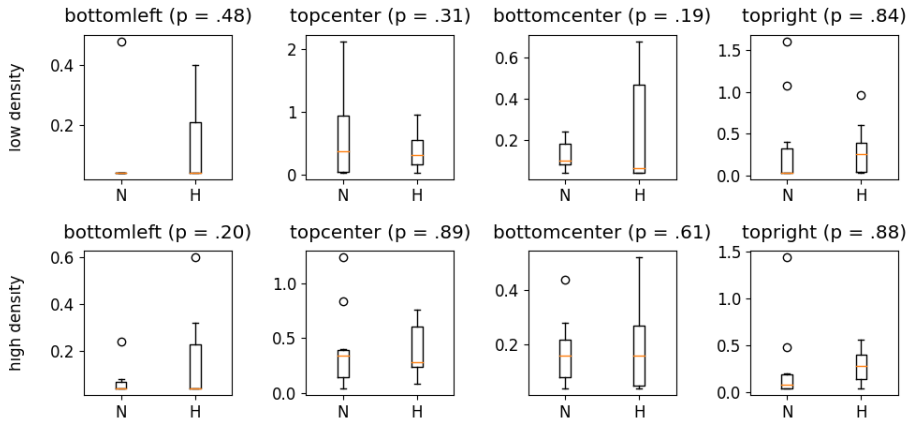
- Left intersections (b1 and b2 in Figure 4.5). For intersections where the ego vehicle is heading left, the target objects are pedestrians B, C and vehicle G (Figure 4.6**B**). The SAGAT pause occurs while pedestrian B is crossing the intersection on the top left (at area 4), C is on the

bottom center (at area 2) and vehicle G is waiting on the top right (at area 4).

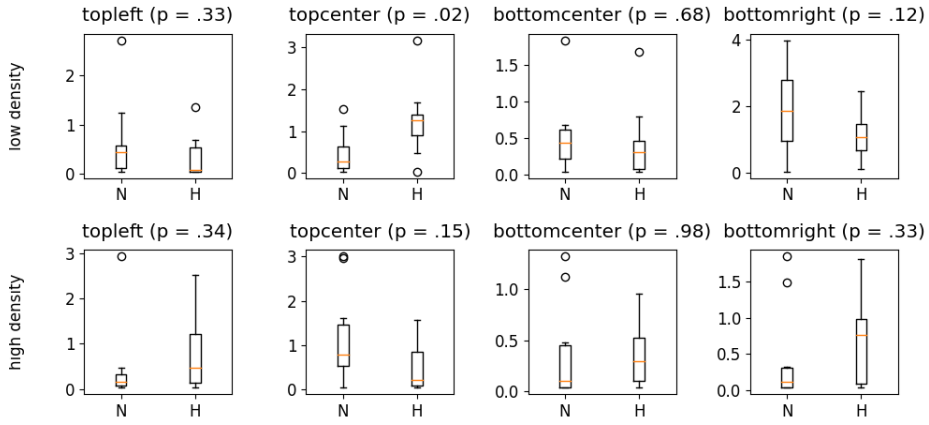
- Right intersections (c1 and c2 in Figure 4.5). For intersections where the ego vehicle is heading right, the target objects include the pedestrian D and vehicle F (Figure 4.6**C**). The SAGAT pause occurs while vehicle F is waiting on the bottom left (at area 3) and pedestrian D is crossing the intersection on the bottom right (at area 3).

Each participant was randomly assigned to one of the four experimental groups and experienced one of the four combinations of two drives (routes) in sequential order: group (i) LT1 and then DT2; group (ii) DT2 and then LT1; group (iii) LT2 and then DT1; and group (iv) DT1 and then LT2. Figure 4.5 illustrates the event/non-event intersections for all four drives. For each drive, there are six event intersections, including two left intersections, two right intersections and two forward intersections. In some event intersections (blue dots in Figure 4.5), the target objects are highlighted. In order to reduce the learning effect between the two intersections of the same type within a drive, we also design a non-event intersection (green dots) between them: in these non-event intersections, the driver also needs to answer a dummy SAGAT question, such as the heading direction of the vehicle and the color of the traffic light. The average duration of a drive is 15 minutes.

Our goal is to understand how highlighting would change the SA and attention for different objects. Thus we measure SA from the same driver at two different timings in one type of intersection: 1) before treatment (i.e. highlighting or not highlighting the target object) and 2) 1 second after the treatment. To reduce the order effect, we implemented two separate intersections with the same type of events, but with different timing of the SAGAT pauses. For example, in drive LT1, the SAGAT pause in one of the forward event intersections (a2) is delayed by 1 second, while the SAGAT pause in the other forward event intersection (a1) is not delayed. The purpose of this delay is to study how highlighting or not highlighting the target object within this delayed period would change drivers' SA. Since the delayed and undelayed intersections have exactly the same event, we can compare the driver's SA responses to better understand the effect of highlighting.



(**A**) Fixation time on cars



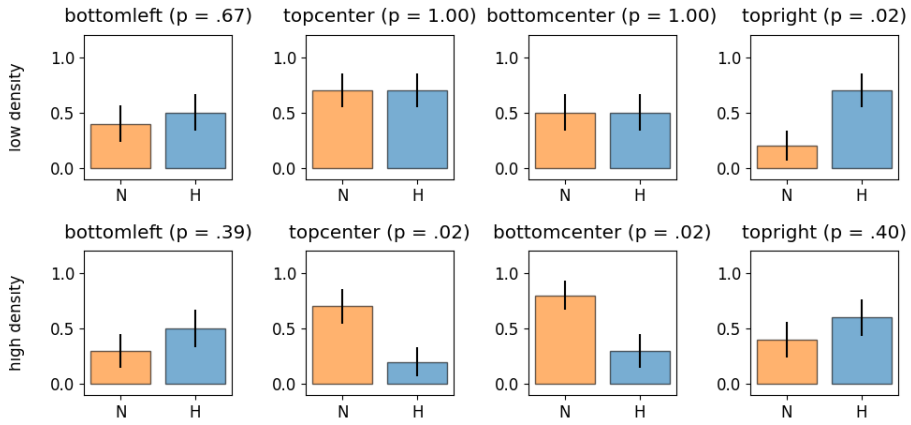
(**B**) Fixation time on pedestrians

Figure 4.7: Drivers' fixation time (in second) on each car and pedestrian given traffic density. "N" represents non-highlighting results and "H" represents highlighting results. We report the p-value between highlighting conditions for each object.

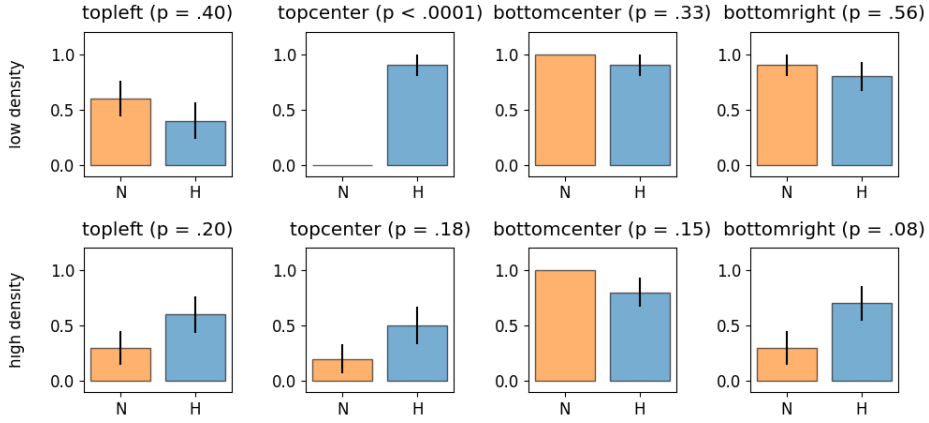
4.3.5 Dependent Variables

Attention allocation. Attention allocation is strongly associated with situational awareness. To form situational awareness, one needs to perceive and process the environment [End88a]. However, the limited capacity of human attentional resources in combination with the excessive attentional demands in a dynamic driving environment can result in a loss of situational awareness. To study attention allocation, one well-established measurement is to track human fixation behavior. We collect drivers' eye-tracking data with Tobii Pro Glasses 3. We also annotate the target objects in each event intersection using Vatic [VPR13]. Based on the finding that humans can recognize information in the fovea (2.5 deg [NL80, QCF19]) within 120 ms [RC07], we define that the driver has fixated on an object if the gaze has stayed within 2.5 degrees from the center of the object for more than 120 milliseconds.

Situational awareness. To measure drivers' situational awareness, we adopt the SAGAT technique and ask drivers questions about the location of pedestrians and vehicles during the pauses in the event intersections (Figure 4.3). We focus on the Level 1 SA (perception) since it is the most fundamental one. Participants were asked to select all the regions that they think had a pedestrian or a vehicle the moment before the SAGAT pause. Two images with region highlightings, each corresponding to pedestrians and vehicles (e.g., Figure 4.3 is for pedestrians), are presented in sequence with a corresponding SAGAT question. They were also asked to provide a confidence level (from 0 to 100) for each region they specified, which is further discretized to low confidence (0-50) and high confidence (51-100). Regions that are not selected by the participant are treated as low confidence. The order of SAGAT questions (i.e. pedestrian or vehicle) is randomized to reduce the order effect. During the analysis, we study the SA response on the discretized regions that are occupied by the target objects.

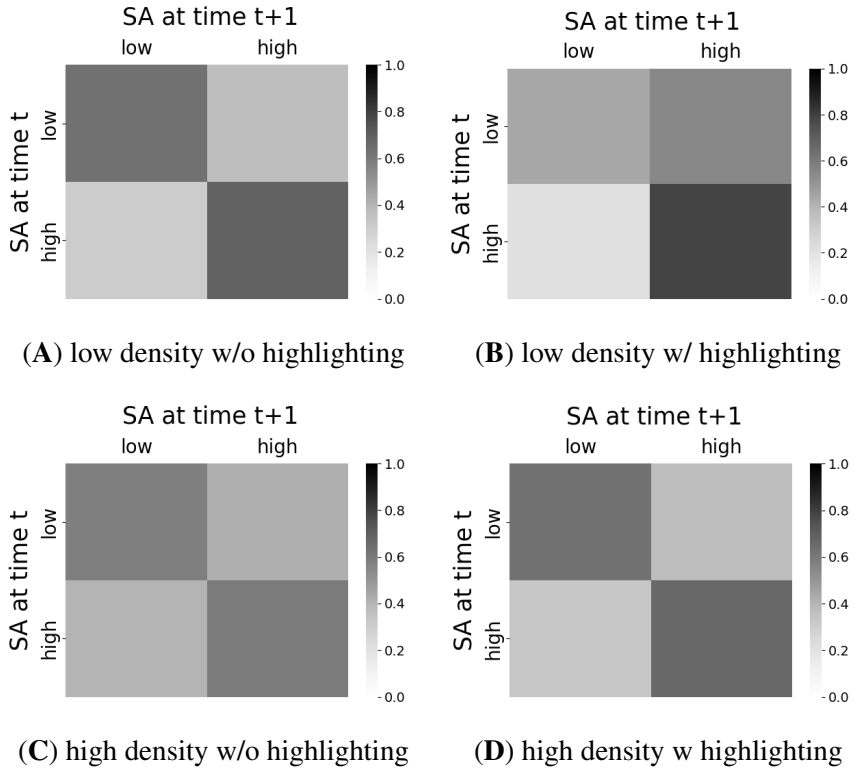


(**A**) Response accuracy on cars



(**B**) Response accuracy on pedestrians

Figure 4.8: Drivers' SAGAT question response accuracy in delayed intersections. "N" represents non-highlighting results and "H" represents highlighting results. We report the p-value between highlighting conditions for each object.



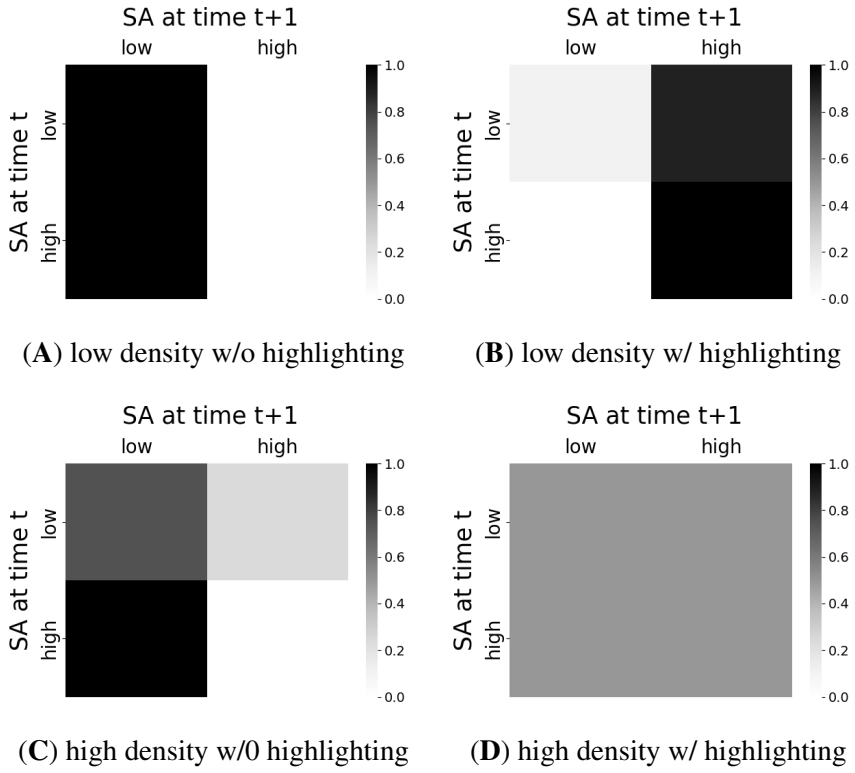
(**A**) low density w/o highlighting (**B**) low density w/ highlighting

(**C**) high density w/o highlighting (**D**) high density w highlighting

Figure 4.9: SA transition conditioned on traffic density and highlighting across all objects. ”SA at time t” represents drivers’ SA response before the treatment, while ”SA at time t+1” is for SA response after the treatment. The shade of each region represents the proportion of the samples falling into each category. Darker color represents a higher proportion.

4.3.6 Procedure

Participants first completed a pre-study survey to provide demographics and driving experience. They also filled in a questionnaire designed to evaluate their trust in automation [JBD00]. The moderator then gave each participant a brief introduction to the study and set up the Tobii glasses and the driving simulator. The study began with a practice drive where the participant was asked a sample question during a SAGAT pause. During the practice drive, the participant was asked to take over control of the vehicle using the brake pedal whenever they felt uncomfortable with the AD system. The participant was also given a chance to practice answering the SAGAT question at an intersection as well as indicating their intention to take over. After the practice drive, the participant was randomly assigned to an experimental group, and went through two standard drives



(**A**) low density w/o highlighting (**B**) low density w/ highlighting

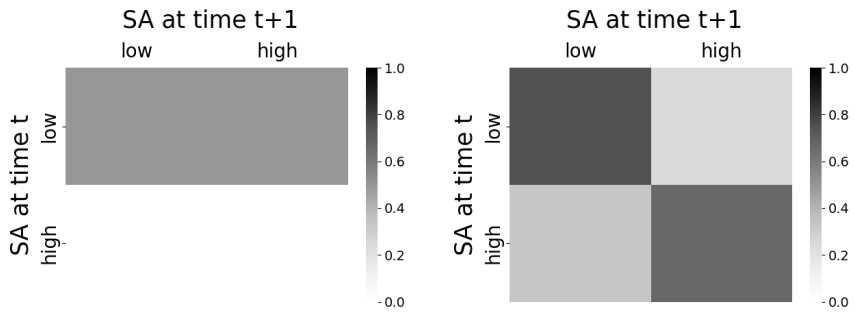
(**C**) high density w/0 highlighting (**D**) high density w/ highlighting

Figure 4.10: SA transition for the top center pedestrian (pedestrian A in Figure 4.6**A**). The shade of each region represents the proportion of the samples falling into each category. Darker color represents a higher proportion.

of different traffic densities. During a drive, each participant experienced six event intersections and two non-event intersections (Figure 4.5). In each event intersection, the participant was asked about the locations of vehicles and pedestrians during the SAGAT pause (Figure 4.3).

4.4 Results

In this section, we present the results of our study, analyzing how highlighting objects changes drivers' attention allocation and situational awareness during their interaction with an AD system based on the data collected from the driving simulator experiment.



(**A**) low density w/o highlighting (**B**) low density w/ highlighting

(**C**) high density w/o highlighting (**D**) high density w/ highlighting

Figure 4.11: SA transition for the bottom center car (car F in Figure 4.6**A**). The shade of each region represents the proportion of the samples falling into each category. Darker color represents a higher proportion.

4.4.1 Attention Allocation

For attention allocation, we analyze the driver's fixation time on target objects. Since our goal is to study how highlighting would change the driver's attention, we focus on the driver's fixation during the delayed period (Section 4.3.4). We show the results for specific cars and pedestrians in Figure 4.7 at different traffic densities. Running a pairwise t-test, we found a significant effect ($p = .02$) for highlighting the top center pedestrian, i.e. the pedestrian A, when the traffic density is low. We don't find the same trend for top center pedestrians at high traffic density.

4.4.2 Situational Awareness

SA response accuracy. We analyze drivers' responses to the SAGAT questions at delayed intersections, when different highlighting conditions have been applied to the target objects (Figure 4.8). Across all objects, driver's SA on highlighted objects ($M = 0.60, SD = 0.49$) are higher than the unhighlighted ones ($M = 0.52, SD = 0.50$), but the difference is not statistically significant ($p = .14$). For cars (Figure 4.8**A**), we observed a significant difference between highlighting conditions for the top right car (car G) in a low traffic density environment($p = .02$) and for the top center car and bottom center car (cars F and G) in high traffic density environment($p = .02$). For pedestrians (Figure 4.8**B**), we only found a significant change in SA for top center pedestrians during light traffic routes ($p < .0001$). The results indicate that highlighting can improve the SA for the top right car and the top center pedestrian at low traffic density, while decreasing the SA for the bottom center car and top center car at high traffic density. The Pearson correlation coefficient between fixation time and SA response accuracy is $r = .12$ ($p = .03$), indicating a weak correlation between attention and SA.

SA transition. We first analyze the transition of drivers' SA from undelayed pauses to delayed pauses across all objects, when we apply different highlighting conditions to the target objects during time t and time $t + 1$ (Figure 4.9). Given low traffic density, for drivers with an initial low SA on target objects, highlighting leads to SA improvement (from low to high) for 55.3% of the drivers, compared to 36.8% in the non-highlighting conditions. For drivers with an initial high SA on target objects, we found that those in the highlighting conditions are more likely to maintain their high SA (78.6%) compared to drivers in the non-highlighting conditions (69.0%) for low density. Similarly, when the traffic density is high, highlighting also helps more drivers maintain high SA (66.7%) compared to the no highlighting (59.5%). Running a two-sample proportion test, however, we didn't find any significant effect of highlighting on SA transition for either traffic density across all objects.

Looking at the SA transition for specific objects, we found from a proportion test that for the

top center pedestrian, highlighting can significantly increase the proportion of drivers that improve low SA ($p = .0007$) and maintain high SA ($p = .03$) compared to the control condition when the traffic density is low (Figure 4.10). On the contrary, for the bottom center car, we found that highlighting actually decreases the proportion of drivers that maintain high SA ($p = .02$) when the traffic density is high (Figure 4.11). We didn't find any significant difference in SA transition between highlighting conditions for other objects.

4.5 Discussion

The results indicate that the effect of highlighting varies a lot depending on the situation. Highlighting can significantly improve SA on certain objects at low traffic density. However, it can also decrease drivers' SA of some objects at high traffic density. These findings can provide guidance in selecting which object to highlight for the UI to improve the driver's SA while driving and monitoring SAE L2 or L3 AVs.

4.5.1 Attention Allocation, Workload and SA

Driving is a visual and motor control process. Thus, drivers' attention allocation and workload play important roles in establishing their SA. Previous works have proposed quantitative methods to model the interplay between attention allocation, workload and SA [WMA08,SXD14,LWZ14]than on unhighlighted ones. Specifically, the attention allocation process can be largely influenced by the salience of an object and workload [Wic02]. A high workload can result in attention tunneling and negatively impact SA. In our study results, highlighting the cars in the center of the driver's field of view significantly decreases SA when the traffic density is high, while the difference is not significant when the traffic density is lower. This can probably be explained by (i) the driver's high workload given the dense traffic ii) the highlighting AR cues induce additional workload (iii) the fact that cars in the center are already very salient even without highlighting. These reasons can also explain why significant SA improvement was found for the top center pedestrian (which is not

visually salient and easy to be ignored by the driver) at low traffic density and why improvement is not significant at higher traffic density (due to the driver being overwhelmed by the dense traffic). We believe these results shed light on designing object-specific AR cues on human-machine interfaces.

4.5.2 Comparison with Previous Studies

Previous works focus on evaluating the driver's average SA across all traffic objects in different experimental groups. By controlling a specific object's spatial characteristic in the driving simulator, we are able to further study the transition of the user's SA on the object before and after the highlighting. Results from previous works [PTF16, LLR18, CER21] showed that using an AR interface could improve drivers' average SA across all objects. Thanks to the unique study design covering objects properties and traffic conditions in common intersections as well as the proposed SAGAT protocol with temporal variations, we are able to see significant positive effects of AR cues on some objects and negative effects on some other objects. These results extend knowledge of the community on the effects of AR cues beyond specifically-designed scenarios and hand-picked objects, showing how different objects can benefit from the AR cues in more general driving scenarios.

4.5.3 Limitations and Future Work

Our UI is implemented in a driving simulator, which enables us to control the timing of events accurately. Driving scenarios in the real world are more complex and have more variety than our examined scenarios. In reality, the AR cues can be implemented by detecting vehicles and pedestrians from sensors and highlighting them using bounding boxes on the AR-HUD. In addition, every participant experienced two similar event intersections - one before and one after the highlighting. We ask dummy SAGAT questions in non-event intersections between the two events intersections to reduce the learning effect, but the effect may not be canceled off com-

pletely. Additionally, we measure SA using SAGAT, which is known to be highly reliable [EB94]. The drawback is that SAGAT requires the participant to memorize the objects and thus can also increase the workload [FLH20]. Non-intrusive SA measures can be considered in a future study to ensure an accurate measure of drivers' workload when interacting with an AD system. Finally, in the future work, we plan to consider other object features (e.g. object colors and speed) and the differences in the intersections' background environment, which are also likely to affect SA.

4.6 Conclusion

This work aims to investigate the effects of highlighting objects with an AR interface on drivers' perception-level SA for SAE L2 or L3 AVs under different circumstances, including object types, locations and traffic densities in urban environments. We conducted a user study in a driving simulator ($N = 20$). The results show that highlighting has a positive impact on SA when the traffic density is low and the highlighted object has originally low visual saliency, and sometimes causes a reduction in SA when the object is already very salient even without highlighting during dense traffic. This work extends the knowledge on methods to improve driver's situational awareness for autonomous vehicles, and enables the development of a smart driver-assistance interface that can selectively highlight objects to improve SA for drivers monitoring partially autonomous vehicles.

CHAPTER 5

Joint Mind Modeling for Explanation Generation in Human-Machine Collaboration

In recent years, there has been a great amount of success on building powerful artificial intelligence (AI) systems to solve complex tasks [LFD16, BPS17]. As highly autonomous robots are being developed, there is a growing need to make them quickly understood to avoid consequences caused by misunderstanding [Gun17]. However, existing robot systems are often not human compatible – i) they do not understand humans' minds and ii) they are just black boxes to humans too. Such limits prevent the AI systems from working with humans effectively.

Inspired by studies on the Theory-of-Mind [PW78, Den89], we believe that a crucial step towards building human compatible systems, particularly for human-robot collaborations, is to understand human activities and their underlying mental state. As a motivating example, consider a robot chef helping a human make salads in the kitchen shown in Figure 5.1. Even when the robot understands how to perform the task on its own, it would be challenging to finish the task efficiently without having a shared mental model with its human partner. For making the salad, the robot believes the plate should be picked up by the user while the human agent believes the other way. If the robot can identify such discrepancies between different agents' mental states, it can generate explanations to mitigate the differences and encourage the correction of sub-optimal human behavior.

To this end, we propose a framework that improves human-robot teaming performance through explanations. With a graph-based representation, the robot can maintain the mental states of both team members during a highly-structured collaborative task. The robot can then generate explana-

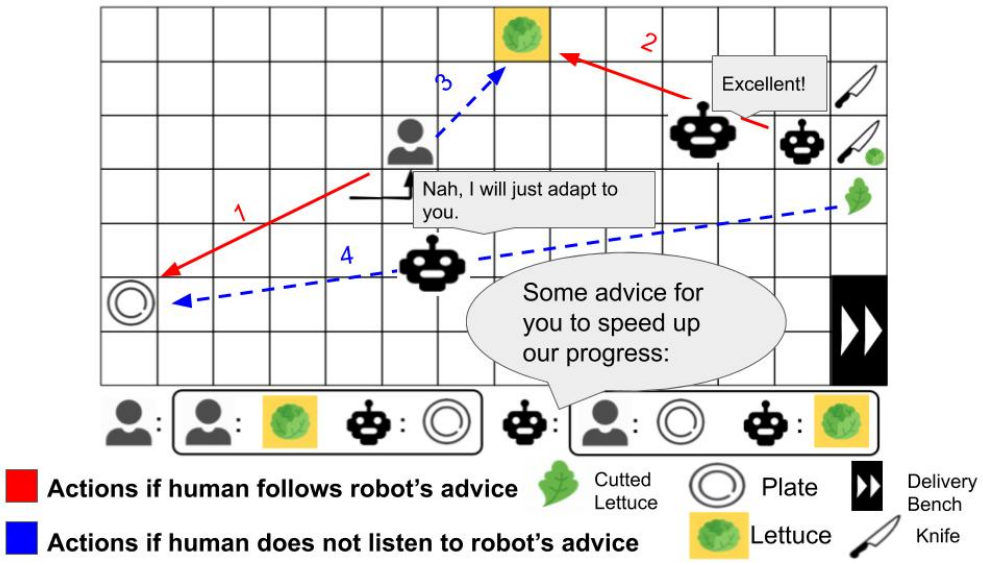


Figure 5.1: The task *making salad* requires team members to take three lettuce from the basket and cut each one with a knife, before it can be put into the plate and served. After the first lettuce has been cut, the robot is cutting the second one. The robot can identify human's sub-optimal behavior (taking new lettuce from the basket) before generating explanations to the human.

tions when difference between mental states is detected, which implies sub-optimal user behaviors.

In summary, the main contribution of this paper is three-fold:

- We design a real-time collaborative cooking game as an online user study system and develop an evaluation protocol, which can be accessed from our website.
- We propose to understand complex human activities using an action parsing algorithm based on an And-Or graph task representation, which allows the robot to infer human mental states in complex environments.
- Based on the inferred human mental state, we propose an explanation generation framework. Experiments on a real-time cooking task show that our approach successfully improves user perception of the robot and leads to better human-robot collaborations.

5.1 Related Work

Human-aware planning. Designing robots that can work with humans has been widely studied by researchers. Most of the prior works hope to create robots to better understand and adapt to human collaborators. [LHF16] evaluates a collaborative task allocation framework based on a Bayesian inference of human intention. [HRA16] proposes a formulation of the value alignment problem assuming the robot learning an unknown human reward function. Optimal solutions can be achieved when the human demonstrates active teaching behavior. To deal with sensor uncertainty and task ambiguity in a collaborative assembly task, [HBV14] uses an And-Or tree structure as the task representation, which is similar to our approach. When sub-optimal user behavior are encountered, [RDL18] proposes to learn the incorrect human internal dynamics model via inverse RL and then perform an internal-to-real dynamics transfer to assist users in shared-autonomy tasks. Our framework differs from this line of research in that we also aim at improving humans' understanding of robots' models using communicative actions. Such two-way understanding will further help human-robot collaborations.

Goal-driven explainable AI. In contrast to data-driven XAI which improves understanding of "black-box" machine learning algorithms given input data, goal-directed XAI typically explains the behavior of an agent or robot for a specific task [LMS17, ANC19, Mil19], in order to increase model transparency [SRK19], human's trust [WPH16] or task performance [XD15]. Some of the works achieve this aim by enabling robots to directly generate easy-to-understand motions [DS13, KHD18] or task plans [ZSK17]. Other works, similar to ours, focus on using explicit communication to change user mental state, e.g., updating users' incorrect reward functions [TAH19], correcting users' false belief or misunderstanding about the environment [GZ18, SSK18], resolving the disagreement between collaborators' actions [NKF18] or providing users with necessary knowledge about the current situation [DA16]. Compared to these work that often require offline training with humans or theoretical assumptions on the human models, this paper takes a direct approach to generate explanations solely based on an online estimation of human model and

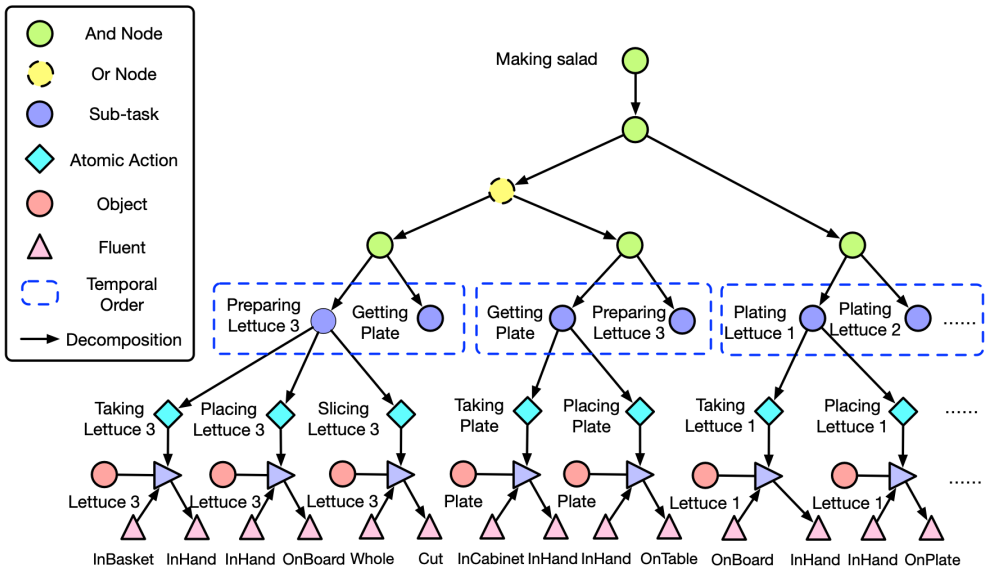


Figure 5.2: The hierarchical mind model for the collaboration task, "making salad", represented by an AoG. The And node represents temporal relations between sub-tasks. The Or node represents two possible ways for the team to finish the tasks. Each terminal node (diamond) denotes an atomic action that would cause certain fluent changes (triangles) for objects.

knowledge of the task structure. The experiment results show our approach is empirically effective in an ad-hoc human-robot teaming settings [SKK10] where pre-coordination is not available.

5.2 Single Agent Mind Model

And-Or graphs (AoGs) have been widely used for robot task planning [XSX16,SGR17,LZS18] and human activity modeling [TPZ13, SXR15]. As a hierarchical representation, a spatial-temporal-causal And-Or graph (STC-AoG) encodes a joint task plan and corresponding spatial, temporal, and causal relations an agent could have about the task [XSX16]. In this work, we propose to use a STC-AoG as a unified representation of a robot's knowledge and plan regarding the task as well as the inferred human's knowledge and plan. An example of a single-agent plan for *making salad* is in Figure 5.2.

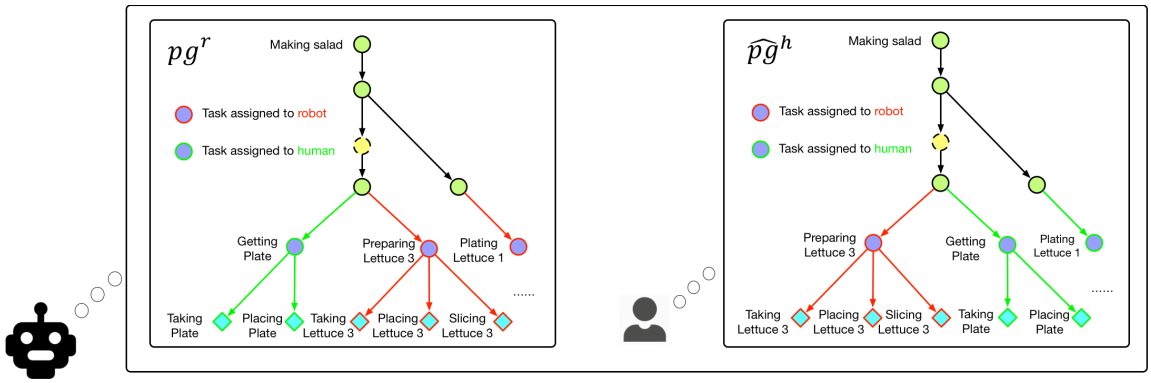


Figure 5.3: Robot mental state pg^r and inferred human mental state $\hat{pg}^h$ represented as parse graphs.

5.2.1 STC-AoG as a Hierarchical Mind Model

In general, an And-Or Graph consists of nodes and edges. The set of nodes includes Or node, And node, and Terminal node. Each **Or node** specifies the Or relation: only one of its children nodes would be performed at a given time. An **And node** represents the And relation and is composed of several children nodes. Each **Terminal node** represents a set of entities that cannot be further decomposed. The edge represents the top-down sampling process from a parent node to its children nodes. The root node of the And-Or tree is always an And node connected to a set of And/Or nodes. Each And-node represents a sub-task which can be further decomposed into a series of sub-tasks or atomic actions.

In this paper, the graph $G =< A, F, T, V, R, P >$ is formally defined as the following:

- A is a set of terminal nodes. Each node corresponds to an atomic action $a \in A$.
- F is a set of object states essential to the task, including possible pre-conditions and post-effects of atomic actions.
- $T : F \times A \rightarrow F$ is a set of transition rules that represent state changes caused by atomic actions.
- V is a set of non-terminal nodes, which can be further decomposed into two sets: the And nodes S and the Or nodes O. Each sub-task corresponds to an And node s, which encodes a temporal

relationship between its children. An Or node o forms a production rule with an associated probability, i.e. you may choose one of its children each weighted with a certain probability.

- R is the set of production rules.
- P is the set of probabilities on production rules.

Causal relation. Causal knowledge represents the pre-conditions and the post-effects of atomic actions. We define it as a fluent change caused by an action. Fluent $f \in F$ can be viewed as some essential properties in a state x that can change over time, e.g., the temperature in a room and the status of a heater. For each atomic action, there are pre-conditions characterized by certain fluents of the states. E.g., an agent cannot successfully turn on the heater unless it is plugged in. As the effect of an action, certain fluents would be changed, and the state x would evolve to x'. For example, if someone turns on a heater, the temperature of the room will be higher (and the heater would be on). It is formulated as one of the transition rules T.

Temporal relation. Temporal knowledge encodes the schedule for an agent to finish each sub-task. It also contains the temporal relations between atomic actions in a low level sub-task. The sub-task preparing salad, for example, consists of taking salad, placing it onto the cutting board, and using the knife.

Spatial relation. Spatial knowledge represents the physical configuration of the environment that is necessary to finish the task. In our case, to make the salad, an agent needs to know the locations of ingredients (e.g., lettuce), tool benches (e.g., basket, cutting board), delivery benches, etc.

5.2.2 Parse Graphs as Mental State Representations

During the collaboration, an agent can use parse graphs to represent the mental states of itself or the other agent. A parse graph is an instance of an And-Or Graph, each of its Or nodes selects one child node. Figure 5.3 shows two parse graphs represent the robot and human's plan for the situation shown in 5.1. In our case, the parse graph $pg_t =< s_t^h, s_t^r, a_t^h, a_t^r, f_t^h, f_t^r >$ is one possible

plan for both agents to finish the task. Particularly, the root node leads to a selection of individual sub-tasks (s_t^h, s_t^r) as sub-goals assigned to human and robot agent. To achieve these sub-goals, agents perform atomic action (a_t^h, a_t^r) based on their belief of current fluent (f_t^h, f_t^r).

5.2.3 Joint task planning by parsing STC-AoG

To construct the mental state representation for the robot, we design an algorithm based on STC-AoG parsing to select the optimal task plan for the team.

Given a set of sub-tasks S necessary to complete the joint task, the objective is to minimize the total task completion time by assigning a sub-task to either a robot or human agent, without violating any latent constraint:

$$\begin{aligned} &\min_{x_s^v, \tau_s} \max_{v \in \{r,h\}} \sum_{s \in S} x_s^v \delta_s^v \\ &\text{s.t. } x_s^v \in X_{\text{feasible}}, \tau_s \in \Gamma_{\text{feasible}}. \end{aligned} \tag{5.1}$$

where x_s^v is a binary variable indicating whether to assign sub-task s to agent v, and τ_s is a continuous variable representing the finishing time for the sub-task s. Constant δ_s^v represents the amount of time for agent v to finish the sub-task s. X_{feasible} and Γ_{feasible} represent the set of valid assignments that satisfies latent causal constraints, e.g., an agent cannot hold two objects at the same time; a sub-task can be performed only if pre-conditions are met; after all assigned sub-tasks have been completed, the final state should satisfy the goal requirement.

We search for the optimal task plan via a dynamic programming algorithm. Starting from the initial state f_b, we make valid sub-tasks assignments and simulate new intermediate state f_e based on the state transition function T. By updating the current optimal consumed time and the corresponding sub-task assignment vectors for every intermediate state, our algorithm will finally reach the optimal plan for the entire task. During the updating process, we also record the sub-task assignment vectors for previous states, in order to generate the whole optimal assignment $\{x_s^v\}_{s=1,\ldots,|S|}$ and completion time $\tau_1, \ldots, \tau_{|S|}$ for each sub-task. After the task plan is computed,

the robot's mental model is represented by a parse graph, as shown in the left part of 5.3: each sub-task in the task plan indicates a sub-goal that an agent needs to achieve at the time being. Sub-tasks are further connected with a sequence of corresponding atomic actions, which have certain pre-conditions and post-effects.

5.3 Joint Mind Modeling for Human-Robot Collaborations

Our goal is to enable efficient human-aware collaboration for a human-robot team. Specifically, robots need to understand human agents based on their actions and decide whether the team is moving in the right direction. We propose to model the robot mental state pg^r and the human mental state pg^h.

5.3.1 Mind Models for Human and Robot

We treat the robot's mind as the oracle, i.e., it contains all necessary spatial, temporal, and causal information the team needs to finish the task. For example, at any given time t, the robot has a certain expectation of (i) current low level sub-goals (s_t^h, s_t^r) both agents should be pursuing; (ii) the actions (a_t^h, a_t^r) agents should perform; (iii) whether current object fluents satisfy pre-conditions of such actions, and what would be the post-effects.

It is also necessary to model the user's mind, which acts as a strong inductive bias in predicting user activities. As the user's mental state pg_t^h is not directly available to the robot, we propose to infer it from user behavior and the history of communication.

5.3.2 Human Mental State Inference

Based on the observed user behavior, we infer the most likely human mental state $\hat{pg}^h$, including the belief, goal and action plans. On a high level, this inference process uses observed user actions and communication history to infer human mental state. Specifically, given the And-Or graph G

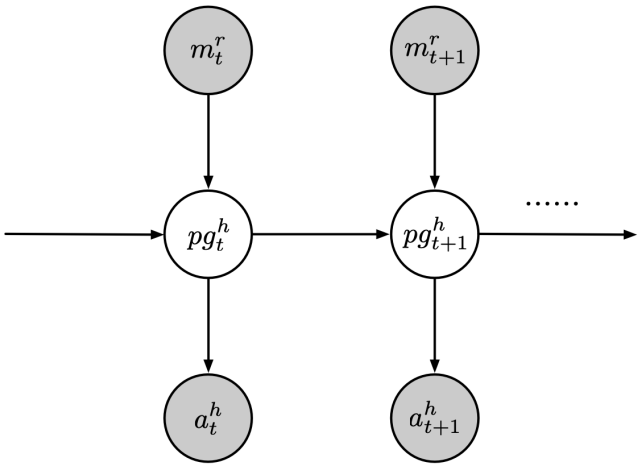


Figure 5.4: Human mental model update process. We use it to infer user mental state pg^h, which is hidden to the robot. Here we assume human actions a_t^h and robot message m_t^r are conditional independent given human mental state pg_t^h at time t.

and human-robot interaction data $D_T = \{d_t\}_{t=1,\ldots,T}$, we infer the user mind $\hat{pg}^h$ iteratively:

$$\hat{pg}^h = \arg\max_{pg^h} p(pg^h|D_T, G), \tag{5.2}$$

$$p(pg^h|D_T, G) \propto p(pg^h|G, D_{T-1})p(d_T|pg^h, G). \tag{5.3}$$

Here the first term models the prior on the user mind given previous data D_{T-1} and AoG structure G. The second term models the likelihood for new data d_T.

To model the likelihood function $p(d_T|pg^h, G)$, we take a sampling-based approach. For each interaction data d, we consider user atomic action a_{obs}^h and communication between the two agents m. The idea is to model how likely the user performs action a_{obs}^h when receiving message from the robot m^r, with current mental state pg^h, as shown in Figure 5.4. Assuming a_{obs}^h and m^r are

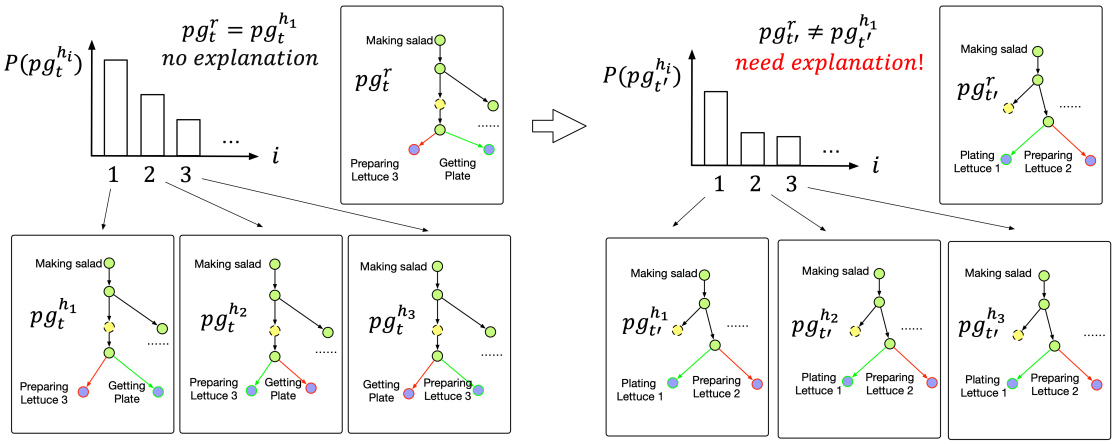


Figure 5.5: Explanation timing. At time t, sort posterior probability of $pg_t^{h_i}$ in descending order, and then compare the most possible user mental state $pg_t^{h_1}$ with robot mental state pg_t^r. Since they are the same, there is no need to explain to the user. At time t', $pg_{t'}^{h_1}$ is not equal to $pg_{t'}^r$, therefore, the robot should provide the explanation.

conditional independent given pg^h we have:

$$p(d|pg^h, G) = p(a^h_{obs}|pg^h, G)p(m^r|pg^h, G), \tag{5.4}$$

$$p(a^h_{obs}|pg^h, G) = \sum_{a^h_{samp}} p(a^h_{samp}|pg^h)p(a^h_{obs}|a^h_{samp}), \tag{5.5}$$

where $p(a^h_{samp}|pg^h)$ denotes the probability of sampled human action a^h_{samp} given current estimation of human mental state pg^h. $p(a^h_{obs}|a^h_{samp})$ measures the similarity between observed human trajectory a^h_{obs} and sampled trajectory a^h_{samp}.

In practice, we use rapid-exploring random tree (RRT*) for trajectory sampling and dynamic time warping (DTW) based approach to compare trajectories. DTW outputs a difference score $diff$. We use it in the energy function for the Boltzmann distribution. Then we update the human mental state in every time-step through the following equation:

$$P(\hat{pg}^h_{t+1}|D_T, G) = \frac{1}{Z}e^{-\frac{diff}{T}}\lambda^n P(\hat{pg}^h_t|D_{T-1}, G), \tag{5.6}$$

Algorithm 1: Planning and explanation generation

```
while Task not finished do
    if Replan needed then
        Collect state information from the game;
        Collect predicted human intentions from the last time step ;
        Call DP planner ;
        Obtain a new sequence of sub-tasks from planner and re-organize AoG based on it;
        Parse AoG through checking pre-conditions and post-effects against the current
          environment state information ;
        Find out the next atomic action to execute based on parsing result ;
    Predict human intentions by equation (5.6) ;
    Measure the difference between predicted intention and expected human actions;
    Generate an explanation if difference is significant;
```

where T is a constant temperature term, Z is a normalization constant, and λ (> 1) is a constant that controls the importance of an explanation. It models how much information the user can retain for an explanation. n is the number of times an explanation about $\hat{pg}^h$ is generated for the user in this task. Therefore, λ^n implicitly encodes the communication history m. Right now, we only consider communications from robot to human m^r. Communication from human to robot m^h can be considered in the future by adding corresponding energy terms. For now, some parameters (T and λ) are set heuristically. These parameters can be learned from annotated user data [CH05].

5.3.3 Robot Mental State Update

Based on the observations in the environment, the robot can update its joint task plan. It is a two-step process. First, the robot collects all relevant information about the task and calls a DP planner described in Section 5.2.3 to obtain an optimal sequence of sub-tasks. Then the robot updates its mental state through re-organizing AoG (Delete finished nodes. Re-order unfinished nodes. If necessary, add back nodes deleted previously). Second, the robot uses causal knowledge (pre-conditions and post-effects of each atomic action) in the AoG terminal nodes to determine the next atomic action. If pre-conditions for the next atomic action are satisfied, the robot will execute it. Otherwise, the robot will be idle, waiting for the user to complete the other part of the job.

5.4 Explanation-based task coaching

In this section, we propose a framework for explanation generation to enable efficient human-robot collaboration.

5.4.1 Explanation framework

As shown in Algorithm 1, the framework includes an iterative process of online planning and explanation generation:

1. At a given time, the robot updates its mental state to represent the expected current goals of both agents and corresponding atomic actions;

2. The mental state of the human agent can be inferred, which would be further compared to the robot's mental state. Based on the result, the robot would decide whether explanations are necessary;

3. On the occasions where users perform an action other than that indicated in the explanation, the robot would update its task plan and mental model to reflect the best joint policy and expected mental models in the new state.

   Take the task *making salad* for example. At the beginning of the game, an optimal plan requires the user to first take the plate. A sub-optimal plan could be the user first taking the lettuce. If the user insists on taking the lettuce first regardless of whether explanations are given, the robot will update the task plan and expect the user to gather the plate afterwards.

5.4.2 Explanation Timing

The explanation serves to provide users with the knowledge necessary to finish the task efficiently. This is achieved by inferring the user's mental model during the interaction and comparing it with the robot's. Whenever a disparity between these two models is detected, we can generate

explanations to encourage correction of the user's mental state.

During collaboration, we use temporal parsing to get robot mental state pg_t^r from its And-Or graph at time t. As in Section 5.3.2, user mental states $\hat{pg}_t^h$ can be inferred based on communication history and action sequences. The system generates explanations when there is a mismatch between the robot mental state and inferred human mental state: $|pg_t^r - \hat{pg}_t^h| > \varepsilon$. In practice, we measure $P(\hat{pg}_t^h|D_T, G)$ for every sub-tasks at each time step based on equation (5.6). If the probability $P(\hat{pg}_t^h = pg_t^r|D_T, G)$ is lower than a threshold, we generate an explanation for the user. This process is shown in Figure 5.5.

5.4.3 Explanation Content

We envision the disparity occurred between the user's mental state and robot's due to several reasons:

1. The user wants to achieve goals that are different from the robot's expectation;
2. The user performs incorrect atomic actions to achieve a sub-goal;
3. The user is unaware of the pre-condition or effect of an atomic action.

In this paper, we do not distinguish between the possible causes of disparity when choosing the explanation timing, as they are too ambiguous. Instead, we propose to generate hierarchical explanation which consists of three components of the robot's mind representation:

1. The robot would explain the current expected sub-goals of both agents (s_t^h, s_t^r) based on its mental state pg^r, e.g., "My current goal is preparing the lettuce. Meanwhile, your expected goal is getting the plate.";
2. The robot communicates the expected atomic actions that both agents are supposed to perform (a_t^h, a_t^r), e.g., "Currently, I'm performing the action slicing the lettuce. You are supposed to perform the action taking the plate.";

3. In addition, by showing images of world states before and after an action (as shown in Figure 5.6), the robot would also demonstrate the fluent change caused by an atomic action $f_t \xrightarrow{a_t} f_{t+1}$.

5.5 User Study

We conducted a user study in a gaming environment to evaluate our algorithm, where participants can collaborate with agents on a virtual cooking task. The gaming environment and explanation interface are displayed in Figure 5.6.

5.5.1 Experiment Domain

Our experiment domain is inspired by the video game **Overcooked**[1], where multiple agents are supposed to make use of various tools and take different roles to prepare, cook, and serve various dishes. Particularly, we use Unreal Engine 4 (UE4) to create a real-time cooking task, namely making *apple juice*. To finish the task, teammates need to take apples from the box and slice them with a knife near the chopping board. Three apple slices should be put into the juicer before producing and delivering apple juice. Figure 5.6 shows a top-down view of the environment. The game interface is designed to be interactive (e.g., object appearance will change after taking valid actions) so that people can easily play through.

To finish the task, each user needs to complete a sequence of 62 atomic actions, if acting optimally, and observe 5 different object fluent changes with a total state space around 10^9. An example task schedule is shown in Figure fig. 5.7.

5.5.2 Experiment Design

Hypotheses. The user study tests the following hypotheses with respect to our algorithm in the collaboration:

[1] `http://www.ghosttowngames.com/overcooked/`

- **H1: Task completion time.** Participants would collaborate with the robot more efficiently if the robot generates explanations based on the human mental state modeling, compared to the other conditions.

- **H2: Perception of the robot.** Participants would have higher perceived helpfulness and efficiency of the robot, as a result of receiving explanations based on the human mental state modeling, compared to the other conditions.

Manipulated Variables. We use a between-subject design for our experiment. In particular, users are randomly assigned to one of three groups and receive different explanations from the robot:

- **Control:** Users would not get any explanations from the robot. As a result, they can learn to finish the task by interacting with the environment.

- **Heuristics:** The robot gives explanations when there is no detected user action for a period of time. This serves as a simple heuristic for the robot to infer whether the user is having difficulties in finishing the task. The timing threshold is set to 9.3 seconds, based on the result of a pre-study in which users can actively ask for explanations when they get stuck.

- **Mind modeling:** The robot gives explanations when there is a disparity between robot and human mental states.

Study Protocol. Before starting the experiment, each participant signs an informed consent form. An introduction is given afterward, including rules and basic controls of the game. As a part of the introduction, participants are given three chances to work on a simple single-agent training task, to verify their understanding. Those who fail to complete the training task in one minute would not continue the study. This is a comprehension test to exclude people who do not understand game control.

Participants who finish training get to see further instructions before starting to collaborate with the robot. They are first educated about the goal of a collaboration task (i.e., making *apple juice*)

and what actions the team should perform to finish it. This is done to make sure every participant has sufficient knowledge to finish the task, so that the impact of user-specific prior knowledge can be minimized. To prepare users to interact and communicate with the robot agent, we would also show them a top-down view of the level map (as shown in Figure 5.6), the appearance of the robot agent as well as an example of an explanation. During the task, the team is required to make and serve two orders of dishes in the virtual kitchen. At the end of the study, each participant is asked to complete a post-experiment survey to provide background information and evaluate the robot teammate.

Measurement. In the background study, we have collected from users their basic demographic information, education, as well as experience with video games.

Our objective measure is intended to evaluate the human-robot teaming performance and subjective measure is designed for evaluating users' perception of the robot. Our dependent measures are listed below:

- **Teaming performance.** We evaluate teaming performance by recording the time for the team to complete each order.
- **Perception of the robot.** We measure user's perception about the robot, in terms of its helpfulness and efficiency. Helpfulness is comprised of questions that measure users' opinion on the robot's ability to provide necessary help. Efficiency is comprised of questions that measure users' opinion on how efficiently and fluently the team is able to finish the task.

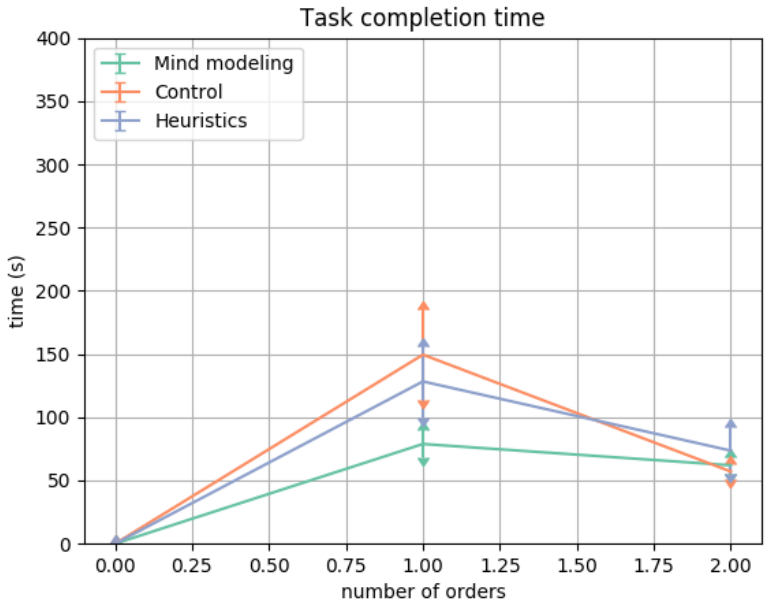


Figure 5.8: Time taken for the team to complete two orders under different testing conditions.

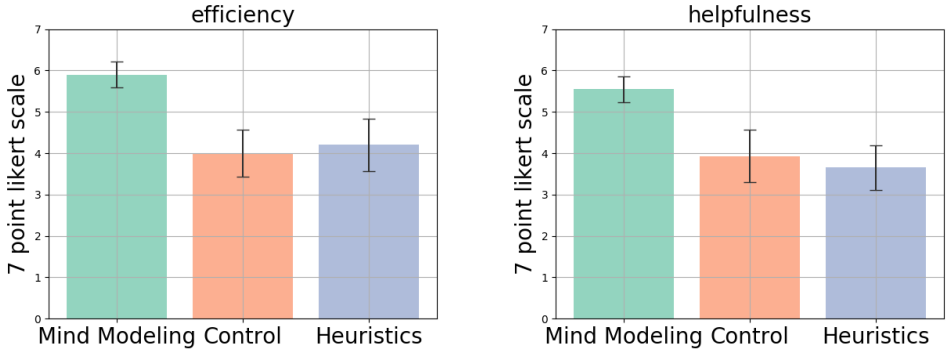


Figure 5.9: User's self-reported perception of the robot in terms of its efficiency and helpfulness.

5.5.3 Results and Analysis

We recruited 29 subjects for our IRB-approved study from the university's subject pool. Most of the participants (69.3%) came from a non-STEM background. Their reported ages ranged from 17 to 36 (M=19.52, SD=2.89). All the participants have moderate experience with video games and have not played the video game **Overcooked**, which inspired our study design. Each participant

got 1 course credit after completing the study. In addition, for ease of conducting the study, we discarded the data of 2 participants from the control group, as they got completely lost and failed to finish the designated task. As a result, there are 10 valid participants in the ”mind modeling” and ”heuristics” group, and 7 in the ”control” group.

Generally, we use ANOVA to test the effects of different experimental conditions on teaming performance and subjective perception of the robot. Tukey HSD tests are conducted on all possible pairs of experimental conditions.

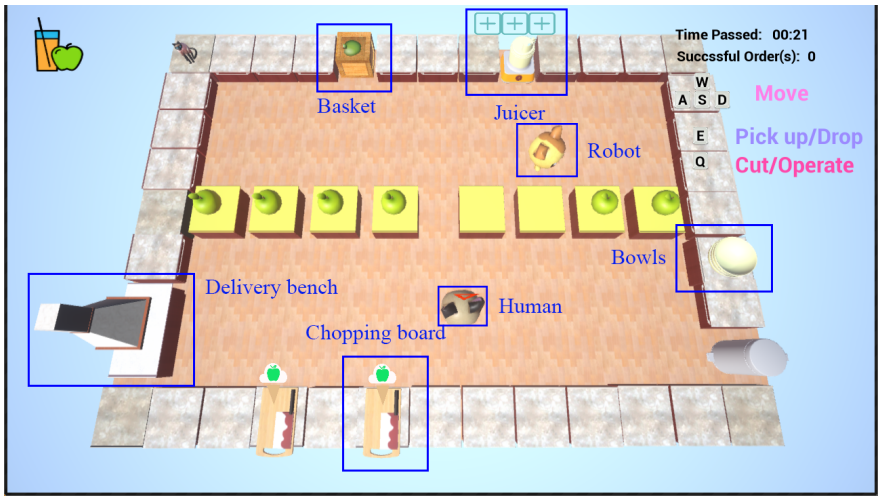
As shown in Figure 5.8, we found marginally significant effects from ”mind modeling” conditions on completion time of the first order ($F(2,24) = 2.038, p = .152$). Post-hoc comparisons using the Tukey HSD tests revealed that teams could finish the first order significantly faster if users were under the ”mind modeling” condition, compared to those under ”control” ($p = .044$). The result is marginally significant compared to those in ”heuristics” ($p = .120$), **confirming H1**. However, for the completion time of the second order, we did not find any significant effect ($F(2,24) = 0.425, p = .658$). This is not surprising since users were asked to finish the same task twice. They could take advantage of their previous experience working with the robot for the second order. Intuitively, the quantitative result showed that our explanation generation algorithm helped non-expert users to finish the task efficiently on their first run, while those in the control group needed to complete the task once to be able to finish it with the same efficiency.

The factorial ANOVA also revealed a significant effect of the explanation system on the perceived helpfulness ($F(2,24) = 4.663, p = .019$) and efficiency ($F(2,24) = 4.136, p = .029$) of the robot (Figure 5.9). **In support of H2**, post-hoc analysis with the Tukey HSD tests showed that the robot's perceived helpfulness was significantly higher under the ”mind modeling” condition, compared to ”control” ($p = .023$) and ”heuristics” ($p < .01$). Users under the ”mind modeling” were also more likely to believe the explanation system resulted in improved collaboration efficiency, compared to ”heuristics” ($p = .026$) and ”control” ($p < .01$).

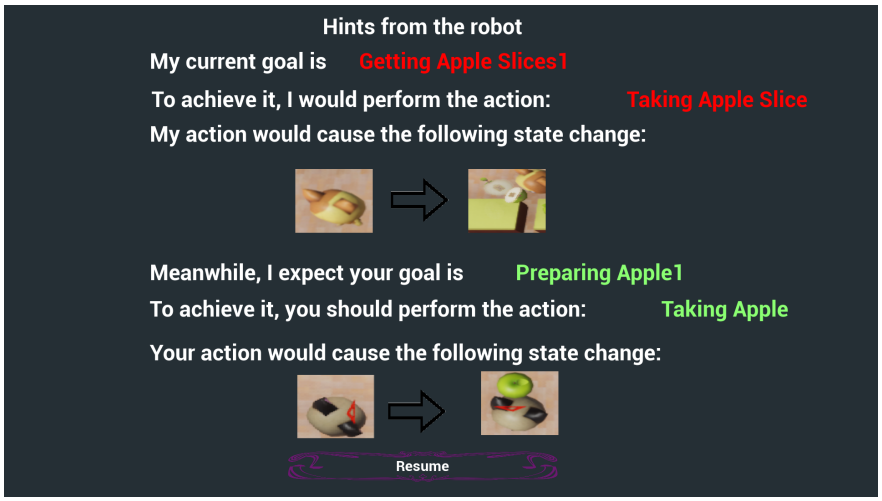
5.6 Conclusion

In this paper, we propose a framework that allows a robot agent to improve teaming performance by communicating compelling explanations to its non-expert human teammate. By maintaining the mental state of both agents, the robot agent successfully generates explanations when the human behavior deviates from the optimal plan. By conducting a user study on a virtual collaborative cooking task, we demonstrate that the proposed algorithm can improve efficiency and quality of the interaction.

For simplicity of implementation, the current environment configuration prevents human and robot from having a shared workspace. For future work, we plan to study more cooking tasks in a diverse set of environments where multiple collaboration strategies can evolve. In addition, to make the robot's model more transparent, we consider to generate contrastive explanations with respect to identified incorrect user beliefs from the user's mental model in the future. Meanwhile, we plan to focus on a more balanced settings where both the human and robot agent have some information (e.g. ability, preference) to share with the teammates before a valid and efficient joint task plan can be formed.



(**A**)



(**B**)

Figure 5.6: (a) A top-down view of our collaborative cooking game, where the user (the bottom character) collaborates with a robot (the top character) on some cooking tasks, e.g. *making apple juice*. (b) The explanation interface exhibits the expected sub-tasks for both agents. Pre-conditions and post-effects of atomic actions are displayed as well.

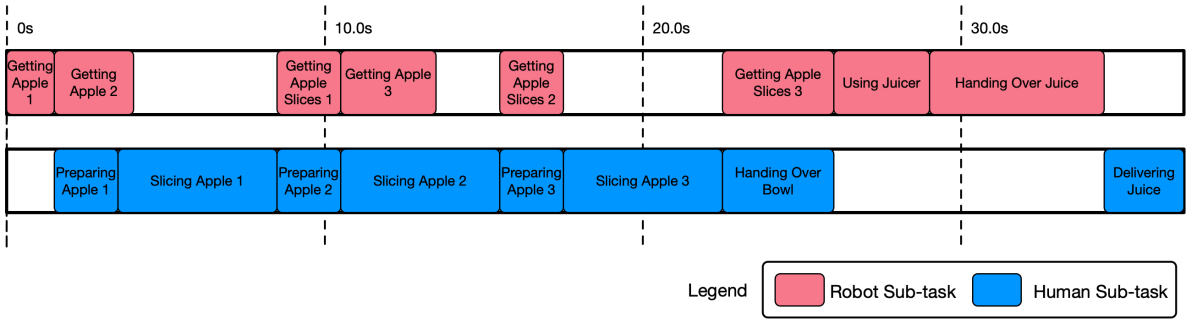


Figure 5.7: An example task schedule for *making apple juice*. The robot maintains the schedule to reflect its expectation on how the team should finish the task. Each color block represents a sub-task, performed by either robot or human. At a specific timing, we can assign tasks to both agents based on the schedule. E.g. at 10.0s, the robot is *getting apple slices 1* while the user is supposed to be *preparing apple 2*. The schedule gets updated based on inferred human mental states, as shown in Algorithm 1.

CHAPTER 6

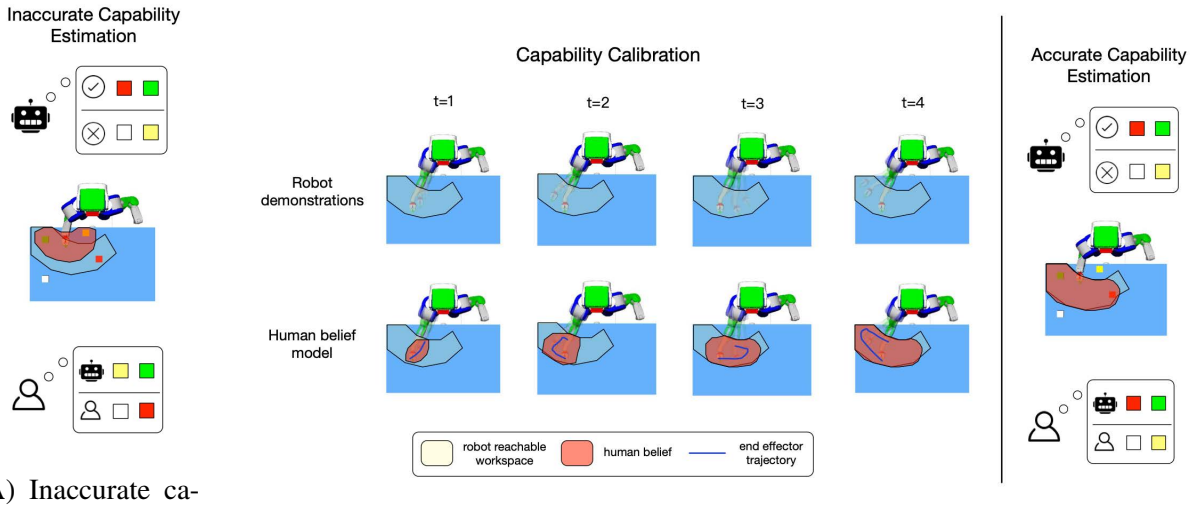
Generating Expressive Motions for Calibration on Robot Reachable Workspace

6.1 Introduction

One of the main challenges in Human-Robot Interaction is that the capacity of the robot perceived by the human partner may not be consistent with its actual capacity [SGS99, PK06, FKS08]. Such discrepancy may lead to overuse or misuse of the robot. Particularly, in an ad-hoc teaming setting where humans do not have prior experience with their robot partners, the consequence caused by such discrepancy could be detrimental to the team collaboration [AS17].

In this work, our key insights to address this challenge are two-fold: i) humans' perception of the capability of a robot can be calibrated by observing its behavior, e.g., robot demonstrating its motion trajectories in pursuit of certain goals, and ii) calibrating the perceived robot capability improves the quality of subsequent human-robot collaboration.

We focus on a case study as shown in Figure 6.1, where a human user and a robot share the same workspace, and they must take turns picking up all objects as fast as possible. As the robot can only reach part of the workspace due to its mechanical limits, the human partner needs to pick up the objects that the robot can not reach to achieve maximum efficiency in completing this joint task. We introduce capability calibration as shown in Figure 6.1b, where we allow the robot to show a small number of demonstrations. After watching each demonstration, the human can estimate the robot's capability accordingly. The goal is to come up with motion plans to pragmatically demonstrate the robot's capability.



(**A**) Inaccurate capability estimation can lead to failure in collaboration.

(**B**) In this example, the human is supposed to pick up the white and the yellow cubes and let the robot collect the red and the green ones.

Figure 6.1: (a) Consider a collaborative table clearing task, where the robot has a limited capability and cannot reach the yellow and white objects. Users who incorrectly estimate that the robot can reach the yellow object would assign it to the robot, resulting in a worse teaming performance. (b) We propose capability calibration, where the robot uses its motion to demonstrate its capability before collaboration.

To achieve a sample-efficient calibration, we propose reachability-expressive motion planning (REMP), a novel planning algorithm that models perceived robot capability as a human's belief over a robot's reachable workspace, and integrates the belief update into motion planning by introducing an additional cost in trajectory optimization. As a result, REMP can generate a series of expressive trajectories for different robots to showcase their reachability to users. We conducted a user study in which participants i) first observed several robot demonstrations, then ii) estimated where the robot could reach, and iii) proceeded to work with the same robot in a joint task: picking up all objects in the shared workspace as fast as possible. We find that i) REMP significantly increases the accuracy of humans' reachability estimation, ii) the subsequent human-robot collaboration benefits from a successful calibration, iii) users perceive the robot as more predictable and reliable.

6.2 Related Work

Perceived robot capability. Various works have studied how humans perceive a robot's capability, differentiating between social and physical capabilities [CDS15]. In prior work, social capabilities were defined as a robot's ability to communicate and interact with humans [JLD13], and physical capabilities were defined on a set of tasks a robot can successfully perform [RVB21], such as lifting different objects on the table [NNP17, CNS18], searching and firefighting in various weather and fire conditions [XBO19]. In these works, robots' capabilities of different tasks are estimated separately based on experience counts of action outcomes – a higher success rate indicates a stronger capability in a task [LFK20]. Since the capability modeled in these works are highly task dependent, the user's knowledge of a robot's capability in one task can not be easily generalized to the knowledge of its capability in other tasks. In contrast, our work focuses on physical capabilities that serve as a basis for a robot to achieve success in a wide range of tasks. In particular, we focus on *reachability*, which is one of the most fundamental physical capabilities for robots. By understanding a robot's reachability, users can better assess its overall capability in various tasks

where reaching is involved. Given an arbitrary task, the user can decide whether the robot can successfully perform it based on perceived reachability.

Robot expressive motions. When deploying robots in real-world settings that are beyond factory environments, functional motions only designed to accomplish tasks are inadequate for human users to correctly understand the robots and establish effective collaborations [VK19]. It is equally important to convey the rationality and intent of a robot through its motion [SMF14, LBA21]. To generate such motions, prior work formulated and optimized the legibility of trajectories via functional gradient descent [DS13, SGB15]. Similar ideas were also adopted to study robots' expression of emotion [FMB15] and style [LHP05]. To express robot (in)capability, prior work used repetitive motions, either generated by simple heuristics [NM01] or hand-crafted for each task [TDJ11]. In contrast, [KHD18] proposed a trajectory optimization-based method that maximizes the similarity between motions and would-be successful executions. Our work takes one further step in this direction: we i) model how humans update their beliefs of the capability of a robot given the observed robot motions and ii) integrate the belief update process into trajectory optimization to generate new motions that can optimally improve humans' beliefs.

6.3 Capability Calibration

We propose a capability calibration framework (as shown in Figure 6.1b) to align a human user's understanding of a robot's capability with the ground truth, where the user can watch a small number of demonstrations of her robot partner before they work together. In this section, we introduce our approach to generate such demonstrations that can efficiently reveal the robot's reachability. We show how this calibration could be applied to collaboration in Section 6.4.

6.3.1 Calibrating Reachable Workspace

In this sub-section, we define the reachability calibration task. In Section 6.3.2, we describe how human belief would be updated before a new trajectory can be generated. In Sec. 6.3.3, we propose

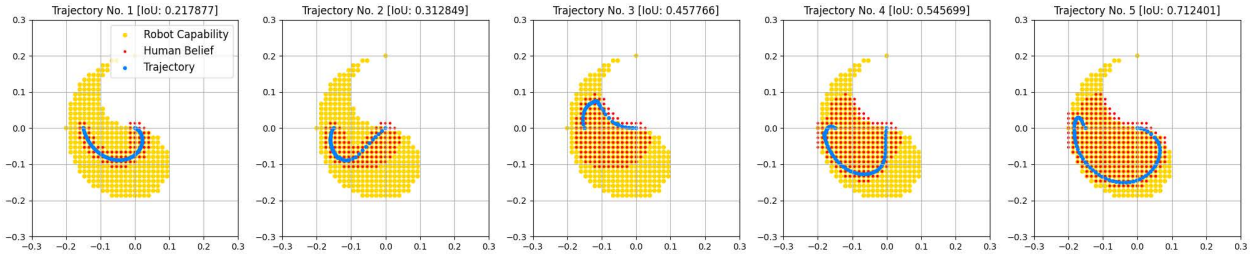


Figure 6.2: Simulated human estimation of robot A's reachability map, after observing each demonstration generated by Algorithm 3, measured by Intersection of Union (IoU) between the human estimation and the ground truth. Robot A is a 2-link arm with link lengths 0.1.

REMP, which enables the robot to generate one trajectory showing its reachable workspace based on a simulated human belief that models what the human has already known about the robot. In Section 6.3.5, we reify our capability calibration framework by combining REMP and task planning. We begin with some notations.

The robot's ground truth reachability is defined as $f : \mathscr{X}_{ws} \to \{0,1\}$, i.e. whether a target position x in the workspace $\mathscr{X}_{ws}$ is reachable by the end-effector according to the robot's kinematic constraints. Meanwhile, we assume the human is maintaining a belief $b_h^t : \mathscr{X}_{ws} \to [0,1]$, modeling how likely a target is reachable after observing a robot trajectory $\xi_{1:N}^t \in \Xi$ with length N at time $t \in \{1...T\}$. After observing all the robot demonstrations, human's final belief would become $b_h^T = \tau(b_h^0, Z)$, with τ denoting the belief transition from the initial guess b_h^0 to the final b_h^T with all seen demonstrations $Z = \xi^{1:T}$. In addition, we define $\mathscr{X}_{rs} \subseteq \mathscr{X}_{ws}$ as the robot's reachable workspace. $\phi_{ee} : \mathscr{Q} \to \mathscr{X}_{rs}$ is the forward kinematic function of the end-effector, generating its position given a configuration.

Using notations above, we can formalize capability calibration as an optimizing problem over a set of trajectories, Z, whose cardinality may not necessarily be known in advance. The goal is to reduce the mismatch between the robot's ground truth capability and the user's final belief:

$$\begin{aligned} &\arg\min_{Z \in \Xi^*} Cost(Z) \\ s.t. &\sum_{x \in \mathscr{X}_{ws}} \left| \tau\left(b_h^0(x), Z\right) - f(x) \right| < \varepsilon, \end{aligned} \tag{6.1}$$

where Ξ^* uses the Kleene star to represent all possible sequences of robot motions, $Cost(\cdot) : \Xi^* \rightarrow \mathbb{R}$ is a function to evaluate the overall cost for trajectories. The condition means the user has a reachability estimation close enough to the robot's true capability.

One intuitive cost is the total length of all the trajectories in Z, optimizing which is equivalent to minimizing the cardinality of Z when the trajectories all have similar length. In this paper, we maintain the homogeneity of trajectories by further regulating the start configurations of all trajectories to be in a set of configurations $S \subset \mathscr{Q}_0$ and target positions in a set of positions $G \subset \mathscr{X}_{rs}$.

Due to the size of motion space, an exact solution to eq. (6.1) is intractable. Thus, we adopt an incremental update: we keep generating new trajectories ξ^t until the user's belief is sufficiently aligned with the robot's capability. Every time we want to expand Z, we first select a pair of starting configuration and target position $(q^t, x_r) \in S \times G$ and then generate a motion using it. We term the former as the task planning problem and the latter as motion planning.

6.3.2 Human Belief Model

Our objective is to make people without any knowledge about robotics easily understand the true capacities of a robot. Thus, our human belief model attempts to capture what a novice user may think about a robot's reachability after watching its trajectories. We assume human updates its belief on an interested point x in the workspace after observing a new robot trajectory ξ. Intuitively, if a point is close to the visited positions in an observed trajectory, the human observer would consider it more likely to be reachable. We model the belief update process as an iterative Bayesian inference beginning from a uniform prior:

$$\tau(b^t, \xi^t)(x) = b_h^{t+1}(x) \propto b_h^t(x) p(\xi^t | x) \tag{6.2}$$

and $p(\xi^t | x) = e^{-\gamma d(\phi(\xi^t), x)}$, where $d(\phi(\xi), x)$ captures the distance between the trajectory ξ and the interested position x. The hyperparameter γ defines how much the human extrapolates the observed trajectory to the points nearby: a large γ means that such extrapolation mainly happens

Algorithm 2: REMP

1 Given a target position x_r and a starting configuration q^t, human belief b_t;
2 Generate trajectory ξ^t based on b_h^t, q^t, Equation (6.4) ;
3 Update human belief b_h^{t+1} using ξ^t, Equation (6.2) ;
4 **return** b_h^{t+1}, ξ^t

to the point which is very close to the trajectory. In particular, we use the end-effector position ϕ_{ee} as the feature, and compute the squared euclidean distance between the interested position and the closest end-effector position in the trajectory:

$$d(\phi(\xi), x) = \min_i ||\phi_{ee}(\xi_i) - x||^2. \tag{6.3}$$

The design of our distance function is motivated by the fact that, given a trajectory, it is straightforward for users to focus on the robot's end-effector which is central to the task, while trying to estimate its reachable workspace.

6.3.3 REMP: Reachability-Expressive Motion Planning

Expressing robot reachability is more than randomly moving the end-effector to somewhere in its reachable workspace. Our insight is that it is essential to understand what the human already knows or does not know about the robot, so that every demonstration can communicate as much information to the human as possible. We believe this can be formulated as an optimization problem: finding a new trajectory that would minimize the misalignment between the ground truth reachability and human's updated estimation. We capture the misalignment using a cost function $c(\xi, b_h^t, f)$ and formulate the optimization problem as the following:

$$\begin{aligned} \xi^t = \arg\min_{\xi} \quad & c(\xi, b_h^t, f) + \frac{1}{\lambda} \sum_{i=1}^{N} ||\xi_{i+1} - \xi_i||^2, \\ \text{subject to} \quad & \phi_{ee}(\xi_n) = x_r, \text{collision-free}(\xi). \end{aligned} \tag{6.4}$$

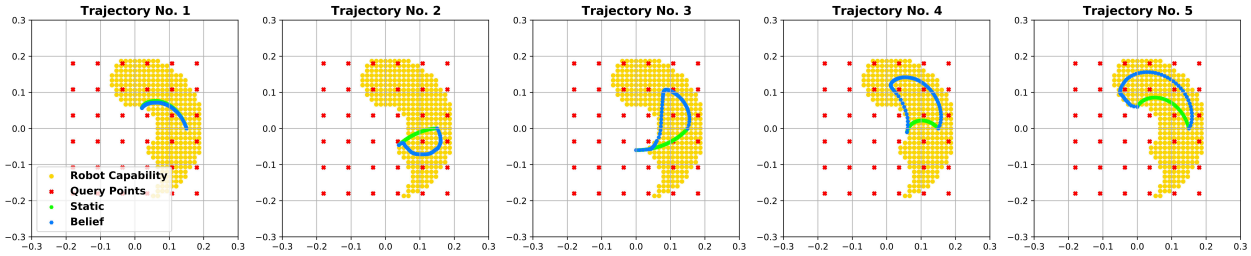
The first term is an expressiveness cost and the second term is a smoothness cost commonly seen in trajectory optimization. The trajectory at the t-th step is generated by minimizing the sum of the two costs, subjecting to a constraint that requires the end effector to reach a target position x_r at the end of the trajectory.

Assuming each point in the trajectory contributes to the cost independently, we design the cost function based on a value $v_i(\xi_i, b_h^t, f)$, which represents the degree of alignment between human's estimation and the robot's ground truth reachable workspace:

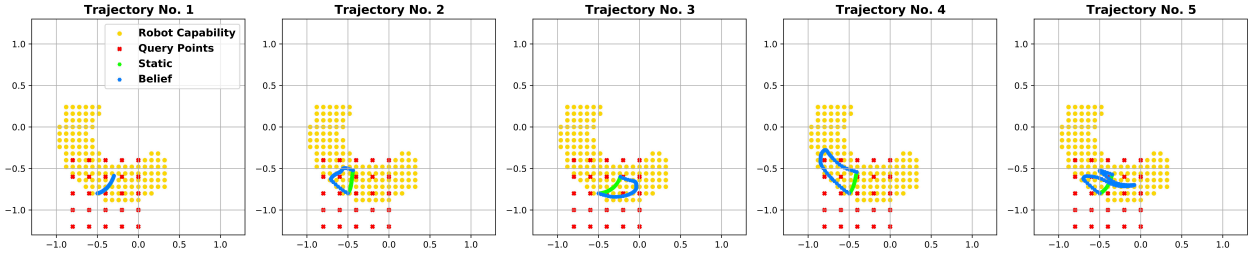
$$\begin{aligned} c_b(\xi, b_h^t, f) &= \alpha \sum_{i=1}^{N} v_i(\xi_i, b_h^t, f) \\ &= \alpha \sum_{i=1}^{N} e^{\beta \left(b_h^t(\phi_{ee}(\xi_i)) - f(\phi_{ee}(\xi_i)) \right)} \end{aligned} \tag{6.5}$$

A small value v_i suggests that the human observer is underestimating the robot's capability at ξ_i. In that case, we want to facilitate calibration by encouraging the robot to move to ξ_i. On the other hand, we would see a large v_i if the human is over-estimating the capability. In that case, it is beneficial for the robot to avoid reaching points near ξ_i. The hyperparameter α and β control how aggressive the trajectory would be in expressing the capability. We call this cost function c_b, which captures human updated belief. Note that the intuition is if the observer previously underestimates the reachability of a point x, $b_h^t(x) - f(x)$ will be negative and give low cost for trajectories covering x. Hence, trajectories passing through underestimated points are more likely to be chosen. Trajectories including overestimated points, on the contrary, have larger costs and are less likely to be selected.

Static human model. Our key intuition is the human would update its belief of the robot's reachability after observing each trajectory. To test it, we also design a fixed cost function as baseline, assuming an underlying uniform belief model $\forall x,\ b_{static}(x) = b_0$. The corresponding cost function under the assumption of a static human model is c_s. Note that this baseline generates functional motions that solely aim to finish the physical task of reaching the target. We envision that in reality, users may also learn from these physical motions the robot's capability by interacting with the



(**A**) Robot B is a 2-link arm with link lengths 0.13 and 0.07.



(**B**) Robot C is a PR2 robot. In this work, we consider the reachable workspace of its right arm.

Figure 6.3: Visualization of the robot reachable workspace and the trajectories generated by cost function c_b (*belief*) and c_s (*static*). (a) and (b) show the results for Robot B and Robot C respectively. It can be seen that the *belief* trajectories cover broader regions of the reachable workspace and new trajectories tend to visit areas that haven't been covered by their predecessors. The red dots, corresponding to Figure 6.6, represent the points we use to query the users in our experiments.

robot on some tasks, but such learning is not as efficient as the learning in a dedicated calibration phase.

6.3.4 Generating Reachability-Expressive Trajectories

Implementation. We implemented our framework using TrajOpt [SHL13] on two kinds of simulated robots in OpenRAVE [Dia10], including a manipulator with 2 links and a PR2 robot. For the 2-link arm, we manipulated its joint limits and link lengths to allow it to have a variety of two-dimensional reachable workspaces. These serve as testing cases for our framework, as we want to study how well the framework copes with reachable workspaces of different sizes and shapes. For the PR2 robot, we didn't do such manipulations since the goal here is to see how practical it is to apply the framework to real robot manipulators. Without loss of generality, we focus on the right arm of the PR2 robot. In practice, we use grid search to find hyperparameters that gener-

Algorithm 3: REMP-T

Given a list of target position and starting configuration pairs $(x,q)^{1:T}$ and human belief b_h;
for $t = 1,\ldots,T$ **do**
    b_h, ξ^t = REMP(x^t, q^t, b_h)
return $b_h, \xi^{1:T}$

Algorithm 4: Calibration on Reachable Workspace

Given a set of target positions G, starting configurations S, initial human belief b_0;
for $t = 1,2,\ldots,$ **do**
    $\forall x,\ b_h^0(x) \leftarrow b_0$;
    Let $\sigma(\zeta) = \sum_{x\in\mathscr{X}_{ws}} \left| f(x) - \text{REMP-}T(\zeta, b_h^0)[b_h] \right|$;
    $\zeta_t^* \leftarrow \arg\min_{\zeta\in(G\times S)^t} \sigma(\zeta)$;
    $\sigma_t \leftarrow \sigma(\zeta_t^*)$;
    if $\sigma_t - \sigma_{t-1} < \varepsilon$ **then**
        return REMP-$T(\zeta_t^*, b_h^0)$

ate trajectories to maximize the accuracy of reachability estimation in simulation, as described in Section 6.4.3.

Qualitative behaviors. Figure 6.2, Figure 6.3**A** and Figure 6.3**B** show the trajectories generated by the cost functions c_b and c_s for the robots by running REMP iteratively using the updated belief, following Algorithm 3. As both c_b and c_s assume a uniform belief on the robot's reachable workspace at the beginning, the first trajectories generated by these cost functions are almost identical. Starting from the second trajectories, we find that the ones generated using c_b can cover a large part of the robot's reachable workspace. On the contrary, trajectories generated by c_s are more sensitive to the physical cost. Overall, It is clear that REMP accommodates human belief at each time step and tries to traverse uncovered regions to better express the robot's reachability.

Algorithm 5: Calibration with Fixed T

Given a set of target positions $G = \{x_1, \ldots, x_N\}$, number of trajectories T, starting configuration set S, initial human belief b_0;
$\forall x,\ b_h^0(x) \leftarrow b_0,\ \delta \leftarrow \infty$;
for $\kappa \in T - combination(G \times S)$ **do**
    $b_h \leftarrow b_h^0$;
    for $t \leftarrow 1$ **to** T **do**
        $b_h, \hat{\xi}^t \leftarrow \text{REMP}(\kappa^t(G), \kappa^t(S), b_h)$;
    $\sigma = \sum_{x \in \mathscr{X}_{ws}} |b_h(x) - f(x)|$;
    if $\sigma < \delta$ **then**
        $\delta \leftarrow \sigma$;
        $\xi^{1:T} \leftarrow \hat{\xi}^{1:T}$;
return $\xi^{1:T}$

6.3.5 Planning for Start and Target Pairs

We have shown how REMP can generate an expressive trajectory *given* a starting configuration and a target position. For a better capability calibration, we also want to optimize the number of trajectories as well as the sequence of starting configurations and target positions. As outlined in Alg. 4, this could be achieved by Task and Motion Planning (TAMP) [KL11], where the plans of start and target pairs come from task planning and the trajectories for a given pair comes from REMP. In Alg. 4, we solve (6.1) in an incremental manner. Namely, we keep increasing the cardinality of Z until user's belief is aligned with the robot's actual capability. For each size of Z, we find the best sequence of starting configurations from S and target positions from G (Line 5 of Alg. 4). To avoid trajectories that are too short or uninformative, we set S as the set of configurations near the neutral configuration of the robot and G as the set of positions far away from the neutral end effector positions:

$$S = \{q, \forall q\ |q - q_{neutral}| < a_1\} \tag{6.6}$$

$$G = \{x, \forall x \min_{q \in S} |x - \phi_{ee}(q)| > a_2\} \tag{6.7}$$

Finding Z incrementally, despite giving the exact optimum, can be time consuming. In practice, rather than demonstrating to humans constantly until converge, we can pre-define T to a reasonable number and find the optimal set of trajectories. In Alg. 5, we assume fixed number of trajectories. We start from a uniform prior for the human belief, and update the belief w.r.t. Eq. (6.2). Section 6.3.5 depicts an example to optimize 4 trajectories that start from different configurations and reach different targets. Note that Alg. 5 plans by enumerating all possible combinations, but any stochastic planning approaches can be used to further accommodate resource constraints and task scalability.

6.4 Applying REMP to Human-Robot Collaboration

In this section, we discuss how to apply REMP to human-robot collaboration *after* the calibration.

6.4.1 Collaborative Table Clearing

We design a human-robot collaboration task in a table clearing scenario, where some objects are scattered on a table and a robot can assist the human with the object collection. The human and the robot take turns picking up the objects. In each step, the human collects first and the robot collects one of the remaining objects. The human can reach all of the objects, while the robot can only reach a subset of them. To finish the task as quickly as possible, the human and the robot need to split the work wisely, so that, in each round, the robot has some objects to pick up. The reward is calculated by the number of objects picked up and the time penalty.

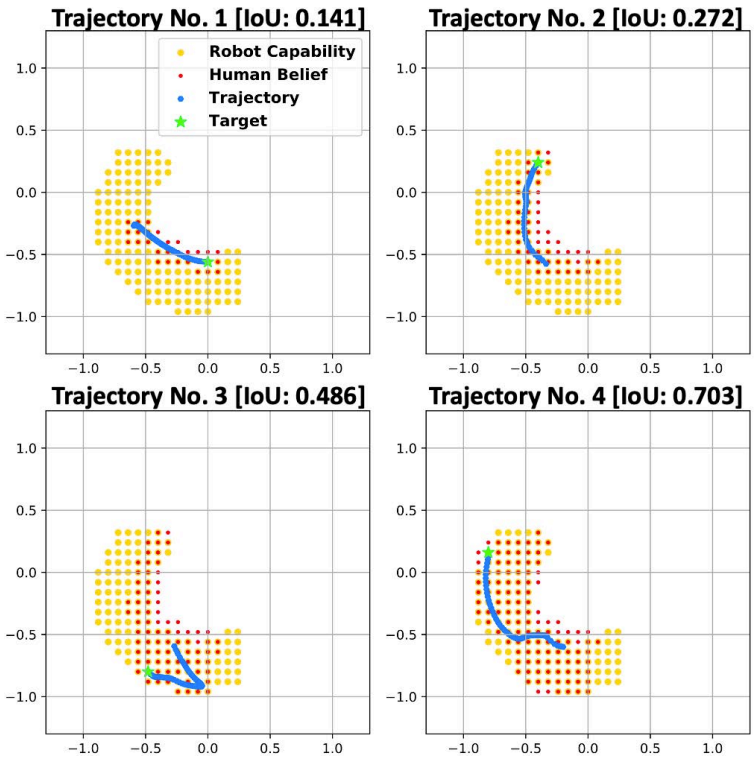
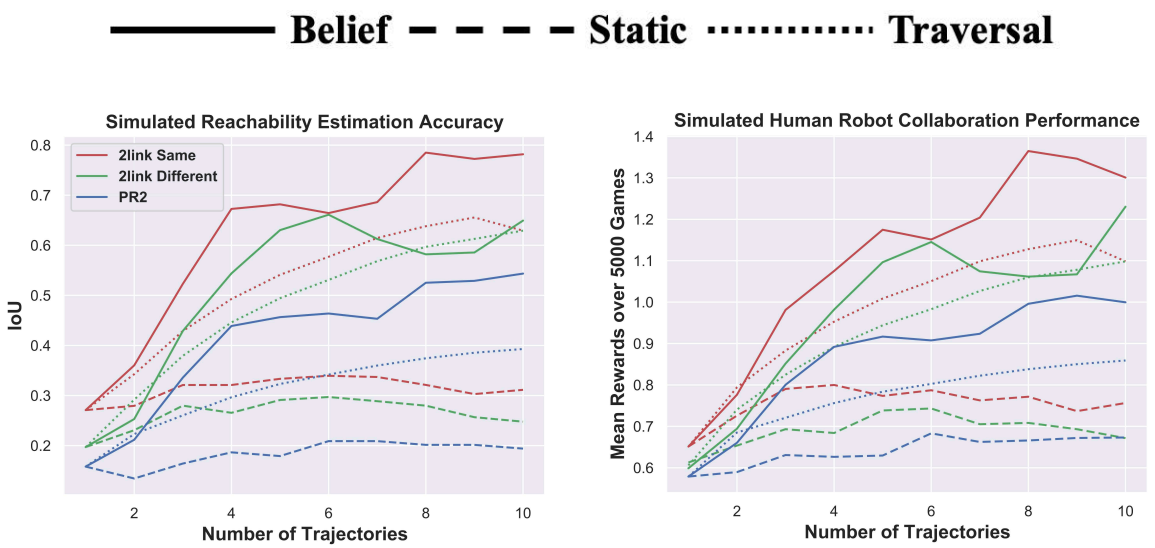


Figure 6.4: Trajectories generated by task and motion planning and the simulated reachability estimation given observations. Combining REMP with task planning, we can optimize the starting and target positions for better calibration.

6.4.2 Human and Robot Policy

After observing robot expressive demonstrations and updating the belief with Eq. (6.2), the human is assumed to act in an approximately rational way with respect to the current estimation of the robot capability, b_h^t. We use a Boltzmann noisily-rational human decision model [MV53], assuming the human is more likely to help the robot with its unreachable objects based on the human's current reachability estimation. Since we want to emphasize the effect of the calibration, we use a simple uniform robot policy in the simulation, i.e., it would randomly pick up objects it can reach, and do nothing if no objects are reachable.



(**A**) Simulated reachability estimation accuracy, measured by Intersection of Union between the human reachability estimation and the ground truth. Higher value indicates better estimation.

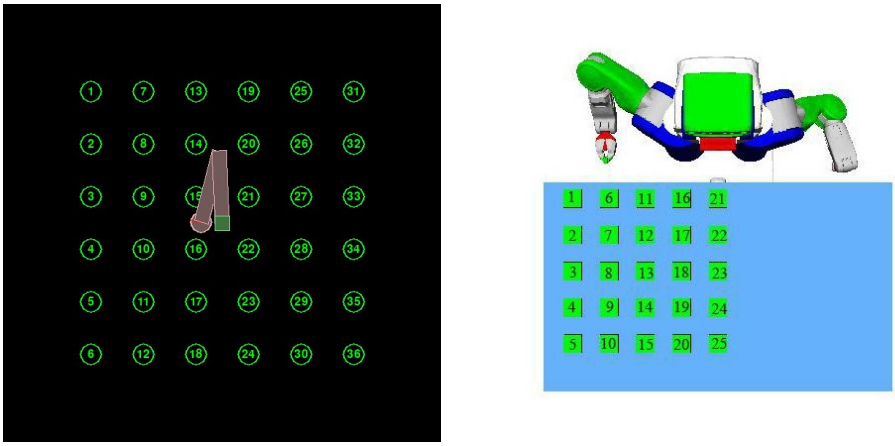
(**B**) Simulated human-robot collaboration performance measured by averaged rewards acquired by the group. Every point on the curves for *traversal* is the mean of 100 trajectories.

Figure 6.5: Simulation results of reachability estimation and collaboration performance. Starting and target positions are chosen greedily.

6.4.3 Simulation Results

Using the behavior model described in the previous sections, we simulated with 3 robots A, B and C with different configurations and reachability: (i) A is a 2-link arm where each link is of equal length, (ii) B is a 2-link arm where the length of its first link (0.13) is larger than the length of the second (0.07), (iii) C is a PR2 robot. The *belief* and *static* methods in the legend correspond to the definition in Section 6.3.2. In addition, we implemented a *traversal* baseline, where the robot moves its end-effector to traverse the workspace to demonstrate its reachability. From the starting position, the end effector moves to unreached waypoints in its reachable workspace one by one. The number of waypoints we sampled corresponds to the number of trajectories in *belief* and *static*.

Figure 6.5**A** and Figure 6.5**B** shows the quantitative results of capability calibration and human-



(**A**) 36 points for 2-link arms. (**B**) 25 points for PR2.

Figure 6.6: To evaluate users' estimation of the robot's reachable workspace, we sample query points in the workspace and ask users to select points that they think the robot's end effector can reach. These points correspond to the red dots in Figure 6.3**A** and Figure 6.3**B**.

robot collaboration. The result suggests that as the robot shows more demonstrations, the human has a better understanding of its capability and collaborates with it more effectively for all baselines. Looking at the sample efficiency, we notice that without modeling human belief changes, the improvement is quite slow and limited: a large number of demonstrations need to be observed before calibration is achieved. On the contrary, trajectories generated by our proposed REMP algorithm keep providing new information to the user. As a result, the user's estimation accuracy increases much faster for *belief* compared to the baselines. There is fluctuation when many trajectories are shown, due to the limited memory of our human model.

6.5 User Study

As we have shown the effectiveness of REMP in simulation, we now turn to investigate how much it helps users work with robots in a user study. This study was certified as exempt from IRB review per 45 CFR 46.104 category 3 by the UCLA Institutional Review Board on 9/4/2020.

Table 6.1: Survey statements to evaluate reachability, predictability, reliability and trust toward robots.

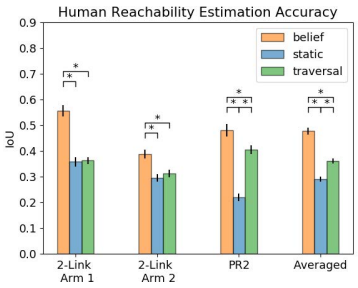
1. It is easy to tell where the robot's hand can reach.
2. The robot behaves in a predictable manner.
3. I can rely on the robot to function properly despite its limited capability.
4. I trust the robot.

6.5.1 Experiment Design

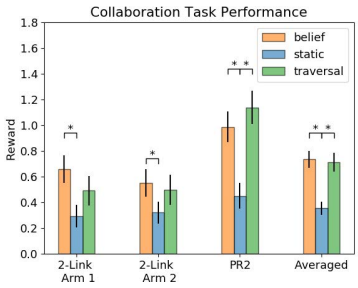
Participants. We recruited 202 participants (37% Female, median age 34) from Amazon Mechanical Turk.

Materials. During the study, participants interact with the three robots in the simulation as described in Section 6.4.3. We measure how well participants can understand the robot's capability and how such understanding can help them in the collaboration, as well as their self-reported perception of the robot. To measure **capability understanding**, we ask users to choose positions that they think the robot can reach from a number of object queries, as shown in Figure 6.6. We record their selections and compare them with the ground truth. For **collaboration task performance**, we use the accumulated reward of the team as a measure. To measure the **perception of the robot**, we ask participants to rate statements listed in Table 6.1 on a 7-point Likert scale labeled from "strongly agree" to "strongly disagree", after they have finished interaction with a robot. Inspired by [MG00], the statements shown in Table 6.1 are designed to evaluate their subjective understanding of the robot in different aspects, including reachability, predictability, reliance and trust.

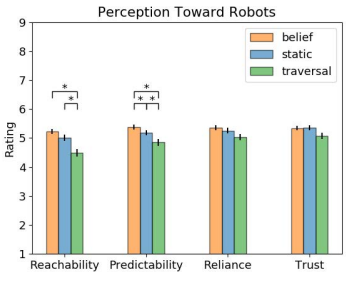
Calibration task. In the calibration task, the participant would be randomly assigned to an experiment group, and observe T demonstrations. Based on the simulations results in Figure 6.5**A**, we believe showing more demonstrations would generally lead to a better calibration. Considering the limited time of participants in our online study, however, we cannot use an arbitrarily large T. From simulation, we witness the most significant improvement during the first 4 trajectories, thus we control the number of demonstrations $T = 4$ in practice. After seeing all demonstrations, participants are asked to estimate the robot's reachable workspace by choosing positions that they



(**A**) Intersection of Union between the human reachability estimation and the ground truth. A higher value indicates better estimation.



(**B**) The human-robot team performance in the collaboration task. A higher value indicates a higher reward.



(**C**) Users' ratings toward the Likert statements in Tab. 6.1. A higher rating indicates higher confidence.

Figure 6.7: User study results. Here we report means and standard errors. * indicates statistical significant pairs ($p < .05$).

think the robot can reach from a number of object queries, as shown in Figure 6.6.

Collaboration task. In the collaboration task, participants are asked to perform an online table clearing task together with the same robot they have just been calibrated. As discussed in Section 6.4.1, the task required the team to clear all four objects on the table. During each time step, the participant would pick up an object first before the robot makes its decision. The team would get rewarded based on how fast they take all the objects. Since two of the objects cannot be reached by the robot, to get the maximum accumulated reward (+2), the participant needs to pick up objects that cannot be reached by the robot. Failure to do so would result in the team getting a lower reward (0).

Experiment conditions. Like simulations, we varied types of motion users observed in the user study, i.e., the *belief*, *static* and *traversal* methods defined in Section 6.4.3.

Design. The robot types are within-subject: participants interacted with all three robots. Demonstrations are between-subject: participants only saw demonstrations from one of the three experiment conditions when interacting with a robot.

Procedure. After a brief introduction, each participant is asked to interact with three robots A, B and C in random order. The purpose is to see how robot's physical configurations affect capa-

bility perception. During the interaction with each robot, the participants would first go through a calibration task before collaborating with the same robot on the table clearing task.

Hypotheses. We hypothesized the capability calibration framework benefits the users in the following aspects:

H1: Participants going through capability calibration in the *belief* condition would have a better understanding of the robot's capability, compared to those in the other conditions.

H2: Teams in the *belief* condition would perform better in the collaboration tasks than those in the other conditions.

H3: Participants in the *belief* condition would have a more positive perception of the robot, compared to those in the other conditions.

6.5.2 Result and Analysis

Capability understanding. We first analyzed the accuracy of the user's estimation of the robot's reachable workspace, by computing the intersection of union (IoU) between their responses and the ground truth. We performed a Kruskal–Wallis H-test of the IoUs using the type of motion independent variables. As a result, we found a significant effect for the motion ($\chi^2(2,603) = 113.52, p < .001$). A post-hoc analysis with Mann-Whitney U test revealed that all three conditions are different from each other, with *belief* significantly better than *static* ($p < .001$) and *traversal* ($p < .001$). This confirms our hypothesis **H1**. Figure 6.7**A** shows the accuracy of participants' reachability estimation w.r.t different robots. On average, *belief* performs 65% better than *static* and 32% better than *traversal*. Compared to the simulation results in Figure 6.5, the user study result follows relatively the same order for different conditions.

Task performance. We also analyzed the collaboration task performance. A Kruskal–Wallis H-test indicates that there is a statistically significant effect of the accumulated rewards between conditions ($\chi^2(2,603) = 22.62, p < .001$). The post-hoc Mann-Whitney U test showed a significant difference between *belief* and *static* ($p < .001$). This partially supports our hypothesis **H2**.

We didn't observe a significant difference between *belief* and *traversal*. Figure 6.7**B** shows the task performance for different robots in three conditions. The Pearson correlation coefficient between reachability estimation and collaboration performance is $r = .203$ with p-value smaller than 0.001, indicating a positive correlation. This validates that calibrating perceived robot capability benefits the collaboration performance. Surprisingly, users in the *traversal* condition have a slightly higher reward when collaborating with the PR2 robot compared with those in *static*, even if their reachability estimation is less accurate, although the difference is not significant. This is probably due to the stochastic nature of the *traversal* baseline and specific object locations in our collaboration task.

Perception of robots. Finally, we analyzed participants' perception toward robots. Running a Kruskal–Wallis H-test, we found significant effects for reachability ($\chi^2(2,603) = 20.39, p < .001$) and predictability ($\chi^2(2,603) = 11.30, p = .003$). The post-hoc Mann-Whitney U test revealed significant difference between *belief* and *traversal* for reachability ($p < .001$) and predictability ($p < .001$), confirming **H3**. Overall, users tended to prefer *belief* over *static*, and *static* over *traversal*. This is unexpected for reachability, considering the fact that users are actually better at predicting robots' reachability in *traversal* than in *static*. The Pearson correlation coefficient between reachability rating and prediction accuracy is $r = .109$, indicating a weak positive correlation. Similarly, we observe a weak correlation $r = .019$ between self-reported reliance and users' actual ability to rely on the robot during collaboration. This suggests that there may be a discrepancy between the users' self-reported capability understanding and what they actually know about the robot.

In summary, we found that users in the *belief* condition had the most accurate estimation of the robots' capability, and reported the robots in this condition as the most reliable, the most predictable, and the easiest to understand among all three conditions. Moreover, users working with the *belief* robots achieved a higher reward than those working with the *static* robots did. These objective and subjective results together suggest that our approach has an overall advantage for improving humans' understanding of robots as well as the quality of collaboration over the

baselines.

6.6 Conclusion

We have proposed an expressive robot motion planning algorithm, REMP, which can generate informative trajectories by integrate human belief update into trajectory optimization. Our experiments show that our approach can efficiently calibrate a user's perception of a robot's reachability and consequently improve human-robot collaboration.

In this work, we focused on the robot's spatial reachability. As reaching is one of the most basic tasks in human-robot interaction, we believe understanding reachability would greatly help users understand robot capacities in more complex tasks. Thus we view our work as a successful first step towards a more general capability calibration setting. In the future, it is possible to extend REMP for other capabilities. Our current work treats predictability and reliability as separate measures from trust. Considering the multidimensional nature of trust, better instruments can be used for a more comprehensive trust measure [MU21]. Also, due to online experiment constraints, we only investigated the reachability calibration problem on a 2D plane. We intend to generalize our algorithm to 3D environment in future work.

CHAPTER 7

In-situ bidirectional human-robot value alignment with communicative learning

7.1 Introduction

What makes a good human-robot team? At the dawn of artificial intelligence (AI), Norbert Wiener [Wie60] identified the foundation of collaborative robots with the warning "if we use, to achieve our purposes, a mechanical agency with whose operation we cannot interfere effectively ... we had better be quite sure that the purpose put into the machine is the purpose which we really desire." Since then, several efforts [KM94, GN07] have demonstrated that effective human-robot collaboration depends on a shared team mental model which includes values [Sch92], goals [RCS92], and current states of the task [RCS92]. To achieve a shared team mental model, humans use communication as an efficient tool to establish a common team understanding of task expectations, with team members adopting anticipatory information-sharing strategies to accomplish collaborative tasks [MES04, BSS16]. In most cases, the sharing process is **bidirectional** among collaborators, as each teammate needs to fulfill the roles of both speaker and listener (*i.e.*, providing private task-relevant information to partners while also accurately comprehending teammates' messages). Successful communication in human-robot collaboration can be signaled by bidirectional value alignment, with robots accurately inferring human values, combined with effective explanations of the robot's behavior to humans. If these prerequisites are not met, the collaboration may encounter unforeseeable difficulties due to erroneous expectations of teammates [ULS20]. Thus, for robots to become beneficial collaborators in human society, they must be receptive lis-

teners and expressive speakers when interacting with their human teammates.

From the listener's perspective, algorithms such as inverse reinforcement learning (IRL) [AN04b] combine human interactive data with conventional machine learning methods to learn human values in specific tasks [KS09, GSS13]. Assuming (sub-)optimal behavior from human experts, IRL aims to recover the underlying reward function that guides human demonstration. However, acquiring human data in some application domains that arise in military and healthcare contexts can be expensive, if not impossible. Dependence on large datasets also prevents these methods from tackling in-situ, real-time, and interactive human-robot collaboration scenarios. From the speaker's perspective, explainable artificial intelligence (XAI) was introduced to facilitate the alignment of mental models between humans and robots [EGL19]. However, existing XAI systems typically emphasize the generation of interpretable rationales to explain model decisions or predictions, either unfolding the model for a human user to probe and inspect [RSG16, LZS18, EGL19, ZWW20, ZZZ20], or reconciling the discrepancy between the human user's mental model and the robot's counterparts, for a world model [CSZ17, GZ18] and goals [TAH19, HHA19]. Critically, human users' active interactions or inputs to the system only influence how explanations of robots' decisions are generated, but rarely influence the model's decision-making process. This amounts to a **unidirectional** alignment of the mental model as *static machine—dynamic human* communication, where only the human user's comprehension of the robot or the task evolves, given explanations about a *fixed* decision model in machines. In a nutshell, existing XAI systems primarily approach the human-robot communication problem from one of the two communication directions, but seldom from both.

To accomplish **bidirectional** human-robot mental alignment, a more human-centric, *dynamic machine—dynamic human* communication is required. In such a new paradigm, a robot, in addition to revealing its decision-making process, would adopt the user's values and change its behavior in *real-time* so that the robot and the human user would cooperatively achieve a set of common goals. To grasp the user's messages instantaneously, conventional data-driven machine learning approaches are replaced by communicative learning within a cooperative team. Explanations from

the robot will be contextually adapted according to the human's current goals. Such a cooperation-oriented human-machine teaming would require the machine to possess a certain level of ToM: A machine would *actively* infer the user's beliefs, desires, and goals [YLF20, GGZ20]. The system's design will not be limited to explaining its decision-making process, but will also aim to understand human needs for cooperation, therefore forming a *human-centric* and *human compatible* process [Rus19]. This mental alignment process, that can be viewed as one core computation for forming communal and personal common ground [Cla96] that guarantee the coherence of human conversations, embarks the success of human-machine collaboration.

Motivated to build an XAI system with the aforementioned capabilities of understanding the human user's beliefs, desires, and goals while being interpretable to the user, we introduce a sequential decision-making task that requires human-machine teaming to deal with complex constraints over problems intractable to the human's inferential capabilities. Specifically, we devise a human-machine teaming system instantiated as a collaborative game, in which the human user needs to work together with a group of robot scouts to accomplish some tasks and optimize the group gain. In this game, the user and robots communicate on a constrained channel: (i) Only the robots directly interact with the physical world; the user does not directly access the physical world or directly control robots' behavior. (ii) Only the user has access to the ground-truth value that encodes human's desirable end-states, which determine how the task should be completed (*e.g.*, minimizing time, maximizing areas to explore), and the robots have to infer this value function through human-machine interactions. Such a setting constitutes a miniature task that realistically mimics real-world human-machine teaming. Many systems perform autonomously and interact directly with the hazardous environments under human users' supervision, but it is challenging [Sam38] for desirable end-states to be explicitly coded in autonomous agents beforehand, or to change dynamically as events unfold. This setting also follows the classic multiagent system collaboration framework, where agents in the system can work in parallel but may rely on their partners' communication and feedback [Hub88]. To complete a game successfully, robots are expected to accomplish bidirectional alignment by both "listening" and "speaking" wisely. First,

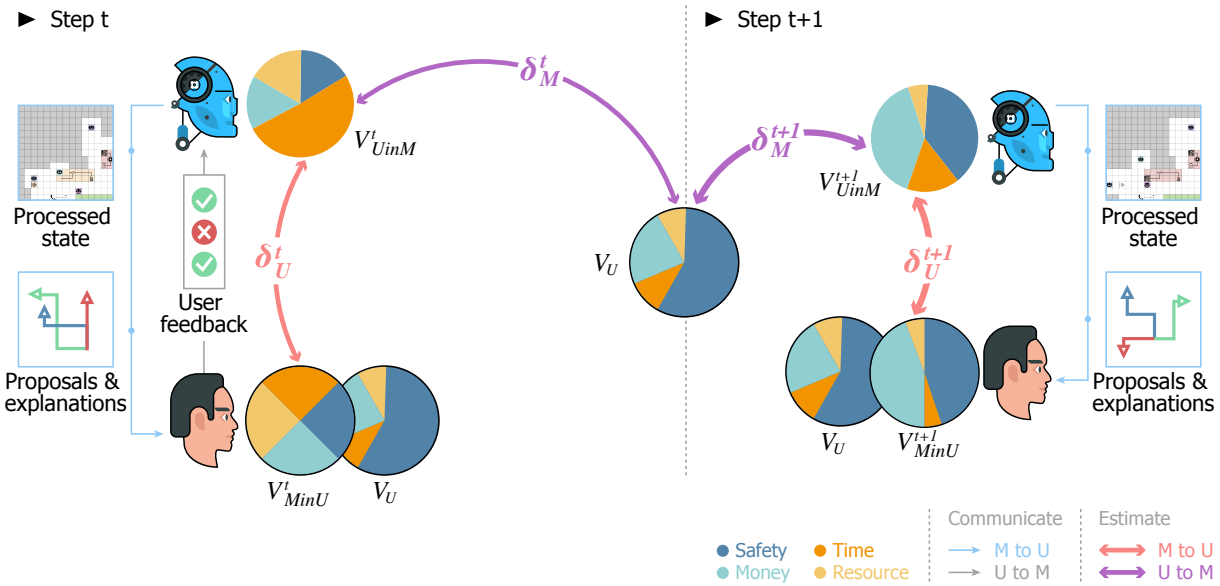


Figure 7.1: **Overview of bidirectional human-robot value alignment.** Pie charts represent the values, *i.e.*, the importance of different goals in a collaboration task, such as simultaneously considering safety, gaining money, saving time, and reserving resources. t in the superscript represents the time step. U and M in the subscript stand for "user" and "machine", respectively. V_U is the user's true value, V_{UinM} is the robot's estimation of the user's value, and V_{MinU} is the user's estimation of the robot's current value. δ denotes the distance between values in the task value space. In every round of interaction, the machine first receives signals from the physical environment and processes its observations to form an abstract state of the environment. Next, the machine presents the processed map together with movement proposals and explanations to human users, who will provide feedback to the system accepting/rejecting the proposals according to human values and current map state. Given the user's feedback, the machine then updates its estimation of human values and takes actions w.r.t. the new values. Cooperative human-robot communication with appropriate explanation aligns the team values in two directions by diminishing the distance between V_{UinM} and V_U, as well as V_{MinU} and V_{UinM}, resulting in final convergence to the true value V_U.

robots need to extract useful information from human feedback to infer the user's values and adjust their policies accordingly. Second, robots are required to effectively explain what they have done and plan to do based on their current value inference, so that the user knows whether the team shares the human values. fig. 7.1 illustrates the bidirectional value alignment process in the game. Taken together, the proposed XAI system aims to address the following two questions: (i) How can robots accurately estimate users' intentions during real-time interaction and feedback? (ii) How can robots explain themselves so that the user can understand their behavior and provide helpful

feedback to aid their value alignment?

To learn human values and intentions, robots make proposals for task plans and ask for the user's feedback (*i.e.*, acceptance or rejection of a proposal), from which the task goals can be inferred. In the collaborative game, knowing that robots are actively learning human values, the user tends to provide helpful pedagogical feedback to facilitate alignment [HLM16]. In particular, every message conveys two aspects of meanings: (a) literal meaning, based on consistency between this message and the value, and (b) pragmatic meaning [Gri75, GS13, SGG14], based on deficiency of alternative feedback. Aware of the user's helpfulness, the robots adopt a human-centric amelioration of iterative teacher-aware learning (ITAL) [YZS21]to learn the human value. ITAL performs maximum likelihood estimation (MLE) based on a two-part likelihood function: The first part models the probability of given feedback being aligned with the human value (literal meaning), and the second part captures the probability of receiving that feedback instead of other alternatives (pragmatic meaning). Leveraging both aspects of meanings, the proposed XAI system demonstrates value alignment in an in-situ, few-round, instantaneous manner, enabling interactive human-machine communication in a cooperative teaming task with a large problem space.

To synchronize the robots' mental status with the human user, our XAI system generates explanations that reveal robots' current estimation of human values and justify the proposed plan. In each step of interaction, to avoid overwhelming the user's cognitive workload with verbose explanations, the robots present customized explanations, such as omitting repetitive signals and emphasizing important updates. The robots model human users' mental dynamics as a Markov process and track the most relevant aspects of the robots' decision process using a sequential statistical graphical model. The explanation that includes all the relevant aspects and best addresses the user's concern at that step will be presented. After receiving explanations from robots and sending feedback to them, the user provides cues to the robots about how satisfying they found the latest proposals and explanations. Using this feedback, the robots constantly update the formats, attention, and contents of the explanations.

To evaluate the performance of our XAI system, we conducted human experiments to examine

the success of bidirectional human-robot value alignment. We adopted three types of explanations, and randomly assigned participants into one of the three groups. Three dependent measurements were used to assess the mental accordance: (i) the consistency between robot's inferred value and human's true value, (ii) human perception of how well robots infer and align with human's value, and (iii) human's cognitive trust [Sim07] of the system. Our results show that the proposed XAI system can achieve bidirectional value alignment in an in-situ, real-time manner for collaborative tasks; the robots can infer the human user's values and make their value estimation comprehensible to the user. We also found that some forms of explanations that benefit the way humans interact with the robots may not necessarily improve the human perception of how well robots infer users' values. These results provide converging evidence supporting the necessity for diverse explanations that promote both the performance quality of robots and their social intelligence [RSP21]. As the goal of an AI collaborator is to reduce the human's cognitive burden and assist task completion, we believe that proactively inferring human values in real-time and fostering human comprehension of the system paves the way for generic human-machine teaming.

7.2 Results

fig. 7.1 illustrates the bidirectional value alignment procedure between the human user and robots during the game. The system's learning algorithm, built on top of ITAL [YZS21], substitutes the conventional likelihood functions for regression or classification tasks by explicitly integrating the Boltzmann rationality human-decision model. The system incorporates both the literal and pragmatic meaning of human feedback to infer the user's value. Meanwhile, the system explains its decision-making process to facilitate human perception of the machine. To test the system, we design a psychological experiment to assess the performance of human-robot value alignment using different forms of explanations. In the coming sections, we first describe the human-robot collaboration game design, followed by an overview of the algorithms we used for bidirectional value alignment and explanation generation. Next, we introduce the human experiment design,

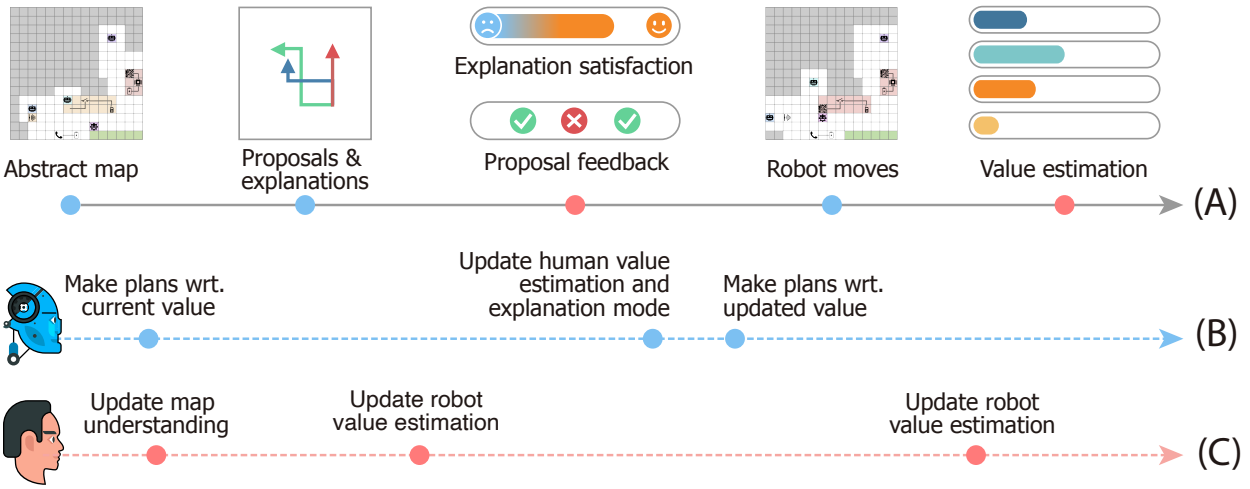


Figure 7.2: **Study design of the Scout Exploration Game**. Timeline **(A)** denotes events happening in a single round of the game, starting from scouts receiving environment signals and ending with their next move. Proposals and explanations are presented differently to users depending on their experimental group; see fig. 7.3 for details. The value estimation asks users to infer scouts' value at current time. Answers to these questions will not be used by the scouts during the game, but only for inspecting users' mental model after the game completes. fig. 7.5**B** shows the detailed UI of these questions. Timelines **(B)** and **(C)** depict mental dynamics of the robots and the user, respectively.

report the empirical results of value alignments between humans and machines, and compare the impact of various types of explanations on value alignment.

7.3 Game design

Our collaborative game, the Scout Exploration Game, involves one human commander and three robot scouts. The game's objective is to find a safe path on an unknown map from the base (located in the bottom right corner of the map) to the destination (located in the upper left corner of the map). The map is represented as a partially observed 20×20 tile board, with each tile potentially holding one of the various devices and remaining unobserved until a robot scout moves close enough to observe (reveal) the tile's contents. Every scout has the same probabilistic observation model, specified later in the "Observation model" section.

There is structural interdependence between human and scouts in the game [JV19] in the fol-

lowing way: (i) the user depends on the scouts to explore the dangerous area and defuse bombs, and (ii) the scouts need the user to provide feedback to better understand the goal of the current mission. We define a set of goals for the robot scouts to pursue as they find the path to reach the destination, including (i) saving time used to reach the destination, (ii) investigating suspicious circuits/bombs (loops of devices connected with wires) on the map, and (iii) exploring tiles, and (iv) collecting resources (gold bricks on the map). The game's performance is measured by these factors, *i.e*. the accomplishment of these goals by the robot scouts, and their relative importance (weights), defined as the human user's value function. Although all goals have intrinsic benefits, a trade-off among the goals has to be made according to the value function. For instance, if time is valued more in the value function than resources, the scouts should ignore some resources along the way to the destination for the sake of time. To emphasize the trade-off essence of value functions, we represent the importance of each factor with a percentage and the four percentages sum to 1. Before the interaction begins, a value function is assigned **only** to the human user as the mission for the game. Just like in the real world, various tasks can be specified with distinctive functions defined on a unified set of features [BDM17]. We coined seven value functions in this study to cover diverse types of tasks: four of them have one dominant goal, two of them have two equally important goals, and the rest values everything but resources.

We mimic a realistic scenario, where human needs can be too diverse to code in the robot beforehand and value functions can be difficult to transfer between human and machine due to different mental representations. Without knowing the value function, to complete a task, the robot scouts (as a team) must quickly infer the commander's value. In each step, we let the robot team make three movement proposals, one for each scout, to the user, and the user can either accept or reject a proposal. To help the commander make decisions, the robot team also explains the reason for every proposal. With the user's feedback, conditioned on the interaction history as well as the current map status, the robot team adjusts its estimation of the human value and takes actions accordingly. Specifically, if a plan is accepted, the proposer will follow that plan as much as possible (a plan may be interrupted by unexpected blocks in the partially observable map);

otherwise, the robot will execute a new plan with the updated value estimation. We only allow the robot team to make proposals once in every round so that they must rely on their own autonomy to complete the task, instead of proposing until acceptance and effectively being teleoperated by the user. This concludes one round of interaction, and this process will be repeated. Of note, to avoid inefficient communication, the robot only makes proposals when necessary and acts according to the basis of the latest human value estimation. fig. 7.2 summarizes the human-machine interaction flow.

This game is complex through the lens of the combinatorial game theory. With an average planning step of 35 and a branching factor of 2^{12}, the estimated game tree complexity is of 10^{126} for scouts to generate task plans. In comparison, chess has a game tree complexity of 10^{123}. On the player side, with an average round of feedback of 18 and a branching factor of 8, the estimated game tree complexity is of 10^{16} for the player to provide feedback.

7.4 Bidirectional value alignment

Bidirectional value alignment, as one of the primary contributions, provides a more human-centric, *dynamic machine—dynamic human* communication framework for human-robot teaming. To estimate the human user's value during the communication process, we integrate two levels of ToM into our computation model. The level-1 ToM encodes the cooperative assumption. Namely, given a cooperative human user, the accepted proposals are more likely to align with the correct value function than the rejected ones. The level-2 ToM further accommodates users' pedagogy into the model. That is, the feedback that drives robots' value closer to the true value is more likely to be selected than other alternative feedback combinations. The pedagogical inclination requires an additional level of ToM because it demands recursive modeling of the user's model of the robots. Combining both levels of ToM, we formulize human behavior with distributions parameterized by the value and develop a learning algorithm with a closed-form parameter update function; see details in the **Human-robot value alignment** section.

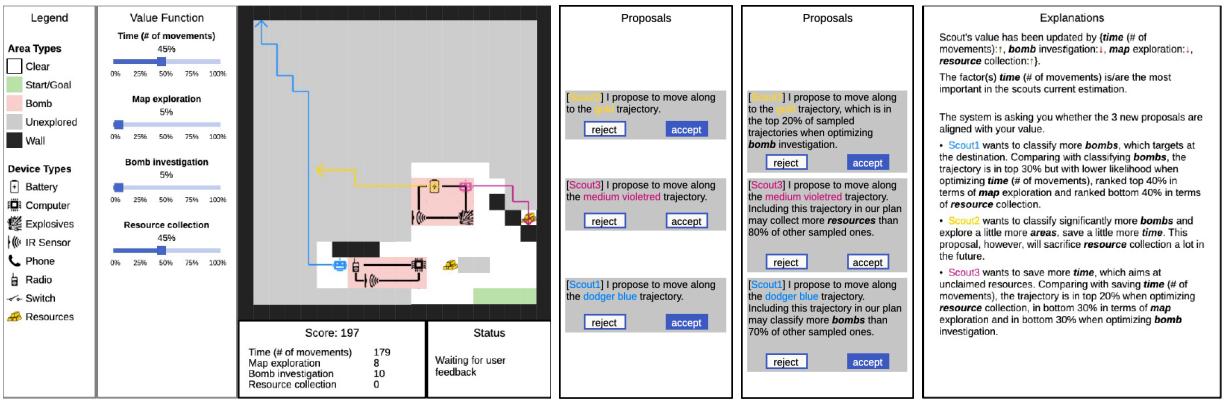
To facilitate such a bidirectional alignment and gain human trust, we provide different forms of explanations along with proposals, which unveils the rationales behind scouts' proposals. Specifically, the explainer takes in current estimations of two levels of ToM as semantic input and fills it in a syntactic template. To provide *concise* explanations that are interpretable *to* humans and facilitate learning *from* humans, we devise a sequential generation process that selects templates by taking human's preference over previously observed explanations (*i.e.*, satisfaction score) into consideration. We call such preferences human's explanation utility; see details in the **Utility-aware explanation generation** section.

7.5 Human experiment

7.5.1 Experimental design

The human study examines whether our XAI system achieves real-time bidirectional value alignment between the human and the machine. In particular, we evaluate the efficacy of different forms of explanation of the robots' plans to human users. We conducted a psychological study with 135 participants. Participants were randomly assigned to one of three groups: (i) a proposal-only group (ii) a brief-explanation group, and (iii) a full-explanation group, each with 45 participants. In the proposal-only group, the scouts only make proposals and give no explanations to the human. In the brief-explanation group, every proposal consists of a one-sentence brief explanation, explicating its positive outcomes. In the full-explanation group, a more detailed explanation accompanies every proposal, expounding the gains and costs of scouts' tentative actions and the dynamics of their values for the importance of different goals. Across all three groups, the robot scouts follow the same action policy and decision process for belief updating. The three groups differ only in terms of the forms of explanations provided to the human participants. fig. 7.3 compares the game interface that appeared in each group.

Our experimental setup consists of three phases: introduction, familiarization, and game playing. The first two phases prepare participants for the game. Game overview, rules, and UI are



D

	A	B.0	B.1	C
Proposal	✓	✓		
Brief-Explanation	✓		✓	
Full-Explanation	✓		✓	✓

Figure 7.3: **User interface for the scout exploration game.** (**A**) From left to right: Legend panel in the first column explains the meaning of various icons used in the game. The value function panel in the second column shows the true values indicating the relative importance of various goals; the values are unknown to the robot scouts and cannot be modified by the user. The four right panels change dynamically over the course of the game. The central panel in the third column shows the current status of the map in the game. The score panel at the bottom shows the current scores for achieving individual goals. The overall score is the sum of the scores for individual goals, weighted by the values known to human users in the value function panel. The Status panel provides a text summary of the current status of the robot system. (**B**) The Proposal panel shows the robot scouts' current proposals; human users can accept or reject proposals of individual scouts. In the proposal-only group, participants only see a descriptive sentence for each proposal (**B.0**), whereas, in the brief-explanation and full-explanation groups, participants are presented with a brief explanation about the proposal's purpose (**B.1**). (**C**) The Explanation panel shows detailed explanations provided by the scouts, only displayed to the full-explanation group.(**D**)The bottom table summarizes key components of the game display included in each group.

explained in the introduction phase. In the familiarization phase, the system tests participants with a set of questions to validate their understanding of the game; only participants who correctly answer all the questions in the familiarization phase proceed to the game playing phase. During the game, participants are asked to accept or reject scouts' proposals and assess satisfaction with the

scouts' communication after every feedback. The feedback for proposals are given using buttons shown in fig. 7.3**B**. The satisfaction are provided via Likert-scale questions shown in fig. 7.5**A**. In addition, we also ask participants to estimate the machine's internal states, such as the scouts' current value function and their qualitative trust of the XAI system. The scouts' value estimation questions are asked every 2 rounds of communication, and the trust questions are asked every 5 rounds. fig. 7.5**B** and fig. 7.5**C** show the scouts' value estimation question and the trust question respectively. Note that human judgments about the value estimation and trust are not used to adjust scouts' behavior; these additional measures are only used for evaluation purposes.

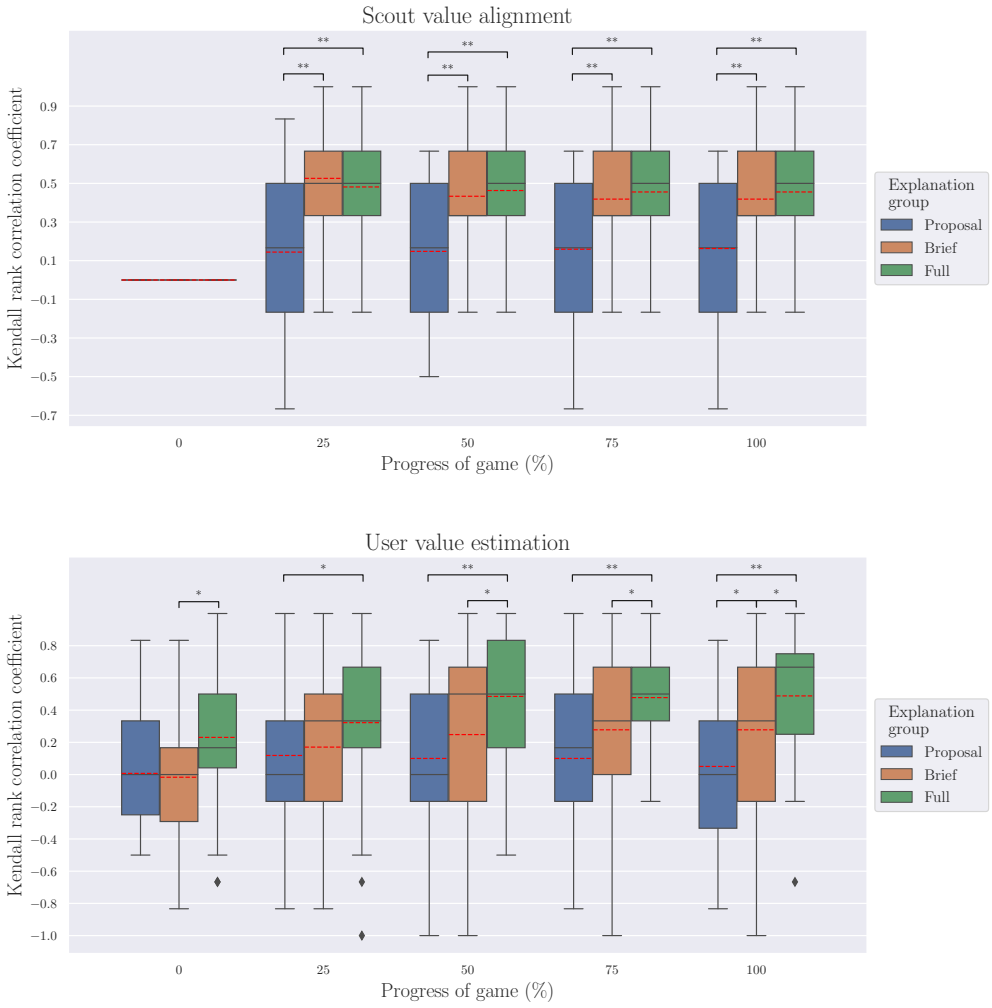


Figure 7.4: **Results of value estimation for scouts and humans in three groups.** The legends: proposal, brief, and full refer to the proposal-only group, the brief-explanation group, and the full-explanation group, respectively. Horizontal axis indicates the progress of the game for human participants; vertical axis indicates Kendall's rank correlation coefficient between estimated values by scouts and humans; higher correlation indicates better value alignment. Top panel **A**: correlation between scouts' value estimate and the true values that are known to human users as a function of game progress (*i.e.*, scout's accuracy in estimating human values). Before the game starts, the scouts' value estimate is initialized as uniform across all goals. Bottom panel **B**: correlation between the human estimate of the scouts' values and scouts' estimate of the true values as a function of game progress (*i.e.*, humans' accuracy in estimating scouts' values). Asterisks in the plot indicate significant group differences in paired t-test with P-value smaller than 5%. The error bars indicate the observation minimum and maximum. The solid lines and red dashed lines in the bars respectively indicate the median and mean.

7.5.2 Human study results

fig. 7.4 illustrates the bidirectional human-robot value alignment results for all three groups. We compute the Kendall rank correlation coefficient, commonly referred to as Kendall's τ coefficient, to assess the value alignment between scouts and humans. To compare two sets of values (*e.g.*, values estimated by scouts versus true values known to human users), we first rank the task goals by their corresponding values and then calculate the Kendall's τ coefficient between the two rankings. Perfectly agreed/disagreed rankings have $\tau = \pm 1$, and independent rankings expect $\tau \approx 0$. To demonstrate dynamic changes in bidirectional value alignment between scouts and humans, we record the scouts' estimation of the user's value and measure the user's estimation of the scout's value as the game proceeds. Since games have various lengths due to different explanation formats, group values, and individual differences in participants, we normalize game progress as a percentage calculated by dividing the number of current iterations by the total number of iterations. For all three groups, we remove subjects who fail attention checks and outliers whose alignment results are 1.5 Interquartile Range Method (IQR) below the 25th percentile or above the 75th percentile at any game progresses.

fig. 7.4**A** shows the alignment between robots' estimated values and the true values known to human users. First, all groups show higher value alignment at the end of the game compared to the beginning of the game [paired t-test, $t_{Prop}(44) = 2.850, P_{Prop} = 0.007, t_{Brief}(44) = 10.148, P_{Brief} < 0.001, t_{Full}(44) = 11.452, P_{Full} < 0.001$]. Importantly, scouts that interacted with the brief-explanation and full-explanation groups show stronger value alignment, revealed by higher correlations between scouts' estimated values and true values, than alignment in the proposal-only group ($\tau = 0.2, 0.4, 0.5$ for the proposal-only , brief-explanation and full-explanation groups respectively). The group differences emerge in early stages of the game (25% of the game progress), and are maintained to the end of the game, confirmed by analysis of variance (ANOVA) at a range of progress points [from game progress 25%, 50%, 75% and 100%, $F_{(2,132)} = 19.086, 14.202, 11.961, 11.622; P < 0.001, P < 0.001, P < 0.001, P < 0.001$]. Better value alignment in the two groups involving explanation than the baseline proposal-only group without explanation provides strong evidence that

explanations about robot decision processes to human users enhance bidirectional communications between humans and machines. The enhanced communication, in turn, helps machines gain accurate estimates of human values, thereby fostering human-machine teaming. There is no significant difference between the brief-explanation group and full-explanation group, implying that the detail of the explanations may not critically influence humans' feedback in terms of accepting or rejecting robots' proposals, as long as these explanations provide sufficient contexts to justify the robots' intents.

fig. 7.4**B** depicts how well the human users estimate the scouts' values over the progress of the game (*i.e.*, the accuracy with which humans assess the scouts' values). fig. 7.5**B** shows the interface we used to collect human estimates in the experiment. An ANOVA test revealed a significant main effect of groups in the later stage of the game playing at game progress 50%, 75% and 100%, [$F_{(2,132)} = 7.632, F_{(2,132)} = 8.339, F_{(2,128)} = 10.542, P = 0.001, P < 0.001, P < 0.001$ respectively]. The brief-explanation and full-explanation groups show significant enhancement of alignment between human estimates of scouts' values and scouts' values used in determining their decision and proposals at the end of the game compared to the beginning of the game [paired t-test, $t_{Brief}(44) = 3.272, P_{Brief} = 0.002, t_{Full}(39) = 2.810, P_{Full} = 0.007$], whereas the proposal-only group does not show any improvement [paired t-test, $t_{Prop}(38) = 0.286, P_{Prop} = 0.776$]. These results suggest that human users have difficulty understanding robots' intentions by only observing their situational behaviors, highlighting the central role of explanation in revealing the robots' intentions to its human user. Critically, humans show stronger alignment in estimating scout's values in the full-explanation group than in the other two groups in the second half of the game [independent sample t-test, from game progress 50%, 75% and 100%, $t(88) = 4.291, t(88) = 4.511, t(84) = 5.088, P < 0.001, P < 0.001, P < 0.001$ against proposal-only ; $t(88) = 2.387, t(88) = 2.219, t(86) = 2.196, P = 0.019, 0.030, 0.031$ against brief-explanation]. In comparison, the brief-explanation group only yields a more consistent human estimate of scouts' values than the proposal-only group at the end of the game [independent sample t-test, $t(86) = 2.274, P = 0.026$]. Taken together, these results indicate that both forms of explanation

facilitate human estimates of robots' value estimates based on observation of robots' behavior and interactions with the robots. But the full explanation, which provides details about both the advantages and the disadvantages of a proposal, is more helpful to human judgments about the robots' estimates than a brief explanation showing only the major benefit of a proposal.

Both results from robots' estimate of human values in fig. 7.4**A** and from human estimate of robots' values fig. 7.4**B** show that groups with brief and full explanations can maintain a stable trend of value alignments across the entire human-robot teaming process. However, the emergence of value alignment differs across different groups and varies for different alignment metrics over the course of the game. Robots' value alignment metrics measured by the Kendall's τ coefficient converges at 25% of the game. Alignment of human estimates and scouts' values converges at 50% progress for the full-explanation group, and 75% game progress for the brief-explanation group. These results demonstrate our system's capability to maintain the established team mental model during continuous human-robot teaming, with full explanations enabling faster convergence of users' estimates of robots' mental status (estimates of values). The convergence of both alignment metrics shows that (i) our value alignment algorithm enables the robot scouts to learn human values in an in-situ, real-time, and interactive manner, and (ii) explanations generated by the robots enable users to better perceive the machine's values. These results demonstrate a bidirectional human-robot alignment. Moreover, our result pins down the contributions of explanation formats in different facets of human-robot communication. We found that brief and full explanations lead to similar effects in improving the way humans provide feedback to the machine via acceptance or rejection of robots' proposals. However, the full-explanation group shows a significantly greater benefit for human accuracy in estimating robots' values.

7.6 Discussion

Our proposed XAI system successfully demonstrates the feasibility of a bidirectional human-robot value alignment framework. From the listener's perspective, robots in all three explanation groups

can quickly align to the user's value by correctly ranking at least 60% of goals' importance as early as the 25% progress of the game. From the speaker's perspective, by providing proper explanations, robots can reveal their intentions to the user and facilitate better human perception of the machine's values, with convergence occurring at 50% (full-explanation) and 75% (brief-explanation) of the game. Together, both perspectives provide convincing evidence of a bidirectional process of value alignment. Specifically, (i) by receiving cooperative human feedback, robots gradually update their value function to align with the human values, and (ii) by continuously interacting with the robots, the human user gradually forms a coherent perception of the system's capability and intentions. Although the system's values have not converged in the first half of the game, the user's perception of the robots' estimate can still improve. Eventually, when the robots' values become stable, the user's estimation of the robots also becomes stable. The pairing of convergence from robots' estimate of the user's values to user's true values, and from user's estimate of the robots' values to robots' current values forms a bidirectional value alignment anchored by the user's true value.

Despite showing similar converging trends of value alignments, the three explanation groups differ in the precision of their alignments. In both directions of human-robot estimation (*i.e.*, scouts estimating human values and the human's understanding of the scouts' current value estimation), the Kendall's τ coefficients of the proposal-only group are significantly lower than the coefficients of the other two groups. These gaps suggest that human-machine interactions alone are not sufficient to enhance the human perception of the machine, nor are they sufficient to evoke better human feedback/guidance to the robots. Results from the computational ecology models show that a multiagent system can converge to an equilibrium point only when the information delay and uncertainty between agents are fairly small. On contrary, our modeling framework can handle relatively large amount of uncertainty. In our case, the scouts' explanations play an important role in reducing information uncertainty and system convergence: Explanations help the human understand the machine's current value estimation and generate a better response, which in turn enables the machine to estimate the value more accurately. The extent of human-robot mutual

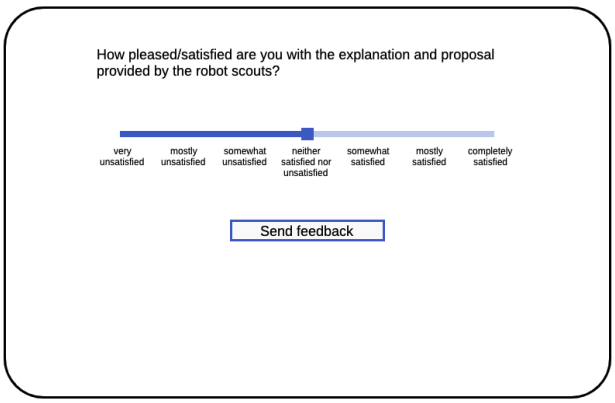
understanding depends upon how an AI system explains itself to the user. In the absence of informative explanations, certain misconceptions cannot be eliminated even with continuous interaction between humans and robots, leading to a slower value alignment.

Compared with the brief-explanation group, the full-explanation group demonstrates significant enhancement in the human estimate of robots' values, but does not show a strong advantage in terms of scouts' value alignment. These results indicate that human users in the two explanation groups provide feedback to the scouts in similar ways, although the full-explanation group acquires a more accurate understanding of the system. One possible cause of this dissociation is that human users exhaust their cognitive resources when processing other complexities of the game, such as comprehending messages from the three scouts or analyzing information on the map. Thus, additional details in the full explanation cannot be accommodated to offer more rationale feedback. An alternative possible reason involves the design of the game, in particular, the granularity of the feedback. As there are only eight possible feedback combinations (two for each scout's proposal) in every round, it is possible to identify the best response to the scouts with only a limited understanding of the system status and the proposals. The extra information provided in the full explanation, though beneficial to the user's perception of the machine, maybe useless for scouts' value learning.

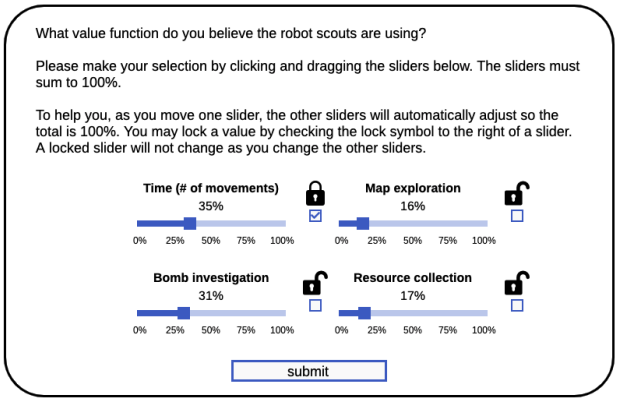
We also measured users' qualitative trust in the system as the game proceeds. However, we did not find any significant differences across the three explanation groups. This result suggests that human trust toward machines depends on many facets of the machines [WPH16, RSP21]. Both social intelligence and performance quality of robots are indispensable to fostering trust [RSP21]. Better bidirectional value alignment can improve team performance, but may not be sufficient to enhance the human perception of scouts' social intelligence through short-term human-robot collaborations. Also, in the current game, scouts are not likely to make catastrophic mistakes throughout the task, and are guaranteed to reach the upper left corner of the map successfully. Since the robots can always accomplish the task in the end, users may tend to trust the robots, so that explanations have less impact on trust formation.

To summarize, we present a bidirectional human-robot value alignment framework and use an XAI system to verify its feasibility. The proposed XAI system demonstrates that, with ToM integrated into the machine's learning module and appropriate explanations provided to the user, humans and robots are able to achieve alignment of mental models through an in-situ, real-time and interactive manner. The coherent computational framework reported in our study provides promising results to address the question raised at the beginning of this article, "what constitutes a good human-robot team", by contributing to the formation of a shared mental model between a human and a machine. Particularly, our work focuses on the **task-specific** aspects of the mental model, namely the value and intentions. In more intricate scenarios, mental alignments can further entail other aspects going beyond the context of a single task, *e.g.*, capabilities of every team member, prerequisites and outcomes of actions (also referred to as the world transition model in reinforcement learning (RL)), individual duties and roles. These components in the mental model are useful across various task contexts. In human language using, such a mental alignment process is often referred to as personal common ground and can be established via episodic evidence [Cla96], *i.e.*, the actions or events the speakers are part of together. In our setting, the episodic evidence could be acquired from human-robot collaboration in multiple games, possibly with different value functions and maps. As such, the universal human model described in eq. (7.2) and eq. (7.6) can be replaced by a customized model parameterized for individual person's characteristics.

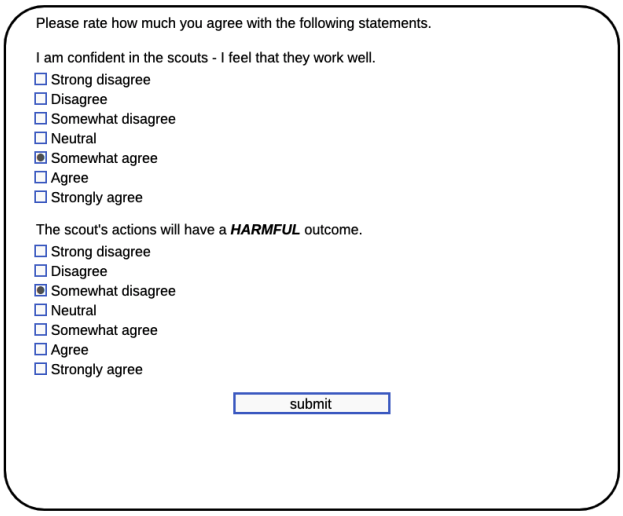
In this work, we focus on the alignment of value functions, which captures the relative importance of a wide range of goals. Aligning the values can greatly help the human and machine establish common ground for task-oriented collaborations. Thus, we consider our work as the first step towards a more general mental model alignment setting in human-machine collaboration. In future work, we plan to explore factors that can further enhance human users' trust (*e.g.*, enabling counterfactual queries to the robots), validate the effects of alignment on task performance, and apply our system to tasks involving more complicated environments and value functions.



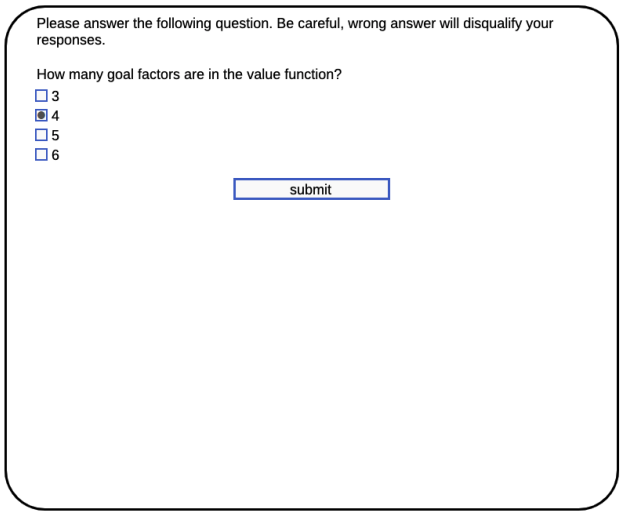
(**A**) Explanation/proposal satisfaction question



(**B**) Value estimation question



(**C**) Qualitative trust question



(**D**) Attention check question

Figure 7.5: **Examples of questions participants received during the game. (A) Explanation/proposal satisfaction question.** Participants are asked to provide a satisfaction score for the explainer in every round when they receive scout's proposals and explanations. This satisfaction score is used to update models for generating future explanations. **(B) Value estimation question.** Participants predict the robot scouts' belief about the true human value by sliding the bars to set a relative importance of each goal; of note, this is a question about level-2 ToM. Our interface ensures that the total value of all goals sums to 100%; if the participant moves one slider, the others will automatically change proportional w.r.t. their original values, such that all values still sum to 100%. Meanwhile, participants can lock a particular slider by checking the lock symbol to the right of the slider. **(C) Qualitative trust question.** We ask the participants "how confident you are in the scouts?" and "how much do you think the scout's actions will have a **HARMFUL** outcome?" **(D) Attention check question.** These questions are shown after trust questions; participants receive one of the four questions about the game logic and UI. Participants who failed the attention check are later removed from data analysis.

7.7 Game setup details

We implement this game using HaxeFlixel, a 2D game engine for JavaScript-based games, such that participants can access the game on web browsers; this setup is necessary in the situation of COVID-19. Our between-subject design is divided by the explanation format provided to the participants: the proposal-only group, the brief-explanation group, and the full-explanation group. The proposal-only group only shows the proposed trajectory on the map and a basic descriptive text about the proposal, such as "Scout 1 proposes to move along the blue trajectory." For the brief-explanation and full-explanation group, brief explanations accompany the proposals to clarify the motivation of the robot scouts; *e.g.*, "Scout 1 proposes to move along the blue trajectory, which is in the top 1% of sampled trajectories when saving time." The full-explanation includes a more detailed full explanation besides the brief one in the proposal panel; *e.g.*, "Scout 1 wants to save more time at the cost of map exploration and resource collection." More details about explanations can be found in the "Explanation Generation" section. The full user interface of the game is displayed in fig. 7.3 with the actual explanations used in the game.

After giving feedback to the scouts' proposals, participants are asked a few questions before the next round of explanation and proposing. These questions are, by the order of showing up, satisfaction about the latest proposal and explanation, value estimation, qualitative trust, and attention check. Only answers to the satisfaction questions are used by the system's explainer for explanation utility tracking; all other questions are used only for post-game analysis. fig. 7.5 includes some example questions queried during the game. To avoid overwhelming users, value estimation questions are asked every 2 proposals, and qualitative trust and attention check questions are asked every 5 proposals.

Algorithm 6: Overview of the Scout Exploration Game

Set $t = 1$, initialize s^t, agent's mental state x_0^R;
while not task-complete(x_{t-1}^R) **do**
  $o_t \sim$ observation_model(s_t) // collect observations from the environment
  $\widehat{x_t^R} =$ update_state_belief(x_{t-1}^R, o_t) // update belief given observations
  $m_t^R \sim$ proposal_explanation_generation($\widehat{x_t^R}$) // generate messages (proposal & explanation) to the user
  $x_t^R =$ update_value_belief($\widehat{x_{t-1}^R}, m_t^R, m_t^H$) // update beliefs given user feedback
  $\mathbf{a}_t^R \sim$ action_policy(x_t^R) // agent's policy
  $s_{t+1} \sim$ game_dynamics($s_t, \mathbf{a}_t^R$) // state transition
  $t = t + 1$
end

7.8 Computational model details

7.8.1 Overview

Before diving into the technical details of how the proposed robot scouts act, align value, and interact with the human user in a bidirectional communicative learning framework, we first provide an overview of the game flow and the notations of the computational model. We use R and H to denote the robot scouts and the human user, respectively. θ encodes the parameters of the value function, s the physical state, υ the utility of explanations, $b(\cdot)$ the belief over latent variables. $x^R = \big(b(s), b(\theta), b(\upsilon)\big)$, the mental state [RCS92, GN07] of the robots (the robot team shares one mental state), depicts their current beliefs of all the unknown task-relevant variables. m the message used for human-machine communication. In every round of the game, the robot scouts receive observations from the environment and make a task plan based on their current mental state. Next, they send messages (proposals and/or explanations) to the human user for feedback; this user feedback is used for robots' final movement plans in this round. Alg. 6 sketches the high-level game flow, and fig. 7.6 shows the computation pipeline for one round of human-machine teaming.

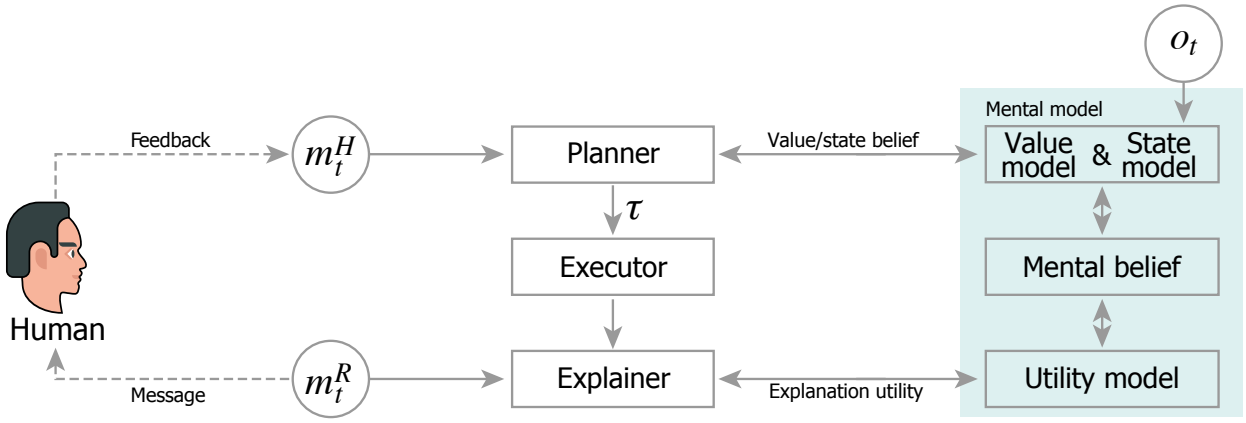


Figure 7.6: **Algorithmic flow of the computational model.**

Of note, one comparable but different setting to our human-machine teaming framework is IRL [AD21]. Nevertheless, IRL aims to recover an underlying reward function given pre-recorded expert demonstrations in an offline passive learning setting. In contrast, the robot scouts in our setting are designed to learn *interactively* from *scarce* supervisions given by the human user. Crucially, our design requires the robots to actively infer the human user's value in **real-time** and **in-situ** as the task proceeds. Furthermore, to consummate a collaboration, not only must the robot scouts quickly comprehend the human user's intent, but also elucidate themselves to ensure smooth communication with the human user throughout the entire game. In brief, the robots are tasked to perform value alignment by inferring the human user's mental model, actively making proposals, and evaluating the human user's feedback, which requires complex and recursive mind modeling of the human user.

In the coming sections, we will introduce how the robots select actions, make proposals, update belief of human user's value function, and generate communication messages.

7.8.2 Action selection

Suppose the robot scouts already know about the human user's value function. The game simplifies to a partially observable Markov decision process (POMDP) setting, solvable by planning-based

methods [SV10]. Let τ_i denote the plan proposed by the i-th scout and $\tau = \{\tau_1, ..., \tau_K\}$ as the complete plan of the scout group, where K is the number of scouts in the group. When constructing a plan, the scouts utilize the following policy:

$$\begin{aligned} \arg\max_{\tau \in \mathscr{T}} E_{s\sim b(s),\theta\sim b(\theta)}[\theta^T f(\tau, s)] &= \arg\max_{\tau \in \mathscr{T}} E_{s\sim b(s)}[f(\tau, s)]^T E_{\theta\sim b(\theta)}[\theta] \\ &\approx \arg\max_{\tau \in \mathscr{T}} \bar{\theta}^T \big(\frac{1}{N_S} \sum_{n=1}^{N_S} f(\tau, s_n)\big) = \arg\max_{\tau \in \mathscr{T}} \bar{\theta}^T \overline{f(\tau)}, \end{aligned} \tag{7.1}$$

where $f(\tau, s)$ is the status of the four goals in the terminal game state given that current state is s and the scouts follow the plan τ; we call it the features of (τ, s). The first equality holds because s and θ are independent of each other in our setting. Given the dynamics of the game, f can be forward simulated in our planner, such that the expectation of $f(\tau, s)$ can be approximated using Monte Carlo methods with N_S state samples, giving us $\overline{f(\tau)}$, the feature of τ. Instead of computing the full distribution, the agent only needs to keep track of the mean of the belief over human user's value function as we are using a linear model to calculate the gain of the game; we use $\bar{\theta}$ to denote the mean of $b(\theta)$. Since the space of all possible plans is too large ($(20 \times 20)^K$) to be calculated exactly, we use heuristics to approximate the space of all possible plans by constructing another space $\mathscr{T}$ and select the optimal plan from it. After a plan τ is determined, the joint action of all robot scouts is the first move of the generated plans $\mathbf{a}^R = (\tau_1[0], \ldots, \tau_K[0])$.

7.8.3 Proposal selection

To improve user experience during the interaction and foster human trust, the robots ought to make good proposals at the proper time to collect users' informative feedback. In active learning [RLG18], the query usually maximizes the expected information gain. However, such criteria of asking questions to an oracle cannot be applied to human-robot interaction (HRI). Critically, besides acquiring information from the human, robots' questions also ought to reveal their mental status to the user and gain their trust. Of note, a clear dilemma always exists: The proposal with the most expected information gain is usually the most uncertain one as well, querying of which easily

leaves an unreliable impression of the system to the user and impairs the human perception of the machine's value. To tackle this issue, we design our communicative learning framework such that the scouts will propose the optimal plan given their current estimation of the user's value. Such proposals can reveal the robot team's current mental status to the user for better human perception of the scouts, hence receiving more helpful supervision. For instance, if all three proposals ignore suspicious devices, but bomb exploration is an important factor, the user will be aware of the discrepancy between the scout's value and the intended value and adjust it with feedback. As a result, plans used for proposals are calculated in the same way as plans for action selection described in the last paragraph, only with $b(\theta)$ from the previous time step.

7.8.4 Human-robot value alignment

Level-1 ToM The robot scouts need to estimate the human user's value from their interactions. In the collaborative game, the more a proposal facilitates goals with high values, the more it is likely to be accepted. In this paper, we refer to the ability to infer humans' value from their actions as level-1 ToM. Bearing level-1 ToM, the scouts can interpret the user's feedback and update the value estimation given the current map status. For example, if a trajectory towards a partially explored circuit is accepted, the scouts are likely to increase the value to bomb investigation and lower the other goals. We integrate level-1 ToM into our computation model and develop a learning algorithm with a closed-form parameter update function.

Belief update with level-1 ToM Let $m^H(fb)$ denote the human user's feedback, which is a binary code with the i-th bit indicating the acceptance or rejection of the proposal from the i-th scout. Assuming the human user considers each proposal separately and follows a Bernoulli

acceptance distribution [CNS18], the likelihood function of the human user's feedback is:

$$\begin{aligned} p\big(m^H(fb)|\tau;\bar{\theta}\big) &= \prod_{i=1}^{K} p\big(m^H(fb)_i|\tau_i;\bar{\theta}\big) \\ &= \prod_{i=1}^{K} \frac{\exp\big(\beta_1 \bar{\theta}^T \overline{f(\tau_i)}\big)^{m^H(fb)_i} \exp\big(\beta_1 \bar{\theta}^T \overline{f(\neg\tau_i)}\big)^{(1-m^H(fb)_i)}}{\exp\big(\beta_1 \bar{\theta}^T \overline{f(\tau_i)}\big) + \exp\big(\beta_1 \bar{\theta}^T \overline{f(\neg\tau_i)}\big)}, \end{aligned} \tag{7.2}$$

where

$$\overline{f(\tau_i)} = \sum_{\tau\in\mathscr{T}:\tau_i\in\tau} \overline{f(\tau)}, \quad \text{and} \quad \overline{f(\neg\tau_i)} = \sum_{\tau\in\mathscr{T}:\tau_i\notin\tau} \overline{f(\tau)}. \tag{7.3}$$

That is, a proposal is more likely to be accepted if including it in the scouts' plan is more beneficial than excluding it when $\bar{\theta}$ is the value parameter. Given this likelihood function, we utilize MLE to learn $\bar{\theta}$ by maximizing $\log p(m^H(fb)|\tau;\bar{\theta})$ w.r.t. $\bar{\theta}$:

$$\bar{\theta} = \bar{\theta} + \eta \frac{\partial \log p(m^H(fb)|\tau;\bar{\theta})}{\partial \bar{\theta}}, \tag{7.4}$$

where η is the learning rate, and

$$\begin{aligned} \frac{\partial \log p(m^H(fb)|\tau;\bar{\theta})}{\partial \bar{\theta}} =& \beta_1 \sum_{i=1}^{K} \Big[m^H(fb)_i \overline{f(\tau_i)} + \big(1 - m^H(fb)_i\big) \overline{f(\neg\tau_i)} \\ & - E_{m\sim p(m^H(fb)_i|\tau_i;\bar{\theta})} \big[m\overline{f(\tau_i)} + (1-m)\overline{f(\neg\tau_i)} \big] \Big]. \end{aligned} \tag{7.5}$$

where acceptance/rejection selects the feature of including/excluding τ_i and the expectation is taken w.r.t. the feedback distribution given current $\bar{\theta}$. The expectation computes the average feature if the plan, τ, is randomly accepted/rejected according to current $\bar{\theta}$. The difference between the user's designated feature and the expected feature forms the gradient. Since $\bar{\theta} > 0$ and $\|\bar{\theta}\|_1 = 1$, we perform MLE with the projected stochastic gradient ascent algorithm.

Level-2 ToM Intuitive but limited, the comprehension of feedback endowed by level-1 ToM is constrained to its plain content, *i.e.*, the literal meaning of the feedback. In human communication, messages often convey both literal meanings and pragmatic meanings [SGF13]. In other words,

one can acquire not only explicit information from what others said but also implicit information from what others did not say. A typical concretization is the Gricean Maxims of quantity [Gri75] or the scalar implicature: When people say "I like drinking warm coffee," though the lexical meaning of "warm" is semantically close to "hot," they mean "not hot"; otherwise people would have said "hot" directly [Car98, VBP13]. Similarly, the human user's selection of a certain combination of feedback but not other combinations can also help robot value alignment. To comprehend this process, it requires the robots to mentally simulate and plan based on human users' pedagogical tendency and belief about the robots' current plan. We refer to such a recursive inference ability as level-2 ToM.

Belief update with level-2 ToM To enable level-2 ToM, robots need to conduct a recursive mental simulation in a counterfactual fashion and consider the advantage of the received feedback over others not being sent. Intuitively, suppose the user knows how the robots with level-1 ToM update the value given feedback; the more the feedback leads to changes towards the ground-truth value, the more it is likely to be selected. Computationally, the level-2 robots first simulate level-1 value update given all possible feedback. Next, the robots find a ground-truth value such that the update brought by the received feedback is better than the other alternative feedback. Mathematically, we formulate the human user providing feedback based on its anticipatory improvement following

$$q(m^H(fb)|\bar{\theta},\tau;\theta^*) = \frac{\exp\left(-\beta_2\|\bar{\theta}+\eta\frac{\partial \log p(m^H(fb)|\tau;\bar{\theta})}{\partial\bar{\theta}}-\theta^*\|^2\right)}{\sum_{\widehat{m^H(fb)}\in FB}\exp\left(-\beta_2\|\bar{\theta}+\eta\frac{\partial \log p(\widehat{m^H(fb)}|\tau;\bar{\theta})}{\partial\bar{\theta}}-\theta^*\|^2\right)}, \tag{7.6}$$

where $\beta_2 \geq 0$ controls the extremeness of the Boltzmann rationality, η is the learning rate, and θ^* is the set of ground-truth parameters of the value function possessed by the human user. The intuition of this equation is: The feedback from the human user is sampled from a soft-min distribution of the distance between the updated parameters given the feedback and the ground-truth parameters. The smaller the distance is, the larger the improvement brought by that feedback, and the larger the improvement is, the more likely the feedback is provided. Further analysis of the above distance

can be found in Liu *et al*. [LDL18]. Integrating this feedback function into our value learning algorithm, we can derive a new parameter update function:

$$\bar{\theta} = \bar{\theta} + \eta g\left(m^H(fb)\right) + 2\beta_2\eta^2\left(g\left(m^H(fb)\right) - E_{m(fb)\sim q(m(fb)|\bar{\theta},\tau;\theta^*)}\left[g\left(m(fb)\right)\right]\right), \tag{7.7}$$

where

$$g(m(fb)) = \frac{\partial \log p(m(fb)|\tau;\bar{\theta})}{\partial \bar{\theta}}. \tag{7.8}$$

The first two terms in eq. (7.7) are the same as the level-1 belief update, whereas the third term grasps the message's context by comparing the selected message against the also-runs and leverages the advantage to further update the belief. Notice that θ^* is unknown to the agent, so q in the expectation does not have an exact solution. Thus, we use $\bar{\theta} + \eta g(m^H(fb))$ as an approximation of θ^*. That is, we calculate level-1 ToM update on the parameters of the value function and take an additional gradient ascent step for level-2 ToM update. In this work, we always initialize scouts' value as uniform across all goals, *i.e*., $\bar{\theta}^0 = [0.25, 0.25, 0.25, 0.25]$.

The difference between robots with level-1 ToM and level-2 ToM is the likelihood function they used to model the user. A level-1 robot assumes the user provides feedback only by thinking about how good the proposals are, whereas a level-2 robot is also aware of the pedagogical perspective of the human user in the collaborative game and accommodating the information of both the literal and the pragmatic meaning of user feedback.

Theoretically, the recursive reasoning between robots and the human user can continue infinitely with unlimited resources or up to a fixed-point of convergence [WWP20]. In this work, we only model the human user as knowing the value update mechanism of scouts with level-1 ToM, a manageable extent of reasoning for human cognitive capability [WVV17], which is also adopted by recent literature (*e.g*., [PCD19]).

The effectiveness of this computational model in the Scout Exploration Game has been verified by the empirical results in the previous section. For other settings, in which task performance has a linear relationship with the value, as depicted by eq. (7.1), the same model can be applied

with minor modifications. For settings involving non-linear value functions, the inner product in eq. (7.1) is to be replaced, as well as the gradient function in eq. (7.5). Still, the core computations in the algorithm, namely the MLE learning of the value function and the level-1/2 ToM integration, remains the same.

7.8.5 Utility-aware explanation generation

We generate explanations to aid the human user in collaborating with the robots by accepting/rejecting specific proposals. Given trajectories produced by the planner, the explainer aims to generate human-like explanations that not only provide sufficient semantic information but also match the human user's syntactic preferences, namely, the explanation utility. Specifically, an explanation is defined by its semantic inputs and a set of syntactic rules. The former is produced by the planner, providing explanations regarding *what*. This includes the current observation, physical state, and belief over the value function. The latter is to provide explanations regarding *how*, *i.e.*, user's explanation utility.

To quantitatively estimate the utility values, after each round, we use a Likert-scale questionnaire on explanation/proposal satisfaction (see fig. 7.5**A**). Answers to these questions reflect the participant's belief regarding how helpful the explanations are for them to understand the game and provide correct guidance to the robot team towards plans that are better suited to the scenarios and their value functions.

Given the satisfactory score, we formulate the overall generation as an Hidden Markov Model (HMM)-based sequential generation process, capable of adopting the temporal dynamics of the human user's explanation utility. More precisely, at each step, we first predefine a set of templates, each of which is accompanied by a combination of attributes, *e.g.*, `isCounterfactual`, `hasTarget`.; these templates provide the basis of an explanation and are filled in according to relevant slots. Next, the explainer determines the optimal syntax that matches the human's syntactic utility based on the satisfactory score.

Of note, one distinguished attribute to highlight is `isRitualized`, stemmed from the term "*ontogenetic ritualization*" in evolutionary anthropology literature. Conventionally, ritualization is referred to the evidence that early infants learn to communicate, especially in a symbolic manner, not based on imitation but rather on an individual learning process [Loc80]. Tomasello and Call [TC97] argue such communicative behavior is a communicative signal that can be formed by two individuals shaping each other's behavior in repeated instances of interaction over time. Similar phenomena have also been observed and investigated on other primates, such as great apes [Tom96]. For example, many individual chimpanzees come to use a stylized "arm-raise" to indicate that they are about to hit the other and thus initiate play [TC97]. In this way, a behavior that was not at first a communicative signal would become one over time. Inspired by this non-verbal behavior, the process of "*ontogenetic ritualization*" can also be formed during human-robot teaming, specifically when understanding and reacting to explanations. Intuitively, human speakers are reluctant to repeat similar messages that they have already conveyed before and would rather deliver a more concise version. To achieve this goal, we explicitly define the "ritualized form" of explanation templates.

7.9 Human experiment details and demographics

Human participants were recruited from University of California, San Diego (UCSD) undergraduate students taking psychology courses and the University of California, Los Angeles (UCLA) Department of Psychology subject pool. All subjects were compensated with course credit for their participation. A total of 167 students completed the introduction phase and passed the familiarization test (56, 53, 58 for the proposal-only , brief-explanation , full-explanation group, respectively). 19 subjects were removed from the analysis for failing the attention check during the game play, resulting in 148 subjects (49, 47, 52 for the proposal-only , brief-explanation , full-explanation group, respectively) considered in the final results. In fig. 7.4, we report results after the removal of outliers that are 1.5 IQR below the 25th percentile or above the 75th percentile,

resulting in 135 valid subjects (45 subjects per group). Before the game starts, all subjects were assigned to one of the three explanation groups and given one of seven value functions randomly. The explanations for the brief-explanation and full-explanation groups are generated as described in the "Utility-aware explanation generation" section. Subjects in the proposal-only group did not have access to any explanations.

The experiment included three phases: introduction, familiarization, and game play. In the introduction phase, participants were presented with the context and rules of the Scout Exploration Game. Icons, scores and UI in the game were explained to the participants with both text descriptions and video demonstrations. Because subjects in different explanation groups will see different UI in the game, we guaranteed that the UI in the video demonstrations is consistent with the one in the actual game; video demonstrations for other groups were not presented, which ensures the between-subject design. In the familiarization stage, participants were tested with multiple-choice questions about their understanding of the game flow, rules, and the UI. Participants who correctly answered all questions proceeded to game play. Participants having at least one wrong answer were asked to review the introduction and retake the familiarization test. Participants who could not pass the familiarization test twice or took more than 20 minutes before starting the game play were removed from the study. Credits were awarded to participants regardless of their familiarization test results. The computational model used for scouts' value alignment is the same across all groups to attribute the difference in the performance of bidirectional value alignment to the lack or distinction of explanations. Participants in the proposal-only , brief-explanation , and full-explanation group communicate with the scouts for 15.4, 16.4, 16.2 rounds on average, with the standard deviation of 3.2, 4.0, 3.7 rounds, respectively. The average time of game play is 20.8, 22.5, 32.3 minutes, with the standard deviation of 4.5, 6.3, 10.7 minutes for the proposal-only , brief-explanation , and full-explanation group, respectively.

To measure the value alignment performance, we use the Kendall τ coefficient to compare the goals' importance ranking in the target value with the ranking in the value estimation. The null hypothesis is that explanations yield the same value alignment across different groups, and

therefore, no difference in the ranking statistics would be observed. The test is a two-tailed independent samples t test to compare performance from two groups of participants, because we used a between-subjects design in the study, with a commonly used significance level $\alpha = 0.05$, assuming t-distribution, and the rejection region is $P \leq 0.05$.

CHAPTER 8

Conclusion

As intelligent machines powered by AI are growing more and more capable, there is an ongoing debate over to what extent they will replace human labor. Some argue that machines will coexist with human workers, while others believe full replacement is inevitable at the end. Whatever the answer may be, as their creators and users, we are responsible for making sure machines are achieving what we expect them to achieve. As Robert Weiner stated [Wie60], "if we use, to achieve our purposes, a mechanical agency with whose operation we cannot interfere effectively ... we had better be quite sure that the purpose put into the machine is the purpose which we really desire." Indeed, it is vital to create machines that are understandable and at the same time understand users' mental states, for the acceptance of AI and maybe for the society in general.

This dissertation addresses the challenge of bidirectional mental state alignment between humans and machines from three perspectives: i) creating standardized simulation environment and benchmark to support data-driven learning approaches (chapter 2 and chapter 3), ii) proposing experiment protocol to better understand human mental states from user study (chapter 4), and iii) integrating the human mental state estimation into task planning and motion planning to generate communications for collaboration (chapter 5, chapter 6 and chapter 7).

Nevertheless, we are still far from solving this challenge. Below I list open problems in each directions. I hope this dissertation can provide valuable insights and inspire future works in human-machine interaction.

- Tasks and benchmarks. Although many recent works have contributed to creating simulation environments and tasks for training and evaluating AI agents in general [CDF17, KMH17b,

PRB18, XSL20, BGK20, GSA20], human-machine interaction systems are evaluated on different tasks created by individual researchers, with few exceptions [CSH19]. We believe that creating standardized benchmark and simulation environments in each domain should be a valuable contribution, since it enables researchers to build on top of existing algorithms for new solutions.

- Computational models for inferring human mental states. Humans have different types of mental states, including trust, attention, perception, emotion, belief, motivation, intention and memory. They are typically not directly observable and have to be inferred from humans' behaviors. There exists works modeling some of these mental states separately [LS04, BTS07, VKZ14, YLF20, FQZ21] in rather constrained settings. However, these mental states can be very much correlated for the same individual, thus calling for a holistic approach.

- Communicative actions. To communicate effeciently with each other, humans often rely on common ground, a set of propositions which everyone assumes to be true. Forming the common ground requires implicit and explicit communication. Moreover, based on common ground, humans take advantage of multiple means of communication, including physical motions, language and non-verbal behaviors, suited to the specific situation to achieve their communication intent. Intelligent machines with various embodiment representations should find the most suitable modality for communication.

REFERENCES

[AC18] Mike Ananny and Kate Crawford. "Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability." *new media & society*, **20**(3):973–989, 2018.

[ACM15] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. "Learning to see by moving." In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[AD08] Luis von Ahn and Laura Dabbish. "Designing games with a purpose." *Communications of the ACM*, 2008.

[AD21] Saurabh Arora and Prashant Doshi. "A survey of inverse reinforcement learning: Challenges, methods and progress." *Artificial Intelligence*, p. 103500, 2021.

[AKL16] J Andreas, D Klein, and S Levine. "Modular multitask reinforcement learning with policy sketches. ArXiv e-prints." *arXiv preprint arXiv:1611.01796*, 2016.

[AN04a] Pieter Abbeel and Andrew Y. Ng. "Apprenticeship learning via inverse reinforcement learning." In *Twenty-first international conference on Machine learning - ICML '04*, p. 1, 2004.

[AN04b] Pieter Abbeel and Andrew Y Ng. "Apprenticeship learning via inverse reinforcement learning." In *Proceedings of International Conference on Machine Learning (ICML)*, 2004.

[ANC19] Sule Anjomshoae, Amro Najjar, Davide Calvaresi, and Kary Främling. "Explainable agents and robots: Results from a systematic literature review." In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1078–1088. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[AS17] Stefano V. Albrecht and Peter Stone. "Reasoning about Hypothetical Agent Behaviours and Their Parameters." In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, p. 547–555, Richland, SC, 2017. International Foundation for Autonomous Agents and Multiagent Systems.

[AWT18] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3674–3683, 2018.

[AWZ20] Arjun Akula, Shuai Wang, and Song-Chun Zhu. "Cocox: Generating conceptual and counterfactual explanations via fault-lines." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 2594–2601, 2020.

[BAR44] WINSTON H. F. BARNES. "The Nature of Explanation." *Nature*, 1944.

[BCC01] Igor R. Belousov, Ryad Chellali, and Gordon J. Clapworthy. "Virtual reality tools for Internet robotics." *Proceedings - IEEE International Conference on Robotics and Automation*, 2001.

[BDM17] André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado van Hasselt, and David Silver. "Successor features for transfer in reinforcement learning." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[BDS18] Andrew Brock, Jeff Donahue, and Karen Simonyan. "Large Scale GAN Training for High Fidelity Natural Image Synthesis." *CoRR*, **abs/1809.11096**, 2018.

[BGK20] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. "ObjectNav Revisited: On Evaluation of Embodied Agents Navigating to Objects." In *arXiv:2006.13171*, 2020.

[BL10] Yvonne Barnard and Frank Lai. "Spotting sheep in Yorkshire: Using eye-tracking for studying situation awareness in a driving simulator." In *Human Factors: A System View of Human, Technology and Organisation. Annual Conference of the Europe Chapter of the Human Factors and Ergonomics Society 2009*, 2010.

[BLT16] Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, et al. "Deepmind lab." *arXiv preprint arXiv:1612.03801*, 2016.

[Bol01] Cheryl Actor Bolstad. "Situation awareness: does it change with age?" In *Proceedings of the human factors and ergonomics society annual meeting*, volume 45, pp. 272–276. SAGE Publications Sage CA: Los Angeles, CA, 2001.

[BPA17] Simon Brodeur, Ethan Perez, Ankesh Anand, Florian Golemo, Luca Celotti, Florian Strub, Jean Rouat, Hugo Larochelle, and Aaron Courville. "Home: A household multimodal environment." *arXiv preprint arXiv:1711.11017*, 2017.

[BPF21] Valts Blukis, Chris Paxton, Dieter Fox, Animesh Garg, and Yoav Artzi. "A persistent spatial semantic representation for high-level natural language instruction execution." *arXiv preprint arXiv:2107.05612*, 2021.

[BPS17] Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. "Emergent complexity via multi-agent competition." *arXiv preprint arXiv:1710.03748*, 2017.

[BSS16] Abhizna Butchibabu, Christopher Sparano-Huiban, Liz Sonenberg, and Julie Shah. "Implicit coordination strategies for effective team communication." *Human factors*, **58**(4):595–610, 2016.

[BTS07] 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.

[Car98] Robyn Carston. "Informativeness, relevance and scalar implicature." *Pragmatics And Beyond New Series*, pp. 179–238, 1998.

[CB16] E Coumans and Y Bai. "Pybullet, a python module for physics simulation for games, robotics and machine learning." *GitHub repository*, 2016.

[CDF17] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. "Matterport3d: Learning from rgb-d data in indoor environments." *arXiv preprint arXiv:1709.06158*, 2017.

[CDF18] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. "Matterport3D: Learning from RGB-D data in indoor environments." *Proceedings - 2017 International Conference on 3D Vision, 3DV 2017*, pp. 667–676, 2018.

[CDS15] Elizabeth Cha, Anca D Dragan, and Siddhartha S Srinivasa. "Perceived robot capability." In *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 541–548. IEEE, 2015.

[CEC95] Marisa Carrasco, Denise L Evert, Irene Chang, and Svetlana M Katz. "The eccentricity effect: Target eccentricity affects performance on conjunction searches." *Perception & psychophysics*, **57**(8):1241–1261, 1995.

[CER21] 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*, pp. 1–11, 2021.

[CGS18] Joyce Y Chai, Qiaozi Gao, Lanbo She, Shaohua Yang, Sari Saba-Sadiya, and Guangyue Xu. "Language to Action: Towards Interactive Task Learning with Physical Agents." In *IJCAI*, pp. 2–9, 2018.

[CH05] Miguel A Carreira-Perpinan and Geoffrey E Hinton. "On contrastive divergence learning." In *Aistats*, volume 10, pp. 33–40. Citeseer, 2005.

[Cla96] Herbert H Clark. *Using language*. Cambridge university press, 1996.

[CNS18] Min Chen, Stefanos Nikolaidis, Harold Soh, David Hsu, and Siddhartha Srinivasa. "Planning with trust for human-robot collaboration." In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 307–315, 2018.

[CSE20] Ta-Chung Chi, Minmin Shen, Mihail Eric, Seokhwan Kim, and Dilek Hakkani-tur. "Just ask: An interactive learning framework for vision and language navigation." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 2459–2466, 2020.

[CSH19] Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. "On the utility of learning about humans for human-ai coordination." *Advances in neural information processing systems*, **32**, 2019.

[CSZ17] Tathagata Chakraborti, Sarath Sreedharan, Yu Zhang, and Subbarao Kambhampati. "Plan Explanations as Model Reconciliation: Moving Beyond Explanation as Soliloquy." In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 156–163, 2017.

[DA16] Sandra Devin and Rachid Alami. "An implemented theory of mind to improve human-robot shared plans execution." In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 319–326. IEEE, 2016.

[DDG18] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. "Embodied Question Answering." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[Den89] Daniel Clement Dennett. *The intentional stance*. MIT press, 1989.

[DHT98] Francis T Durso, Carla A Hackworth, Todd R Truitt, Jerry Crutchfield, Danko Nikolic, and Carol A Manning. "Situation awareness as a predictor of performance for en route air traffic controllers." *Air Traffic Control Quarterly*, **6**(1):1–20, 1998.

[Dia10] Rosen Diankov. *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, USA, 2010.

[DS13] Anca Dragan and Siddhartha Srinivasa. "Generating Legible Motion." In *Proceedings of Robotics: Science and Systems (RSS '13)*, June 2013.

[EB94] Mica R Endsley and Cheryl A Bolstad. "Individual differences in pilot situation awareness." *The International Journal of Aviation Psychology*, **4**(3):241–264, 1994.

[EGL19] Mark Edmonds, Feng Gao, Hangxin Liu, Xu Xie, Siyuan Qi, Brandon Rothrock, Yixin Zhu, Ying Nian Wu, Hongjing Lu, and Song-Chun Zhu. "A tale of two explanations: Enhancing human trust by explaining robot behavior." *Science Robotics*, **4**(37), 2019.

[End88a] Mica R Endsley. "Design and evaluation for situation awareness enhancement." In *Proceedings of the Human Factors Society annual meeting*, volume 32, pp. 97–101. Sage Publications Sage CA: Los Angeles, CA, 1988.

[End88b] Mica R Endsley. "Situation awareness global assessment technique (SAGAT)." In *Proceedings of the IEEE 1988 national aerospace and electronics conference*, pp. 789–795. IEEE, 1988.

[End90] Mica R Endsley. "Predictive utility of an objective measure of situation awareness." In *Proceedings of the Human Factors Society annual meeting*, volume 34, pp. 41–45. SAGE Publications Sage CA: Los Angeles, CA, 1990.

[FKS08] Susan R Fussell, Sara Kiesler, Leslie D Setlock, and Victoria Yew. "How people anthropomorphize robots." In *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 145–152. IEEE, 2008.

[FLH20] Mitsuki Fujino, Jieun Lee, Toshiaki Hirano, Yuichi Saito, and Makoto Itoh. "Comparison of SAGAT and SPAM for Seeking Effective Way to Evaluate Situation Awareness and Workload During Air Traffic Control Task." In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 64, pp. 1836–1840. SAGE Publications Sage CA: Los Angeles, CA, 2020.

[FMB15] Martin L Felis, Katja Mombaur, and Alain Berthoz. "An optimal control approach to reconstruct human gait dynamics from kinematic data." In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 1044–1051. IEEE, 2015.

[FQZ21] Lifeng Fan, Shuwen Qiu, Zilong Zheng, Tao Gao, Song-Chun Zhu, and Yixin Zhu. "Learning triadic belief dynamics in nonverbal communication from videos." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7312–7321, 2021.

[FR13] Alireza Fathi and James M. Rehg. "Modeling actions through state changes." In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013.

[FZZ18] Linxi Fan, Yuke Zhu, Jiren Zhu, Zihua Liu, Orien Zeng, Anchit Gupta, Joan Creus-Costa, Silvio Savarese, and Li Fei-Fei. "SURREAL: Open-Source Reinforcement Learning Framework and Robot Manipulation Benchmark." In *Conference on Robot Learning*, 2018.

[GGC16] A. Giusti, J. Guzzi, D. C. Cireşan, F. He, J. P. Rodríguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. D. Caro, D. Scaramuzza, and L. M. Gambardella. "A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots." *IEEE Robotics and Automation Letters*, **1**(2):661–667, July 2016.

[GGL18] Jianfeng Gao, Michel Galley, and Lihong Li. "Neural approaches to conversational ai." In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 1371–1374, 2018.

[GGS19] Xiaofeng Gao, Ran Gong, Tianmin Shu, Xu Xie, Shu Wang, and Song-Chun Zhu. "VRKitchen: an Interactive 3D Virtual Environment for Task-oriented Learning." *arXiv*, **abs/1903.05757**, 2019.

[GGZ20] Xiaofeng Gao, Ran Gong, Yizhou Zhao, Shu Wang, Tianmin Shu, and Song-Chun Zhu. "Joint mind modeling for explanation generation in complex human-robot collaborative tasks." In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1119–1126. IEEE, 2020.

[GKR18] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. "Iqa: Visual question answering in interactive environments." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4089–4098, 2018.

[GN07] Victoria Groom and Clifford Nass. "Can robots be teammates?: Benchmarks in human–robot teams." *Interaction Studies*, **8**(3):483–500, 2007.

[Gri75] Herbert P Grice. "Logic and conversation." In *Speech acts*, pp. 41–58. Brill, 1975.

[GRO17] Andrea de Giorgio, Mario Romero, Mauro Onori, and Lihui Wang. "Human-machine Collaboration in Virtual Reality for Adaptive Production Engineering." *Procedia Manufacturing*, 2017.

[GS13] Noah D Goodman and Andreas Stuhlmüller. "Knowledge and implicature: Modeling language understanding as social cognition." *Topics in Cognitive Science*, **5**(1):173–184, 2013.

[GSA20] Chuang Gan, Jeremy Schwartz, Seth Alter, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, Megumi Sano, et al. "Threedworld: A platform for interactive multi-modal physical simulation." *arXiv preprint arXiv:2007.04954*, 2020.

[GSS13] Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea Lockerd Thomaz. "Policy Shaping: Integrating Human Feedback with Reinforcement Learning." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

[Gun17] David Gunning. "Explainable artificial intelligence (xai)." *Defense Advanced Research Projects Agency (DARPA), nd Web*, **2**, 2017.

[GZ18] Ze Gong and Yu Zhang. "Behavior explanation as intention signaling in human-robot teaming." In *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 1005–1011. IEEE, 2018.

[HBV14] Kelsey P Hawkins, Shray Bansal, Nam N Vo, and Aaron F Bobick. "Anticipating human actions for collaboration in the presence of task and sensor uncertainty." In *2014 IEEE international conference on robotics and automation (ICRA)*, pp. 2215–2222. IEEE, 2014.

[Hel14] Tove Helldin. *Transparency for Future Semi-Automated Systems: Effects of transparency on operator performance, workload and trust*. PhD thesis, Örebro Universitet, 2014.

[HGD17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. "Mask R-CNN." *CoRR*, **abs/1703.06870**, 2017.

[HHA19] H. Sandy Huang, David Held, Pieter Abbeel, and Anca Dragan. "Enabling Robots to Communicate their Objectives." *Autonomous Robots*, **43**, 02 2019.

[HHB14] Adam Heenan, Chris M Herdman, Matthew S Brown, and Nicole Robert. "Effects of conversation on situation awareness and working memory in simulated driving." *Human factors*, **56**(6):1077–1092, 2014.

[HKB15] Andrei Haidu, Daniel Kohlsdorf, and Michael Beetz. "Learning action failure models from interactive physics-based simulations." In *IEEE International Conference on Intelligent Robots and Systems*, 2015.

[HLM16] Mark K Ho, Michael Littman, James MacGlashan, Fiery Cushman, and Joseph L Austerweil. "Showing versus doing: Teaching by demonstration." *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[HMW20] Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. "A simple language model for task-oriented dialogue." *arXiv preprint arXiv:2005.00796*, 2020.

[HRA16] Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. "Cooperative inverse reinforcement learning." In *Advances in neural information processing systems*, pp. 3909–3917, 2016.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory." *Neural computation*, **9**(8):1735–1780, 1997.

[Hub88] Bernardo A Huberman. "The ecology of computation." In *Digest of Papers. COMPCON Spring 89. Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage*, p. 362. IEEE, 1988.

[HVP17] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Andrew Sendonaris, Gabriel Dulac-Arnold, Ian Osband, John Agapiou, Joel Z. Leibo, and Audrunas Gruslys. "Learning from Demonstrations for Real World Reinforcement Learning." *CoRR*, **abs/1704.03732**, 2017.

[HZR15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition. arXiv 2015." *arXiv preprint arXiv:1512.03385*, 2015.

[ILA15] Phillip Isola, Joseph J. Lim, and Edward H. Adelson. "Discovering states and transformations in image collections." In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[int18] SAE international. "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles." *SAE*, 2018.

[JBD00] 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.

[JH10] Kamilla R Johannsdottir and Chris M Herdman. "The role of working memory in supporting drivers' situation awareness for surrounding traffic." *Human factors*, **52**(6):663–673, 2010.

[JHH16] Matthew Johnson, Katja Hofmann, Tim Hutton, and David Bignell. "The malmo platform for artificial intelligence experimentation." *IJCAI International Joint Conference on Artificial Intelligence*, **2016-Janua**:4246–4247, 2016.

[JLD13] Malte F Jung, Jin Joo Lee, Nick DePalma, Sigurdur O Adalgeirsson, Pamela J Hinds, and Cynthia Breazeal. "Engaging robots: easing complex human-robot teamwork using backchanneling." In *Proceedings of the 2013 conference on Computer supported cooperative work*, pp. 1555–1566, 2013.

[JV19] Matthew Johnson and Alonso Vera. "No AI is an island: the case for teaming intelligence." *AI magazine*, **40**(1):16–28, 2019.

[JWS09] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "ImageNet: A large-scale hierarchical image database." In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[KH12] Alex Krizhevsky and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In *Neural Information Processing Systems*, 2012.

[KHD18] 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*, pp. 87–95, 2018.

[KL11] Leslie Pack Kaelbling and Tomás Lozano-Pérez. "Hierarchical task and motion planning in the now." In *2011 IEEE International Conference on Robotics and Automation*, pp. 1470–1477. IEEE, 2011.

[KM94] Richard Klimoski and Susan Mohammed. "Team mental model: Construct or metaphor?" *Journal of Management*, **20**(2):403–437, 1994.

[KMH17a] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. "AI2-THOR: An Interactive 3D Environment for Visual AI." *arXiv*, 2017.

[KMH17b] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. "Ai2-thor: An interactive 3d environment for visual ai." *arXiv preprint arXiv:1712.05474*, 2017.

[KNM01] H. Kawasaki, K. Nakayama, T. Mouri, and S. Ito. "Virtual teaching based on hand manipulability for multi-fingered robots." *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, 2001.

[KS09] W Bradley Knox and Peter Stone. "Interactively shaping agents via human reinforcement: The TAMER framework." In *Proceedings of the international conference on knowledge capture*, 2009.

[KWG18] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)." In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

[KWR17] Michal Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaskowski. "ViZDoom: A Doom-based AI research platform for visual reinforcement learning." *IEEE Conference on Computatonal Intelligence and Games, CIG*, 2017.

[LBA21] Gregory Lemasurier, Gal Bejerano, Victoria Albanese, Jenna Parrillo, Holly A Yanco, Nicholas Amerson, Rebecca Hetrick, and Elizabeth Phillips. "Methods for Expressing Robot Intent for Human–Robot Collaboration in Shared Workspaces." *ACM Transactions on Human-Robot Interaction (THRI)*, **10**(4):1–27, 2021.

[LDL18] Weiyang Liu, Bo Dai, Xingguo Li, Zhen Liu, James Rehg, and Le Song. "Towards black-box iterative machine teaching." In *Proceedings of International Conference on Machine Learning (ICML)*, 2018.

[LFD16] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. "End-to-end training of deep visuomotor policies." *The Journal of Machine Learning Research*, **17**(1):1334–1373, 2016.

[LFK20] Joshua Lee, Jeffrey Fong, Bing Cai Kok, and Harold Soh. "Getting to Know One Another: Calibrating Intent, Capabilities and Trust for Human-Robot Collaboration." *arXiv preprint arXiv:2008.00699*, 2020.

[LGF16] Adam Lerer, Sam Gross, and Rob Fergus. "Learning Physical Intuition of Block Towers by Example." In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, p. 430–438. JMLR.org, 2016.

[LHF16] Chang Liu, Jessica B Hamrick, Jaime F Fisac, Anca D Dragan, J Karl Hedrick, S Shankar Sastry, and Thomas L Griffiths. "Goal inference improves objective and perceived performance in human-robot collaboration." In *Proceedings of the 2016 international conference on autonomous agents & multiagent systems*, pp. 940–948. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

[LHP05] C Karen Liu, Aaron Hertzmann, and Zoran Popović. "Learning physics-based motion style with nonlinear inverse optimization." *ACM Transactions on Graphics (TOG)*, **24**(3):1071–1081, 2005.

[LHP15] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. "Continuous control with deep reinforcement learning." *arXiv preprint arXiv:1509.02971*, 2015.

[LLR18] Patrick Lindemann, Tae-Young Lee, and Gerhard Rigoll. "Catch my drift: Elevating situation awareness for highly automated driving with an explanatory windshield display user interface." *Multimodal Technologies and Interaction*, **2**(4):71, 2018.

[LMS17] Pat Langley, Ben Meadows, Mohan Sridharan, and Dongkyu Choi. "Explainable agency for intelligent autonomous systems." In *Twenty-Ninth IAAI Conference*, 2017.

[Loc80] A. Lock. *The guided reinvention of language*. Academic Pr, 1980.

[LRM17] Oliver Liu, Daniel Rakita, Bilge Mutlu, and Michael Gleicher. "Understanding human-robot interaction in virtual reality." In *RO-MAN 2017 - 26th IEEE International Symposium on Robot and Human Interactive Communication*, 2017.

[LS04] John D Lee and Katrina A See. "Trust in automation: Designing for appropriate reliance." *Human factors*, **46**(1):50–80, 2004.

[Lu12] Xiaofei Lu. "The relationship of lexical richness to the quality of ESL learners' oral narratives." *The Modern Language Journal*, **96**(2):190–208, 2012.

[LWZ14] Shuang Liu, Xiaoru Wanyan, and Damin Zhuang. "Modeling the situation awareness by the analysis of cognitive process." *Bio-medical materials and engineering*, **24**(6):2311–2318, 2014.

[LWZ17] Yang Liu, Ping Wei, and Song Chun Zhu. "Jointly Recognizing Object Fluents and Tasks in Egocentric Videos." In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-Octob, pp. 2943–2951, 2017.

[LZS18] Hangxin Liu, Yaofang Zhang, Wenwen Si, Xu Xie, Yixin Zhu, and Song-Chun Zhu. "Interactive robot knowledge patching using augmented reality." In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1947–1954. IEEE, 2018.

[MBM16] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. "Asynchronous methods for deep reinforcement learning." In *International conference on machine learning*, pp. 1928–1937. PMLR, 2016.

[MCR21] So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. "FILM: Following Instructions in Language with Modular Methods." *arXiv preprint arXiv:2110.07342*, 2021.

[MES04] Jean MacMillan, Elliot E Entin, and Daniel Serfaty. "Communication overhead: The hidden cost of team cognition." *Team cognition: Understanding the factors that drive process and performance*, 2004.

[MG00] Maria Madsen and Shirley Gregor. "Measuring human-computer trust." In *11th australasian conference on information systems*, volume 53, pp. 6–8. Citeseer, 2000.

[MHL17] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J. Davison. "SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation?" In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-Octob, pp. 2697–2706, 2017.

[Mil19] Tim Miller. "Explanation in artificial intelligence: Insights from the social sciences." *Artificial Intelligence*, **267**:1–38, 2019.

[MKS15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. "Human-level control through deep reinforcement learning." *Nature*, **518**(7540):529–533, feb 2015.

[MKT18] Coleman Merenda, Hyungil Kim, Kyle Tanous, Joseph L Gabbard, Blake Feichtl, Teruhisa Misu, and Chihiro Suga. "Augmented reality interface design approaches for goal-directed and stimulus-driven driving tasks." *IEEE transactions on visualization and computer graphics*, **24**(11):2875–2885, 2018.

[MSW16] Nikola Mrkšić, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. "Neural belief tracker: Data-driven dialogue state tracking." *arXiv preprint arXiv:1606.03777*, 2016.

[MU21] Bertram F Malle and Daniel Ullman. "A multidimensional conception and measure of human-robot trust." In *Trust in Human-Robot Interaction*, pp. 3–25. Elsevier, 2021.

[MV53] Oskar Morgenstern and John Von Neumann. *Theory of games and economic behavior*. Princeton university press, 1953.

[ND19] Khanh Nguyen and Hal Daumé III. "Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning." *arXiv preprint arXiv:1909.01871*, 2019.

[NDB19] Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. "Vision-based navigation with language-based assistance via imitation learning with indirect intervention." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12527–12537, 2019.

[NKF18] Stefanos Nikolaidis, Minae Kwon, Jodi Forlizzi, and Siddhartha Srinivasa. "Planning with verbal communication for human-robot collaboration." *ACM Transactions on Human-Robot Interaction (THRI)*, **7**(3):1–21, 2018.

[NL80] Walter W Nelson and Geoffrey R Loftus. "The functional visual field during picture viewing." *Journal of Experimental Psychology: Human Learning and Memory*, **6**(4):391, 1980.

[NM01] Monica N Nicolescu and Maja J Mataric. "Learning and interacting in human-robot domains." *IEEE Transactions on Systems, man, and Cybernetics-part A: Systems and Humans*, **31**(5):419–430, 2001.

[NNP17] 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*, pp. 323–331, 2017.

[NR00] Andrew Y. Ng and Stuart Russell. "Algorithms for inverse reinforcement learning." In *International Conference on Machine Learning (ICML)*, 2000.

[PAR18] Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. "Multi-goal reinforcement learning: Challenging robotics environments and request for research." *arXiv preprint arXiv:1802.09464*, 2018.

[PCD19] Tomi Peltola, Mustafa Mert Çelikok, Pedram Daee, and Samuel Kaski. "Machine Teaching of Active Sequential Learners." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[PK06] Aaron Powers and Sara Kiesler. "The advisor robot: tracing people's mental model from a robot's physical attributes." In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pp. 218–225, 2006.

[PLL17] Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. "Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning." *arXiv preprint arXiv:1704.03084*, 2017.

[PRB18] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. "Virtualhome: Simulating household activities via programs." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8494–8502, 2018.

[PSS21] Alexander Pashevich, Cordelia Schmid, and Chen Sun. "Episodic Transformer for Vision-and-Language Navigation." *arXiv preprint arXiv:2105.06453*, 2021.

[PTF16] Minh Tien Phan, Indira Thouvenin, and Vincent Frémont. "Enhancing the driver awareness of pedestrian using augmented reality cues." In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1298–1304. IEEE, 2016.

[PTS21] Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramithu, Gokhan Tur, and Dilek Hakkani-Tur. "TEACh: Task-driven Embodied Agents that Chat." *arXiv preprint arXiv:2110.00534*, 2021.

[PW78] David Premack and Guy Woodruff. "Does the chimpanzee have a theory of mind?" *Behavioral and brain sciences*, **1**(4):515–526, 1978.

[QCF19] Nicola Quinn, Lajos Csincsik, Erin Flynn, Christine A Curcio, Szilard Kiss, Srini-Vas R Sadda, Ruth Hogg, Tunde Peto, and Imre Lengyel. "The clinical relevance of visualising the peripheral retina." *Progress in retinal and eye research*, **68**:83–109, 2019.

[QY16] Weichao Qiu and Alan Yuille. "UnrealCV: Connecting computer vision to unreal engine.", 2016.

[RAA12] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. "A database for fine grained activity detection of cooking activities." In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1194–1201, 2012.

[RBT20] Homero Roman Roman, Yonatan Bisk, Jesse Thomason, Asli Celikyilmaz, and Jianfeng Gao. "Rmm: A recursive mental model for dialog navigation." *arXiv preprint arXiv:2005.00728*, 2020.

[RC07] Keith Rayner and Monica Castelhano. "Eye movements." *Scholarpedia*, **2**(10):3649, 2007.

[RCS92] William B Rouse, Janis A Cannon-Bowers, and Eduardo Salas. "The role of mental models in team performance in complex systems." *IEEE transactions on systems, man, and cybernetics*, **22**(6):1296–1308, 1992.

[RDL18] Sid Reddy, Anca Dragan, and Sergey Levine. "Where do you think you're going?: Inferring beliefs about dynamics from behavior." In *Advances in Neural Information Processing Systems*, pp. 1454–1465, 2018.

[RGB10] Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. "A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning." In *Proceedings of AISTATS*, volume 15, pp. 627–635, 2010.

[RLG18] Anselm Rothe, Brenden M Lake, and Todd M Gureckis. "Do people ask good questions?" *Computational Brain & Behavior*, **1**(1):69–89, 2018.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining*, 2016.

[RSP21] Minjin Rheu, Ji Youn Shin, Wei Peng, and Jina Huh-Yoo. "Systematic review: trust-building factors and implications for conversational agent design." *International Journal of Human–Computer Interaction*, **37**(1):81–96, 2021.

[Rus19] Stuart Russell. *Human compatible: Artificial intelligence and the problem of control*. Penguin, 2019.

[RVB21] Frederic Anthony Robinson, Mari Velonaki, and Oliver Bown. "Smooth Operator: Tuning Robot Perception Through Artificial Movement Sound." In *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 53–62, 2021.

[Sam38] Paul A Samuelson. "A note on the pure theory of consumer's behaviour." *Economica*, **5**(17):61–71, 1938.

[SB18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[SCD17] Manolis Savva, Angel X Chang, Alexey Dosovitskiy, Thomas Funkhouser, and Vladlen Koltun. "MINOS: Multimodal indoor simulator for navigation in complex environments." *arXiv preprint arXiv:1712.03931*, 2017.

[Sch92] Shalom H Schwartz. "Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries." In *Advances in experimental social psychology*, volume 25, pp. 1–65. Elsevier, 1992.

[SDL18] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. "Airsim: High-fidelity visual and physical simulation for autonomous vehicles." In *Field and service robotics*, pp. 621–635. Springer, 2018.

[SGB15] Freek Stulp, Jonathan Grizou, Baptiste Busch, and Manuel Lopes. "Facilitating intention prediction for humans by optimizing robot motions." In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 1249–1255. IEEE, 2015.

[SGF13] Nathaniel J Smith, Noah Goodman, and Michael Frank. "Learning and using language via recursive pragmatic reasoning about other agents." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

[SGG14] Patrick Shafto, Noah D Goodman, and Thomas L Griffiths. "A rational account of pedagogical reasoning: Teaching by, and learning from, examples." *Cognitive psychology*, **71**:55–89, 2014.

[SGM16] Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. "On-line active reward learning for policy optimisation in spoken dialogue systems." *arXiv preprint arXiv:1605.07669*, 2016.

[SGR17] Tianmin Shu, Xiaofeng Gao, Michael S Ryoo, and Song-Chun Zhu. "Learning social affordance grammar from videos: Transferring human interactions to human-robot interactions." In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 1669–1676. IEEE, 2017.

[SGS99] Thomas A Stoffregen, Kathleen M Gorday, Yang-Yi Sheng, and Steven B Flynn. "Perceiving affordances for another person's actions." *Journal of Experimental Psychology: Human Perception and Performance*, **25**(1):120, 1999.

[SHL13] John Schulman, Jonathan Ho, Alex X Lee, Ibrahim Awwal, Henry Bradlow, and Pieter Abbeel. "Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization." In *Robotics: science and systems*, volume 9, pp. 1–10. Citeseer, 2013.

[Sim07] Jeffry A Simpson. "Psychological foundations of trust." *Current directions in psychological science*, **16**(5):264–268, 2007.

[SKK10] Peter Stone, Gal A Kaminka, Sarit Kraus, Jeffrey S Rosenschein, et al. "Ad Hoc Autonomous Agent Teams: Collaboration without Pre-Coordination." In *AAAI*, p. 6, 2010.

[SMF14] Daniel Szafir, Bilge Mutlu, and Terrence Fong. "Communication of intent in assistive free flyers." In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pp. 358–365, 2014.

[SMJ17] David Sirkin, Nikolas Martelaro, Mishel Johns, and Wendy Ju. "Toward measurement of situation awareness in autonomous vehicles." In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 405–415, 2017.

[SP02] Martin Stolle and Doina Precup. "Learning Options in Reinforcement Learning." In *Proceedings of the 5th International Symposium on Abstraction, Reformulation and Approximation*, pp. 212–223, London, UK, UK, 2002. Springer-Verlag.

[SRK19] Oliver Struckmeier, Mattia Racca, and Ville Kyrki. "Autonomous Generation of Robust and Focused Explanations for Robot Policies." In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1–8. IEEE, 2019.

[SS08] Umar Syed and Robert E Schapire. "A Game-Theoretic Approach to Apprenticeship Learning." In *Advances in Neural Information Processing Systems 20*, volume 20, pp. 1–8, 2008.

[SSK18] Sarath Sreedharan, Siddharth Srivastava, and Subbarao Kambhampati. "Hierarchical Expertise Level Modeling for User Specific Contrastive Explanations." In *IJCAI*, pp. 4829–4836, 2018.

[SSS17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. "Mastering the game of Go without human knowledge." *Nature*, **550**(7676):354–359, oct 2017.

[SSW06] Paul Salmon, Neville Stanton, Guy Walker, and Damian Green. "Situation awareness measurement: A review of applicability for C4i environments." *Applied ergonomics*, **37**(2):225–238, 2006.

[STG20] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. "Alfred: A benchmark for interpreting grounded instructions for everyday tasks." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10740–10749, 2020.

[SV10] David Silver and Joel Veness. "Monte-Carlo planning in large POMDPs." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2010.

[SWD17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms." *arXiv preprint arXiv:1707.06347*, 2017.

[SXD14] Liu Shuang, Wanyan Xiaoru, and Zhuang Damin. "A quantitative situational awareness model of pilot." In *Proceedings of the International Symposium on Human Factors and Ergonomics in Health Care*, volume 3, pp. 117–122. SAGE Publications Sage CA: Los Angeles, CA, 2014.

[SXR15] Tianmin Shu, Dan Xie, Brandon Rothrock, Sinisa Todorovic, and Song-Chun Zhu. "Joint inference of groups, events and human roles in aerial videos." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4576–4584, 2015.

[SXS18] Tianmin Shu, Caiming Xiong, and Richard Socher. "Hierarchical and Interpretable Skill Acquisition in Multi-task Reinforcement Learning." In *6th International Conference on Learning Representations (ICLR)*, 2018.

[SYZ17] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas Funkhouser. "Semantic scene completion from a single depth image." In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pp. 190–198, 2017.

[TAH19] 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)*, pp. 249–257. IEEE, 2019.

[Tay90] RM Taylor. "Situational Awareness Rating Technique (SART): The development of a tool for aircrew systems design. Situational Awareness in Aerospace Operations (AGARD-CP-478)." *Neuilly Sur Seine, France: NATO-AGARD*, 1990.

[TC97] Michael Tomasello and Josep Call. *Primate cognition*. Oxford University Press, 1997.

[TDJ11] Leila Takayama, Doug Dooley, and Wendy Ju. "Expressing thought: improving robot readability with animation principles." In *Proceedings of the 6th international conference on Human-robot interaction*, pp. 69–76, 2011.

[TET12] Emanuel Todorov, Tom Erez, and Yuval Tassa. "MuJoCo: A physics engine for model-based control." In *IEEE International Conference on Intelligent Robots and Systems*, 2012.

[The] The MakeHuman team. "MakeHuman.".

[TJ19] Yourui Tong and Bochen Jia. "An Augmented-reality-based Warning Interface for Pedestrians: User Interface Design and Evaluation." In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 63, pp. 1834–1838. SAGE Publications Sage CA: Los Angeles, CA, 2019.

[TKL14] Stefanie Tellex, Ross A. Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. "Asking for Help Using Inverse Semantics." In Dieter Fox, Lydia E. Kavraki, and Hanna Kurniawati, editors, *Robotics: Science and Systems X, University of California, Berkeley, USA, July 12-16, 2014*, 2014.

[TMC20] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. "Vision-and-dialog navigation." In *Conference on Robot Learning*, pp. 394–406. PMLR, 2020.

[Tom96] Michael Tomasello. "Do apes ape." *Social learning in animals: The roots of culture*, pp. 319–346, 1996.

[TPZ13] Kewei Tu, Maria Pavlovskaia, and Song-Chun Zhu. "Unsupervised structure learning of stochastic and-or grammars." In *Advances in neural information processing systems*, pp. 1322–1330, 2013.

[ULS20] Vaibhav V Unhelkar, Shen Li, and Julie A Shah. "Decision-making for bidirectional communication in sequential human-robot collaborative tasks." In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 329–341, 2020.

[VBP13] Adam Vogel, Max Bodoia, Christopher Potts, and Dan Jurafsky. "Emergence of Gricean maxims from multi-agent decision theory." In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2013.

[VK19] Gentiane Venture and Dana Kulić. "Robot expressive motions: a survey of generation and evaluation methods." *ACM Transactions on Human-Robot Interaction (THRI)*, **8**(4):1–17, 2019.

[VKZ14] Gentiane Venture, Hideki Kadone, Tianxiang Zhang, Julie Grèzes, Alain Berthoz, and Halim Hicheur. "Recognizing emotions conveyed by human gait." *International Journal of Social Robotics*, **6**(4):621–632, 2014.

[VPR13] Carl Vondrick, Donald Patterson, and Deva Ramanan. "Efficiently scaling up crowdsourced video annotation." *International journal of computer vision*, **101**(1):184–204, 2013.

[WH94] Wayne L Waag and Michael R Houck. "Tools for assessing situational awareness in an operational fighter environment." *Aviation, space, and environmental medicine*, 1994.

[Wic02] Christopher D Wickens. "Situation awareness and workload in aviation." *Current directions in psychological science*, **11**(4):128–133, 2002.

[Wie60] Norbert Wiener. "Some Moral and Technical Consequences of Automation: As machines learn they may develop unforeseen strategies at rates that baffle their programmers." *Science*, **131**(3410):1355–1358, 1960.

[WMA08] Christopher D Wickens, Jason S McCarley, Amy L Alexander, Lisa C Thomas, Michael Ambinder, and Sam Zheng. "Attention-situation awareness (A-SA) model of pilot error." *Human performance modeling in aviation*, pp. 213–239, 2008.

[WPH16] 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*, pp. 109–116. IEEE Press, 2016.

[WSB16] Timothy J Wright, Siby Samuel, Avinoam Borowsky, Shlomo Zilberstein, and Donald L Fisher. "Experienced drivers are quicker to achieve situation awareness than inexperienced drivers in situations of transfer of control within a Level 3 autonomous environment." In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 60, pp. 270–273. Sage Publications Sage CA: Los Angeles, CA, 2016.

[WVV17] Harmen de Weerd, Rineke Verbrugge, and Bart Verheij. "Negotiating with other minds: the role of recursive theory of mind in negotiation with incomplete information." *Autonomous Agents and Multi-Agent Systems*, **31**(2):250–287, 2017.

[WWG18] Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. "Building generalizable agents with a realistic and rich 3d environment." *arXiv preprint arXiv:1801.02209*, 2018.

[WWH20] Jianmin Wang, Wenjuan Wang, Preben Hansen, Yang Li, and Fang You. "The Situation Awareness and Usability Research of Different HUD HMI Design in Driving While Using Adaptive Cruise Control." In *International Conference on Human-Computer Interaction*, pp. 236–248. Springer, 2020.

[WWP20] Pei Wang, Junqi Wang, Pushpi Paranamana, and Patrick Shafto. "A mathematical theory of cooperative communication." *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[XBO19] Yaqi Xie, Indu P Bodala, Desmond C Ong, David Hsu, and Harold Soh. "Robot capability and intention in trust-based decisions across tasks." In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 39–47. IEEE, 2019.

[XD15] Anqi Xu and Gregory Dudek. "Optimo: Online probabilistic trust inference model for asymmetric human-robot collaborations." In *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 221–228. IEEE, 2015.

[XSL20] Fei Xia, William B Shen, Chengshu Li, Priya Kasimbeg, Micael Edmond Tchapmi, Alexander Toshev, Roberto Martín-Martín, and Silvio Savarese. "Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments." *IEEE Robotics and Automation Letters*, **5**(2):713–720, 2020.

[XSX16] Caiming Xiong, Nishant Shukla, Wenlong Xiong, and Song-Chun Zhu. "Robot learning with a spatial, temporal, and causal and-or graph." In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2144–2151. IEEE, 2016.

[XZH18] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. "Gibson Env: Real-World Perception for Embodied Agents." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[YKD18] Yucheng Yang, Burak Karakaya, Giancarlo Caccia Dominioni, Kyosuke Kawabe, and Klaus Bengler. "An hmi concept to improve driver's visual behavior and situation awareness in automated vehicle." In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 650–655. IEEE, 2018.

[YLF20] Tao Yuan, Hangxin Liu, Lifeng Fan, Zilong Zheng, Tao Gao, Yixin Zhu, and Song-Chun Zhu. "Joint inference of states, robot knowledge, and human (false-) beliefs." In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5972–5978. IEEE, 2020.

[YZS21] Luyao Yuan, Dongruo Zhou, Junhong Shen, Jingdong Gao, Jeffrey Chen, Quanquan Gu, Ying n Nian Wu, and Song-Chun Zhu. "Iterative Teacher-Aware Learning." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[ZGK17] Yuke Zhu, Daniel Gordon, Eric Kolve, Dieter Fox, Li Fei-Fei, Abhinav Gupta, Roozbeh Mottaghi, and Ali Farhadi. "Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning." *Proceedings of the IEEE International Conference on Computer Vision*, **2017-Octob**(1):483–492, 2017.

[ZMB08] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. "Maximum Entropy Inverse Reinforcement Learning." In *Proc. AAAI*, pp. 1433–1438, 2008.

[ZMM21] Haibei Zhu, Teruhisa Misu, Sujitha Martin, Xingwei Wu, and Kumar Akash. "Improving Driver Situation Awareness Prediction using Human Visual Sensory and Memory Mechanism." *arXiv preprint arXiv:2111.00087*, 2021.

[ZSK17] Yu Zhang, Sarath Sreedharan, Anagha Kulkarni, Tathagata Chakraborti, Hankz Hankui Zhuo, and Subbarao Kambhampati. "Plan explicability and predictability for robot task planning." In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 1313–1320. IEEE, 2017.

[ZWW20] Quanshi Zhang, Xin Wang, Ying Nian Wu, Huilin Zhou, and Song-Chun Zhu. "Interpretable CNNs for object classification." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **43**(10):3416–3431, 2020.

[ZZZ20] Zhenliang Zhang, Yixin Zhu, and Song-Chun Zhu. "Graph-based Hierarchical Knowledge Representation for Robot Task Transfer from Virtual to Physical World."

In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2020.