Human-Agent Teaming with Implicit Guidance

by

Ryo NAKAHASHI

Dissertation

submitted to the Department of Informatics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

The Graduate University for Advanced Studies, SOKENDAI
March 2022

A dissertation submitted to Department of Informatics,
School of Multidisciplinary Sciences,
The Graduate University for Advanced Studies,
SOKENDAI,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Advisory Committee

1. Prof. Seiji Yamada
   National Institute of Informatics
   and the Graduate University for Advanced Studies

2. Assoc.Prof. Tetsunari Inamura
   National Institute of Informatics
   and the Graduate University for Advanced Studies

3. Assoc.Prof. Ryutaro Ichise
   National Institute of Informatics
   and the Graduate University for Advanced Studies

4. Prof. Atsuhiro Takasu
   National Institute of Informatics
   and the Graduate University for Advanced Studies

5. Prof. Sachiyo Arai
   Chiba University

Acknowledgements

I would like to express my deepest appreciation to my advisor Professor Seiji Yamadafor his guidance, patience, and profound vision regarding the research. His continuous support is essential for my research.

I would thank Dr Kazuo Okamura, Takato Okudo, Takahiro Tsumura, Shin Sano, Nungduk Yun, Chinrin Kou, Jingbo Yan, and the other members of Yamada Laboratory. I greatly appreciate their friendship as well as the good advice and comments on my research. Finally, very special thanks are dedicated to my family for all of the understanding and support of the research.

The Graduate University for Advanced Studies, SOKENDAI

Abstract

School of Multidisciplinary Sciences
Department of Informatics

Doctor of Philosophy

Human-Agent Teaming with Implicit Guidance

by Ryo Nakahashi

In the AI research history, the development of autonomous agents which can collaborate naturally with humans is one of the ultimate goals and has been a significant issue. Although there are many types of collaboration with humans and autonomous agents, we focused on one of the in which humans and autonomous agents collaborate to achieve one task. We call the collaboration as "Human-Agent Teaming".

The most straightforward approach for Human-Agent Teaming is that agents concentrate on supporting humans. In this approach, agents infer human's goals or intentions and take action that preferable action to make humans archive their goal. However, these agents cannot modify the human's plan, which means the ultimate success or failure of the collaborative task depends on the human's ability to plan. In other words, if the human sets the wrong plan, the performance will suffer. Furthermore, humans generally have bounded rationality from their cognitive and computational limitations; thus, it is difficult to make optimal plans and excepts efficiently easy tasks. From this, the performance of such an agent can be limited.

The solution is this problem is that agents who do not have cognitive and computational limitations make optimal plans and guide human behavior to follow the plans. A most naive approach is that the agent explicitly guides the action to humans. However, if agents abuse such explicit guidance, humans may lose their autonomy which is humans' sense of control regarding their decision-making in achieving the task. As a result, humans may think that the agent is controlling them. Such an impression makes the Human-Agent Teaming is undesirable.

Thus, agents should guide humans while enabling them to maintain autonomy. We focus on "Implicit Guidance" offered through behavior. The agent will expect the human to infer its intentions and discard any plans that do not match what they infer the agent is planning. Under this expectation, the agent acts in a way that makes it easy for the human to find the best (or at least better) plan for optimum performance on a collaborative task. Implicit guidance of this nature should help humans maintain autonomy since the discarding of plans is a proactive action.

This dissertation proposes the methodology for autonomous agents who can use implicit guidance for Human-Agent Teaming and contains three studies.

The first study is the basic framework to implement collaborative agents based on implicit guidance and show the advantage of the agent. This framework extends the existing planning by implementing the ability to consider the Theory of Mind function to agents. Theory of Mind is human's cognitive function to infer others goals or intentions from others behavior. Considering this function, agents can control human inference for the agent intention to guide better plans. We conducted a participant experiment in which we asked the participants to achieve the simple synthetic task collaborating with several kinds of autonomous agents, including the agent with implicit guidance. We confirmed that the agent with implicit guidance realizes that balancing the performance of the task and keeping human autonomy.

The second and third studies extend the framework for more realistic problems. The second study introduces the "Plan Predictable Bias" into the existing Theory of Mind modeling. That is a kind of "Bounding rationality" of human cognition and the bias that humans tend to infer others' intentions to make easy for their inference. We conducted a participant experiment in which we asked the participants to infer agents' intentions from their behavior in the complex synthetic task. We confirmed that the Theory of Mind model with Plan Predictable Bias matches human cognition better than the existing Theory of Mind.

The third studies extend the planning algorithm for the more realistic situation that the human has specific information about the rewards, which is unknown by the agent. In this case, the agent can not make the best plan at first. Thus the agent has to infer the specific information for the human from the behavior of humans. We implement our implicit guidance idea to the existing collaborative planning algorithm, which expects the humans to show their intention and infer

it. We conducted a participant experiment to evaluate the advantage of extended agents for implicit guidance. We conducted a participant experiment in which we asked the participants to achieve the complex task collaborating with several kinds of autonomous agents, including our extended agent. We confirmed that our framework with our extended agents improved the performance to achieve the collaborative task.

Despite several limitations, this dissertation contributes to making a more prosperous and natural relationship between humans and artificial intelligence.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter gives an overview and composition of this dissertation. Section 1.1 is the overview. Section 1.2 is a brief list of the studies in the dissertation. Section 1.3 gives the structure of the dissertation.

1.1 Introduction

When humans try to work on tasks too complex and challenging to achieve by themselves, they collaborate with others. By doing so, they can achieve more complex tasks efficiently. Achieving tasks with others through collaboration is one ability that characterizes humans as social animals.

In past years, the role of machines and AI was to carry out orders from humans exactly. However, nowadays, autonomous AI agents who think and decide what they should do are an important research topic. Moreover, the ability to collaborate with humans is essential for future human-like AI agents in living together with humans. Therefore, the research topic of human-agent teaming, in which humans and AI agents work together to achieve one task, has become increasingly important.

The simplest approach for human-agent teaming is when agents concentrate on supporting humans [1, 2]. First, the agent tries to understand what a human wants to do through user instructions, operations, behavior, etc. Then, the agent calculates the action that is most preferable to help the human achieve their goal through inference. By taking action, it tries to support the human in achieving their goal. We call this type of agent a supportive agent. However, this agent has a big limitation; it does not have a function for changing a human's goal or intention. This means that even if the human has a wrong goal or inefficient plan for a goal, the agent cannot modify it. In other words, the upper limit of the degree of success of a human-agent team is bounded by the human's ability. This limitation does not matter when the problem size is small enough. However, for a problem that has above a certain size or complexity, a human cannot choose the optimal action since a human has "bounded rationality" [3] due to cognitive and computational limitations. Thus, there is the limitation on the performance for human-agent teaming featuring a supportive agent.

Implementing an agent with a function for modifying a human's goal or intention would be valid for overcoming this limitation. Because agents have far more computational resources than humans, the rationality of an agent is more excellent than that of a human. The most naive approach is where an agent guides a human toward a preferable action or goal directly. We call such guidance explicit guidance. However, explicit guidance should not be abused. If an agent abuses explicit guidance, most of the human's actions would be guided by the agent. As a result, humans may lose their sense of control regarding their decision-making in achieving human-agent teaming —in other words, their autonomy. Furthermore, they might get the impression that the agent is controlling them. Such an impression is undesirable for the wholesome coexistence of humans and agents.

To reduce that risk, agents have to guide humans while allowing them to maintain autonomy. We assume that a critical factor in maintaining autonomy is for humans to decide their own actions in human-agent teaming. Moreover, we also assume that humans infer others' goals or intentions and take action in accordance with their rationality. Under this assumption, an agent will expect the human to infer its intentions and discard any plans that do not match what they infer the agent to be planning. Furthermore, under this expectation, the agent will try to influence the situation so that the human's decision is closer to the optimal plan. Such

action-decision processes maintain human autonomy since the discarding of plans is a proactive action. We call such agent actions implicit guidance.

This dissertation proposes a methodology for autonomous agents who can use implicit guidance for human-agent teaming. The basic approach is that these agents plan their actions under the assumption that humans infer their goals or intentions from their behavior. We use the idea of "theory of mind" [4] to implement this. Theory of mind refers to the capacity to understand other people by ascribing mental states to them. The mental states include meaning, such as beliefs, desires, intentions, emotions, and thoughts. In recent years, there have been studies done to model humans' theory of mind computationally. Using the result of these studies, we model an agent who has the function of theory of mind as a model of a human. Then, we model human-agent teaming as a planning problem that acts with the agent modeled as a human. When solving a problem, the agent has the best policy for a policy under the premise of collaborating with an agent that has theory of mind. In addition, we explain its advantage from the perspective of balancing human autonomy and improving task performance.

1.2 Overview of Studies

This dissertation comprises three studies. The first study regards the basic framework for implementing collaborative agents based on implicit guidance. The second and third studies extend the framework for more realistic problems.

- The first study regards the basic framework for implementing collaborative agents based on implicit guidance and shows the advantage of such agents. This framework extends the existing planning by implementing the ability to consider a theory of mind function for agents. With this function, agents can control human inference of the intention of the agents to guide humans toward better plans. We evaluated the advantage of human-agent teaming with implicit guidance. We conducted a participant experiment in which we asked participants to complete a simple synthetic task by collaborating with several kinds of autonomous agents, including an agent with implicit guidance. We confirmed that the agent with implicit guidance contributed to balancing the performance of the task and maintaining human autonomy.

- The second study is an extension of theory of mind modeling for complex situations. We introduce "plan predictable bias" into the existing theory of mind modeling. It is a kind of "bounded rationality" for human cognition and a bias in which humans tend to infer others' intentions in a way that makes their own inference easier. This bias can be a kind of egocentric bias. We conducted a participant experiment in which we asked participants to infer agents' intentions from their behavior in a complex synthetic task. We confirmed that the theory of mind model with plan predictable bias matches human cognition better than the existing theory of mind model.

- The third study is an extension of a collaborative planning algorithm for more realistic situations in which a human has specific information on rewards that is unknown by the agent. In the first study, the human and the agent share complete reward information. Furthermore, although humans infer the agent's intention, they act in an egocentric rational manner during inference. However, in a realistic problem, a human might have their own goal or preference toward an action that is hidden from the agent. In this case, the agent cannot make the best plan at first. Thus, the agent has to infer this information specific to the human from the behavior of the human. Moreover, we assume humans may share their information such as through implicit guidance. To support this, we implement our human-agent teaming idea in the existing collaborative planning algorithm, which expects humans to show their intention and for agents to infer it. Furthermore, by implementing our extended theory of mind model in our framework, we improve the planning performance. We conducted a participant experiment to evaluate the advantage of extended agents for implicit guidance. We asked the participants to complete a complex task by collaborating with several kinds of autonomous agents, including our extended agent. We confirmed that our framework with our extended agents improved the performance on the collaborative task.

1.3 Composition of Thesis

This thesis is structured as follows

- Chapter 1: The introduction, motivation, and overview of this dissertation.

- Chapter 2: Review of the related work on existing human-agent collaborate algorithms, human cognitive science, and agent planning.
- Chapter 3: Review of the existing technical topics that are the basis of our thesis.
- Chapter 4: Formalization of human-agent teaming problems.
- Chapter 5: Basic framework and planning algorithm for collaborative agents based on implicit guidance.
- Chapter 6: Extension of the framework for more complex situations.
- Chapter 7: A general discussion for our dissertation.
- Chapter 8: A concluding summary of the thesis.

Chapter 2

Related Work

This chapter shows related work for this dissertation.

2.1 Components of Human-Agent Teaming

Human-agent teaming is a problem in which humans and agents collaborate to try and solve the same problem. There are studies in specific domains such as for human-machine teams [5] and coactive design [6]. For example, in studies on human-machine teams, the important components are communication, coordination, and adaptability [5]. Communication means exchanging information between humans and agents, coordination is the process through which humans and agents manage their roles, and adaptability is the ability for humans and agents to adapt to each other. Furthermore, coactive design [6] defines three functions that agents need. Observability is a function for determining their status, intentions, etc., predictability is a function so that a human can predict an agent's action and rely on it when considering their own action, and directability is a function for being able to direct the behavior of a human as well as be directed by a human. Figure 2.1 is the architecture of an agent based on coactive design. The robot and human are connected by a bridge of observability, predictability, and directability.

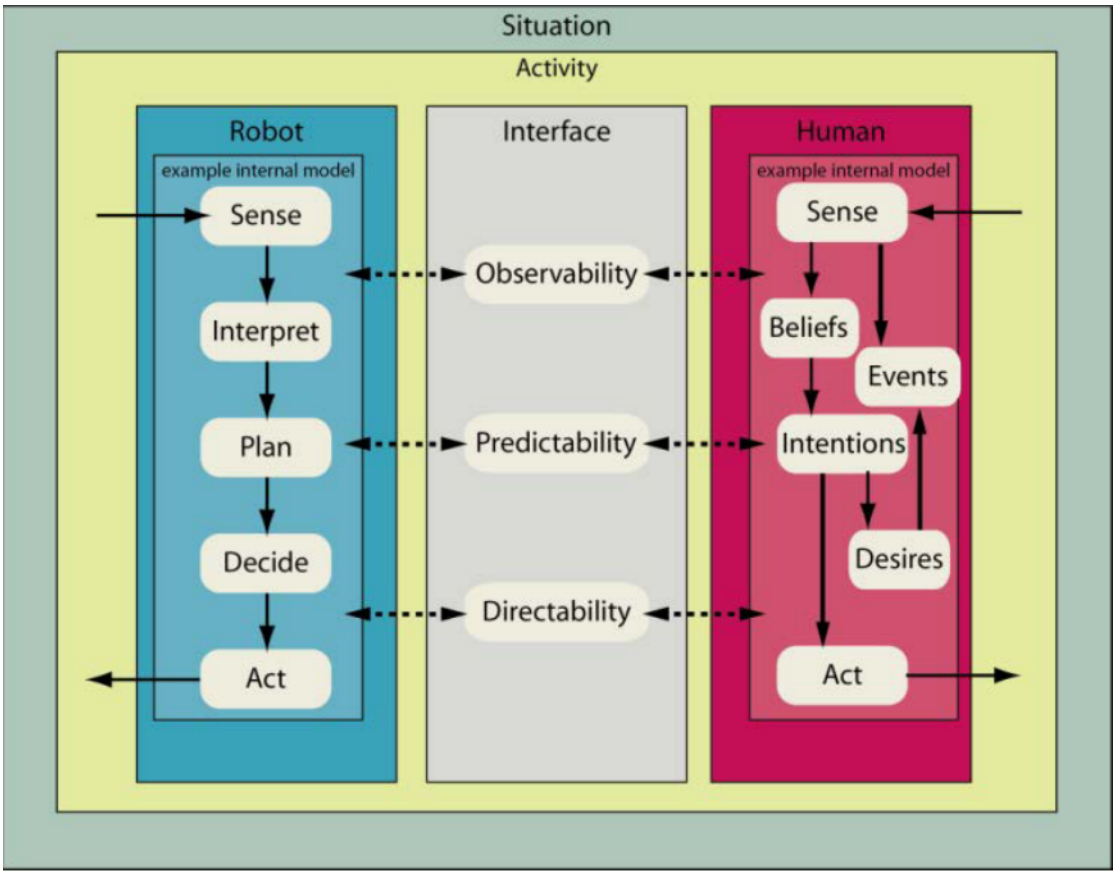

FIGURE 2.1: Architecture of agent based on coactive design

2.2 Multi-Agent Planning / Collaborative Planning

The basis of the agent for human-agent teaming is the ability for them to plan their behavior in an environment in which others exist. A system that has multiple autonomous agents is a multi-agent system [7], and in the human-agent teaming problem in particular, the system can deal with heterogeneous agents of humans and agents. In general, there are two types of solutions to multi-agent problems. One is centralized algorithms, which assume a manager who operates all agents together and considers their planning. The other is distributed algorithms that deal with the strategies of each agent separately. For a human-agent team, only the latter algorithm is available because it is not possible to completely control human behavior.

There are many studies on distributed algorithms. For example, one algorithm includes agents that plan by inferring human subgoals for a partitioned problem

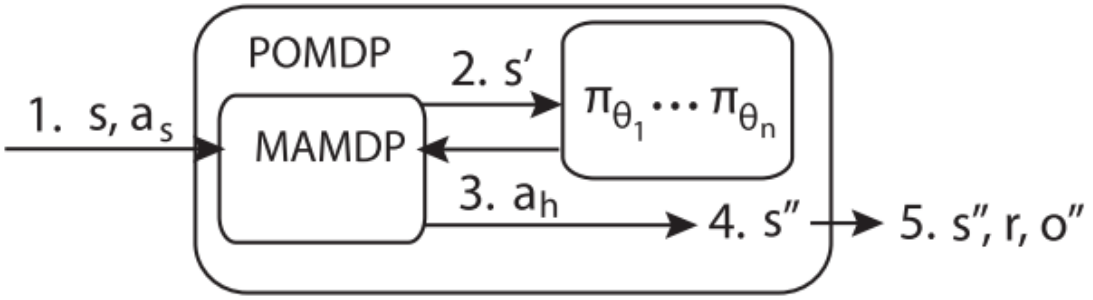

FIGURE 2.2: Example of architecture of POMDP-based collaborative agent

on the basis of Bayesian theory of mind [8]. These agents infer goals from the movements of human agents and then move accordingly (supportive agents in our context). There are also agents that plan while using an information regularizer to control whether to communicate or hide their intention [9] and that plan while using the information of empowerment [10] to increase a human's controllability. These agents are not explicitly aware of their cooperative partner, and the focus is on how to generate biased behavior for generic cooperation.

In addition, there is an approach that considers human behavior explicitly. Most studies solve the single agent problem in an environment by integrating knowledge on human behavior. For example, there is one study on agents that plan by learning the behavior of humans directly from a log [11] or on how the behavior of others changes depending on the agent's actions [12]. However, many interaction logs are needed for the human, who is a partner on a collaborative task. Additionally, there are studies that focus on collaboration with others who meet first. This is called "ad-hoc" coordination [13], which is collaboration without opponent information held in advance.

The naive approach for ad-hoc agents is modeling human as rational agents. This approach is used for assertive robots [1] and sidekicks in games [2]. Figure 2.2 is a simple architecture of a collaborative agent in human-agent teaming [2]. First, human-agent teaming problems are modeled as multi-agent MDP (MAMDP). MAMDP communicates one of multiple human policies, which are pre-calculated for all possible human models. Depending on which model communicates with MAMDP, the total problem becomes POMDP for the agent.

For mutually supportive multi-agent models, interactive POMDP (I-POMDP) [14]

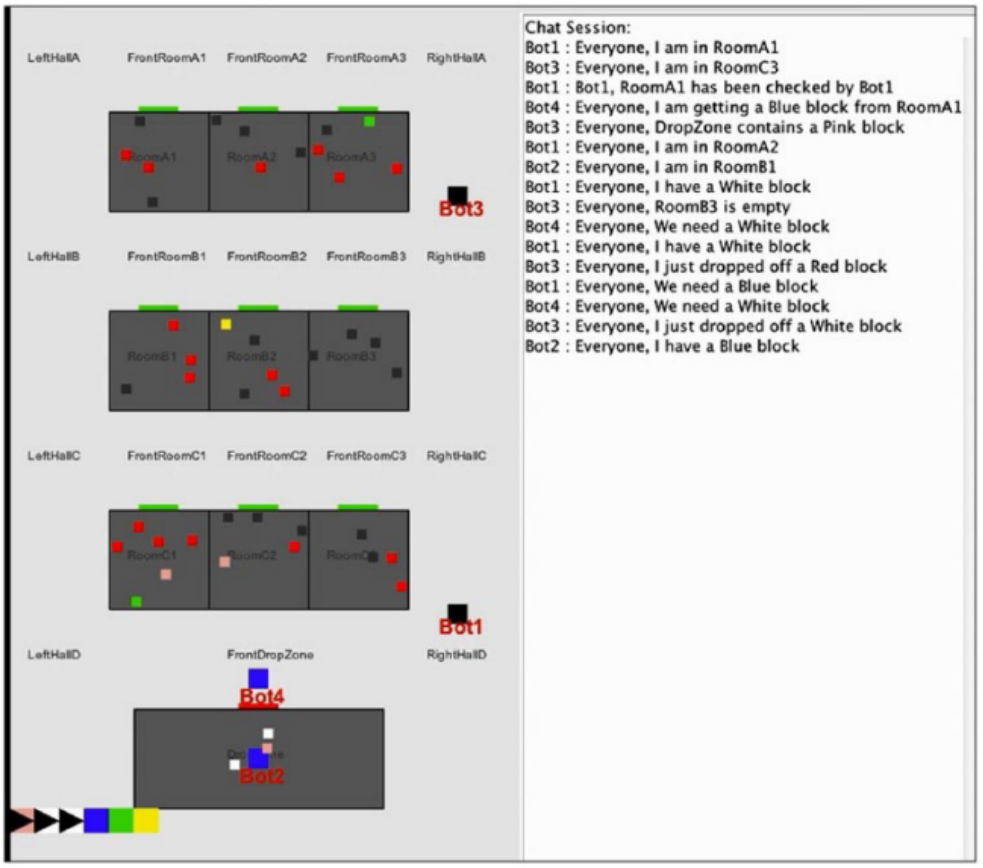

FIGURE 2.3: The example of implicit coodination testbeds

was proposed. I-POMDP can model bi-directional recursive intention inference infinitely; however, the computational costs of considering deep intention inference are huge even if using approximation methods [15–17]. There are studies indicating that humans have cognitive limitations [18, 19] regarding recursive intention inference, so I-POMDP might be too rich of a model for simple environments.

2.3 Collaborative Planning with Communication/-Coordination

The key function of human-agent teaming is communication and coordination with a human. In studies on human-human teams, coordination can be both explicit and implicit [20]. Explicit coordination is communication through mechanisms using commonly understood knowledge such as strategy, planning, and procedure.

The communication protocol can be both verbal and non-verbal. Implicit coordination involves trying to communicate using meta-cognitive approaches such as shared mental models that allow teammates to anticipate each other's actions and intentions, thereby enhancing teaming. In one study, implicit coordination was defined as acting "without consciously trying to coordinate." In another study, it was defined as the "ability of team members to act in concert without the need for overt communication." According to this definition, one study related to implicit guidance showed that a team that uses a lot of "implicit coordination" improves in terms of task performance more than a team that uses a lot of "explicit coordination" in a cooperative task [21]. Figure 2.3 shows a testbed of implicit coordination. The purpose of the task is conveying a load in accordance with a specific rule, and all bots communicate with a language-based protocol. Furthermore, in [21], "implicit coordination" was further divided into "deliberative communication," which involves communicating objectives, and "reactive communication," which involves communicating situations. It is also mentioned that implicit coordination consists of deliberative communication and reactive communication, and that high-performance teams are more likely to use the former type of communication more often than the latter.

Furthermore, there are studies in which the agent communicates with others explicitly. Explicit communication can be regarded as an extra action to convey information to others. In the multi-agent planning domain, there are many studies on communicating using guidance. The major approach is adding a special action for communicating and planning in consideration of when agents communicate with each other [22, 23]. There are studies on support agents whose behavior is helpful for achieving human objectives, such as in [24]. In their setting, helpful behavior has some cost. Their algorithm can judge whether an agent should help others by paying an additional cost.

2.4 Planning with Human's Cognitive Model

The key idea for our agent with implicit guidance is that the considering human cognitive model to make the human inference for the agent's intention. The cognitive function which is the ability to infer other intention, goal, and so on is callde "Theory of Mind". The are some studies for the computational model of the "Theory of Mind". One of populr approach is the Bayesian apploach. It is

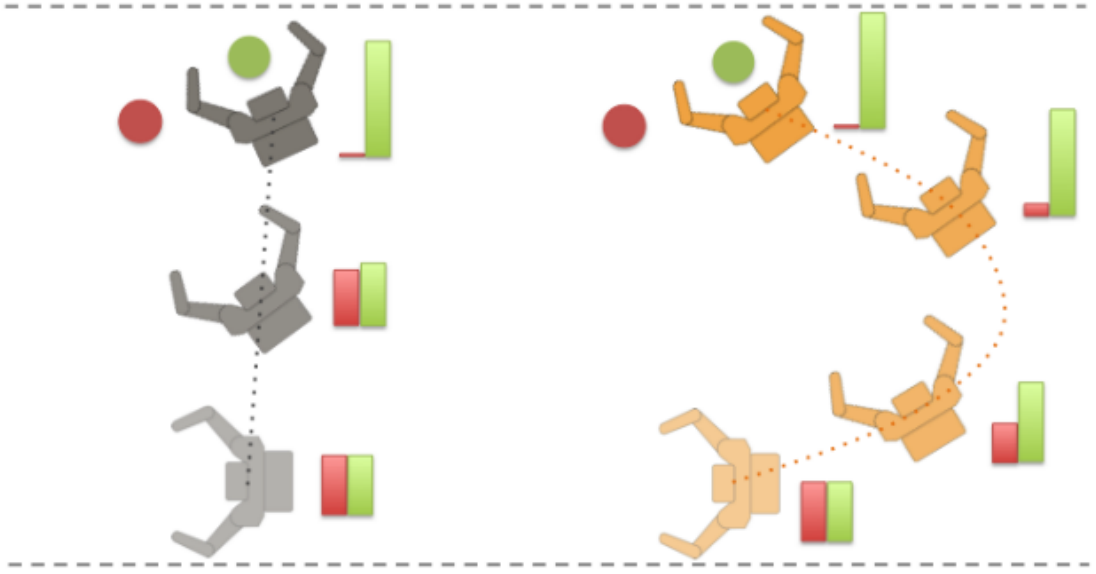

FIGURE 2.4: The example of legible actions

called Bayesian Theory of Mind[25–27]. Bayesian apploach is often used to model human cognition. For example, researchers used a Bayesian model to model how humans teach the concept of an item by showing the item to learners [28]. In other study, a Bayesian model has been used to model how humans teach their own behavioral preferences by giving a demonstration [29]. In addition, there are many extensions of Bayesian Theory of Mind, such as Bayesian Theory of Mind based on an ego-centric perspective [30]. We can improve the performance by choosing appropriate extensions regarding a collaborative task. In addition to the Bayesian Theory of Mind, there are other theories of the Theory of Mind. For example, the Analogical Theory of Mind [31] tries to model the Theory of Mind through the learning of structural knowledge. Furesermore, there is the study for Theory of Mind using neural network [32].

The planing algorithm considering such human cognitive model is the overview of our study. A planning algorithm that considers such human cognitive models is [an overview — the focus?] of our study. In the robotics area, there are studies on integrating a human cognition model to plan for collaboration with humans [33]. One example is robots taking legible action [34]. With a human inference function, a robot can make motions that make their objective understandable to an observer. Figure 2.4 is an example of a legible action. The gray robot arm uses a greedy action policy, the orange one uses a legible action policy, and the purpose of these robots is to grab the green object. The greedy robot heads to the

green object in a straight line. However, the observer has difficulty distinguishing which is the robot's goal, red or green. The legible robot wraps around from the right to reach the object. Although this action needs additional time, observers can understand that the robot wants to get the green object. A more advanced example is collaborative planning under the assumption that a human infers the future planning of a robot and plans their future actions in consideration of this inference, reported in [35]. Human-aware planning [36] has also been proposed as a planning method that incorporates a prediction model for human agents and robots. This planning assumes that the human has learned the behavioral model of the agent/robot, and it adds a bias to the behavior so that the prediction of the model becomes more accurate.

2.5 Human Impression for Collaborative Planning

The main concern of the existing human-agent teaming is improving the performance on tasks, and there are currently not many studies that focus on the impressions that people have of an agent. One of the few studies that have been done investigated task performance and people's impressions for task assignment in a cooperative task involving a human and an AI agent [37]. In this study, it was mentioned that a semi-autonomous setting, in which a human first decides what tasks they want and the agent then decides the rest of the task assignments, is more satisfying than manual control and autonomous control settings in which the human and robot fully assign tasks.

Chapter 3

Background

This chapter introduces the technical background of our methodology. Section 3.1 is about the formalization of a variety of decision-making problems. Section 3.2 is about the existing planning algorithm, which is the basis of our algorithm. Section 3.3 is about the existing collaborative planning algorithm, which is also the basis of our algorithm. Section 3.4 is about the existing computational modeling for theory of mind for humans.

3.1 Formalization of Decision Making Problems

Our human-agent teaming problem is formalized in a framework of decision-making beginning from the Markov decision process. Figure 3.1 is an overview of the decision-making system. There are two factors in a problem for this system: an agent and an environment. The agent is the subject of the decision-making for the problem, and the environment is the target of the problem. The agent observes the current state of the environment and decides on an action for the observed state. Then, the agent performs this action, thereby changing the state of the environment. Next, the agent can observe the changed state and can decide on an action for this changed state. Using this loop, the agent can perform sequential decision-making. Note that this overview is about only the flow of information;

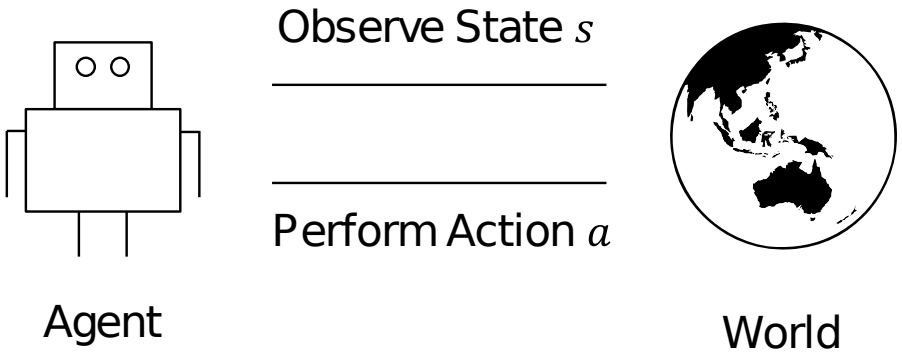

FIGURE 3.1: Overview of decision system

for a real problem, the system needs a mechanism for obtaining information such as from a sensor to get information on states and a mechanism for performing actions that agents decide.

Formally, $\mathcal{S}$ represents a set of states of the environment, and $s \in \mathcal{S}$ represents one state. $\mathcal{A}$ represents a set of actions that the agent can take, and $a \in \mathcal{A}$ represents one state.

An agent needs a rule to decide their actions given the observed state in these decision-making problems. The rule is called a policy π. Theoretically, a policy is a function that returns an action from a state. If the function is deterministic, the form of the policy is represented as $\mathcal{S} \to \mathcal{A}$ formally. If the function is stochastic, the form of the policy is represented as $\mathcal{S} \times \mathcal{A} \to [0..1]$ formally, and $\pi(a|s)$ represents the probability that an agent takes action a when it observes state s. It represents the probability that the agent takes an action in a state.

3.1.1 MDP

A Markov decision process (MDP) [38] is a discrete-time stochastic control process. It provides a mathematical framework for modeling decision-making in situations where outcomes are partly random and partly under the control of a decision-maker (Agent). There are two factors, the agent and environment, in MDP as a decision-making system. In MDP, an agent accesses the environment by observing

the current state and taking action in the environment. Then, the agent can observe the following state that results from the agent's action.

Formally, MDP is represented as a tuple $< \mathcal{S}, \mathcal{A}, T, R, \gamma >$. $\mathcal{S}$ and $\mathcal{A}$ are a set of states and actions, respectively. T is a set of state transition functions. The form of the state transition function is $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, and $T(s'|s, a)$ shows the probability that the agent observes state s' after taking action a when its current state is s. R is a set of reward functions. The form of a reward function is $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, and $R(s, a, s')$ shows the reward when the agent observes state s' after taking action a when its current state is s. R can be in other forms, such as $R(s)$ and $R(s, a)$. γ is a discount factor determining how much an agent cares about rewards in the distant future relative to those in the immediate future.

3.1.2 POMDP

The partially observable Markov decision process (POMDP) [39, 40] is an extension of MDP for situations in which agents cannot observe the complete information of a state to decide their action. In POMDP settings, an agent cannot observe the current state directly; instead, the agent can observe the information depending on the current state.

Formally, POMDP is represented as a tuple $< \mathcal{S}, \mathcal{A}, \Omega, T, O, R, \gamma >$. Ω is a set of observations, and $o \in \Omega$ represents one observation. O is a set of observation functions. The form of the functions is $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, and $O(o|s, a)$ shows the probability that an agent observes o after taking action a when the current state is s. The other factors of the tuple are the same as MDP.

3.1.3 MOMDP

The mixed-observability Markov decision process (MOMDP) [41] is one of the subspecific representations of POMDP for situations in which the state space can be factorized clearly into an observable part and partially-observable part. This representation factors state space into fully-observable and partially-observable components. The computational cost of POMDP depends on the size of the

partially-observable state, so by reducing the size of partially-observable components by separating them from fully-observable components, this problem can be solved more effectively.

Formally, MOMDP is represented as a tuple $< \mathcal{X}, \mathcal{Y}, \mathcal{A}, \Omega, T, O, R, \gamma >$. $\mathcal{X}$, $\mathcal{Y}$ is a set of components of the state space that are fully- and partially- observable, and $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ represent the components for one state. Along with this, the forms of transition function $T(s'|s, a)$ can be also represented as $T((x', y')|(x, y), a)$. Similarly, $O(o|s, a)$ can be also represented as $O(o|(x, y), a)$.

3.1.4 Dec-POMDP

Decentralized POMDP (Dec-POMDP) [42] is an extension of POMDP for multi-agent setups that deals with a specific case in which all agents share the same reward function of a partially observable stochastic game (POSG) [43].

Dec-POMDP is defined in the format $< \mathcal{I}, \mathcal{S}, \{\mathcal{A}_i\}, \{\Omega_i\}, T, R, O >$. $\mathcal{I}$ is a set of agents. $\mathcal{A}_i$ is a set of actions for agent i, with $\mathcal{A} = \times\mathcal{A}_i$ being the set of joint actions for all agents. Ω_i is a set of observations for agent i, with $\Omega = \times\Omega_i$ being the set of joint observations for all agents.

A solution of Dec-POMDP is a set of all agent policies, each that maximizes the value of the obtained total rewards.

3.2 Planning Algorithm

The solution of the planning algorithm is obtained in the form of a policy. As described above, there are two types of policies: deterministic and stochastic. In this thesis, we deal with the agent's policy as stochastic.

3.2.1 Planning on MDP

The best policy is a policy with which an agent can obtain the maximum cumulative rewards accordingly. This policy can be obtained by using simple dynamic programming [38]. First, consider the expected cumulative reward value when the

agent takes actions in accordance with a specific policy π after the agent takes action a from a state; the value is called the action value function, and it is represented as $Q^{\pi}(s, a)$. Given a set of initial states $\mathcal{S}_0$, the condition that the best policy π^* should satisfy is as follows.

$$\pi^* = \underset{\pi}{\operatorname{argmax}} E_{s \in \mathcal{S}_0} \left[Q^{\pi}(s, a) \right] \tag{3.1}$$

As a solution that obtains the best policy, the policy improvement theorem was proposed [44]. The theorem shows that improving the action for one specific state improves the overall cumulative reward of the policy. According to the theory, given a specific policy, choosing the best action for all states is a better policy than the given policy. Formally, the policy π' in the following equation is a better policy than π.

$$\pi'(a|s) = \mathbb{1}(a = \underset{a' \in \mathcal{A}}{\operatorname{argmax}} Q^{\pi}(s, a')) \tag{3.2}$$

Next, the expected total reward value when an agent takes actions according to a specific policy π from a state is considered. The value is called the value state function, and it is represented as $V^{\pi}(s)$. The relation of the action value function and state value function is as follows.

$$Q^{\pi}(s, a) = \sum_{s' \in \mathcal{S}} T(s'|s, a) \left(R(s, a, s') + \gamma V^{\pi}(s') \right) \tag{3.3}$$

Furthermore, the value function can be defined recurrently using the state value function regarding one step after.

$$V^{\pi}(s) = \sum_{a \in \mathcal{A}} \left(\pi(a|s) \sum_{s' \in \mathcal{S}} T(s'|s, a) \left(R(s, a, s') + \gamma V^{\pi}(s') \right) \right) \tag{3.4}$$

The equation is called the Bellman equation. Connecting the policy improvement theorem in Eq. 3.1, the state value function for an improved policy becomes as follows.

$$V^{\pi'}(s) = \max_{a \in \mathcal{A}} \left(\sum_{s' \in \mathcal{S}} T(s'|s,a) \left(R(s,a,s') + \gamma V^{\pi}(s') \right) \right) \tag{3.5}$$

This value update procedure is called value iteration. Since this iteration scans all combinations of source states, target states, and actions regarding the transition function, the computational complexity of one step is $O(|\mathcal{S}|^2|\mathcal{A}|)$.

As the result of value iteration, the value became stable as follows, and such a policy is the best policy.

$$V^{\pi'}(s) = V^{\pi}(s) \ (\forall s \in \mathcal{S}) \tag{3.6}$$

3.2.2 Planning for POMDP

In the POMDP setting, the agent cannot observe the state directly; thus, the agent has to estimate the current state of the environment. The estimation is called the belief of the state. The belief $b(s)$ represents the probability that the current state is s, and the belief is a distribution over the state space, formally $b \in \Delta(\mathcal{S})$.

When the agent performs a, it makes observation o. An observation is obtained from the current actual state, which can be a clue to the current actual state. Thus, the agent can update their belief using the information in each step. We represent the updated belief observing o after action a as b^a_o, and the belief for each state s can be calculated as follows.

$$b^a_o(s) \propto \sum_{s' \in \mathcal{S}, o \in \Omega} P(s,o|s',a) b(s') = \sum_{s' \in \mathcal{S}, o \in \Omega} T(s|s',a) O(o|s',a) b(s') \ (\forall s \in \mathcal{S}) \tag{3.7}$$

Since an agent can observe states partially, the value is defined not regarding state but rather state belief. We can define the action value function for POMDP like in Eq. 3.3.

$$Q(b,a) = \sum_{s,s' \in \mathcal{S}} b(s)T(s'|s,a)R(s,a,s') + \gamma \sum_{o \in \Omega} O(o|s,a)V(b_o^a) \tag{3.8}$$

V is state value function that can be calculated as follows

$$V(b) = \max_{\alpha \in \mathcal{V}}(b \cdot \alpha) = \max_{\alpha \in \mathcal{V}}(\sum_{s \in \mathcal{S}} b(s)\alpha(s)) \tag{3.9}$$

$\mathcal{V}$ is a set of α, which is an α-vector. An α-vector is a $|\mathcal{S}|$ dimensional vector; each value corresponds to the expected cumulative reward when executing one conditional plan (finite action sequence) from a specific state. We represent a conditional plan as $\sigma = (a, v)$, in which v is a mapping from observations to future conditional plans (one step shorter) for the agent to follow. The value of the α-vector for a specific conditional plan σ and s is defined as follows.

$$\alpha_\sigma(s) = \sum_{s' \in \mathcal{S}} T(s'|s,a)R(s,a,s') + \gamma \sum_{s' \in \mathcal{S}, o \in \Omega} P(s',o|s,a)\alpha_{v(o)}(s') \tag{3.10}$$

Furthermore, the set of values of all α-vectors can be calculated recursively using the α-vector for one-step-shorter conditional plans $\mathcal{V}'$ using the following steps [45].

$$\begin{aligned}
\Gamma^a : \alpha^{a,*}(s) &= R(s,a) \\
\Gamma^{a,o} : \alpha^{a,r}(s) &= \gamma \sum_{s' \in \mathcal{S}} P(s',o|s,a)\alpha'(s') \ (\forall \alpha' \in \mathcal{V}') \\
\Gamma^a &= \Gamma^{a,*} \oplus \Gamma^{a,o_1} \oplus \Gamma^{a,o_2} ... \\
\mathcal{V} &= \cup_{a \in \mathcal{A}} \Gamma^a
\end{aligned} \tag{3.11}$$

$\oplus$ is a cross-sum operator. In practice, an α-vector that is completely dominated by another α-vector is pruned from the final set of $\mathcal{V}$. However, the size of α-vectors for one step can be $O(|\mathcal{A}||\mathcal{V}'|^{|\Omega|})$ in the worst case. Taking into account the source and target states, the computational complexity of this step is $|\mathcal{S}|^2|\mathcal{A}||\mathcal{V}'|^{|\Omega|}$.

Since this computational cost is extremely high, this exact solution can be used for only tiny problems. Thus, there are a variety of approximative solutions for larger and more practical problems. Point-based value iteration (PBVI) [46] is one basic approximative solution. In this approach, the belief space is reduced to a finite set of belief points B, and α-vectors corresponding to each belief point are calculated. Taking into consideration finite belief points, the cross-sum step is simplified as follows.

$$\Gamma_b^a = \Gamma^{a,*} + \sum_{o \in \Omega} \underset{\alpha \in \Gamma^{a,o}}{\operatorname{argmax}} (\alpha \cdot b) \tag{3.12}$$

Then, the value function changes correspondingly.

$$V(b) = \underset{\Gamma_b^a, \forall a \in \mathcal{A}}{\operatorname{argmax}} (\Gamma_b^a \cdot b), \quad \forall b \in B \tag{3.13}$$

This simplification reduces the computational cost exponentially. The total computational complexity is $O(|\mathcal{S}||\mathcal{A}||\mathcal{V}'||\Omega||B|)$

3.2.3 Planning for MOMDP

As we described in Sec. 3.1.3, MOMDP is a subspecific representation of POMDP. Thus, the planning algorithm for MOMDP is almost the same as the algorithm of POMDP. The difference is that we need the belief for only a partially-observable part of a state, not the entire state. That is, the belief is the distribution over the partially-observable part of the state space, formally $b \in \Delta(\mathcal{Y})$. In addition, the agent can observe an observable part of the state in each step. Based on this, the updated belief $b_{o,x}^{a,x'}(y)$, that is, the belief for the partially-observable part y when the agent takes action a in x' and observes x and o, can be calculated as follows.

$$\begin{aligned} b_{o,x}^{a,x'}(y) &\propto \sum_{y' \in \dagger, o \in \Omega} P((x,y), o | (x',y'), a) b(y') \\ &= \sum_{s' \in \mathcal{S}, o \in \Omega} T((xy) | (x',y'), a) O(o | (x',y'), a) b(y') \ (\forall y \in \mathcal{Y}) \end{aligned} \tag{3.14}$$

Similarly, the action value function and state value function can be calculated as follows.

$$
\begin{aligned}
Q(b,x,a) &= \sum_{x' \in \mathcal{X}} Q'(b,x,x',a) \\
Q'(b,x,x'a) &= \sum_{y,y' \in \mathcal{Y}} b(y) T((x',y')|(x,y),a) R((x,y),a,(x',y')) + \\
&\quad \gamma \sum_{o \in \Omega} O(o|(x,y),a) V(b^{a,x}_{o,x'}, x') \qquad (3.15) \\
V(b,x) &= \max_{\alpha \in \mathcal{V}} (b \cdot \alpha) = \max_{\alpha \in \mathcal{V}} (\sum_{y \in \mathcal{Y}} b(y) \alpha^x(y)) \qquad (3.16)
\end{aligned}
$$

$\alpha^x(y)$ is the α-vector for MOMDP under the condition that the observable part of the current state is s. According to the definition of the state value function, to plan for MOMDP, we need to solve individual POMDP for each x. From a computational perspective, this is undesirable. However, the computational cost of each POMDP planning is reduced drastically since the dimension of the α-vector is reduced to $|\mathcal{Y}|$. As a result, the total computational cost is also reduced drastically compared with the planning of POMDP.

3.3 Existing Collaborative Planning Algorithm

3.3.1 Cooperatives Inverse Reinforcement Learning (Assistance Game)

Cooperative inverse reinforcement learning (CIRL) [47], which is also called an assistance game, is a special-case of two-player Dec-POMDP with partial information. In CIRL, the information of the reward is known by only the human. Thus, agents expect humans to share enough information for the reward function via their actions for excellent cooperation.

Formally, the CIRL game is represented as a tuple $< \mathcal{X}, \{\mathcal{A}_H, \mathcal{A}_A\}, T, \{\Theta, R\}, \gamma >$. $\mathcal{X}$ is a set of world states, which is observable from both the human and agent. $\mathcal{A}_H$, $\mathcal{A}_A$ are actions available to the human and agent. $T : \mathcal{X} \times \mathcal{A}_H \times \mathcal{A}_A \to [0..1]$ is a state transition distribution. Θ is a set of reward parameters that is observable by only the human, $R : \mathcal{X} \times \mathcal{A}_H \times \mathcal{A}_A \times \Theta \to \mathbb{R}$ is a reward function, and γ is a discount factor.

3.3.2 Planning for CIRL

The solution of CIRL is to obtain a collaborative policy for both the human and the agent. Since humans and agents share one reward function, the solution includes collaborative actions (active teaching, active learning, etc.) as the optimal policy. CIRL is in Dec-POMDP, which is an NEXP-complete problem [48]. CIRL can be reduced to a coordination-POMDP, which is a kind of POMDP. However, the size of the action space is $|\mathcal{A}^H|^{|\Theta|}|\mathcal{A}^A|$, which is too large in most cases. Therefore, it is difficult to solve CIRL by taking a naive approach.

Coordination-POMDP reduced from CIRL can reduce the size of the action space to $|\mathcal{A}^A|$ with a small modification to the Bellman update of POMDP [49]. The idea is that it is assumed that a human plans their action rationally on the basis of their value, and such a human planning process is integrated in the Bellman update.

To consider CIRL planning, we define a variable that is the Cartesian product of the world state and reward parameter $\mathcal{S} = \mathcal{X} \times \Theta$. It corresponds to the state of MOMDP, in which the world state is an observable part and the reward parameter is an unobservable part for an agent. If we see the CIRL problem as a planning problem for the agent, human action can be regarded as an observation. That is, $P(s', o|s, a)$ can be rewritten as $P(s', a_H|s, a_A)$. The probability can be divided into two factors, the probability that the human takes action a_H given s, a_H, and the probability of the transition of the next state given the actions.

We write the probability of human action as a human policy π_H. Generally, the input of a policy is the current state described above; however, the human decides their action after observing the agent's actions in CIRL. Furthermore, the human should decide their actions in accordance with the reward information. Thus, the the human (stochastic) policy becomes $\mathcal{S} \times \mathcal{A}_A \times \mathcal{A}_H \times \Theta \to [0..1]$.

Summarizing the above, $P(s', a_H|s, a_A)$ can be rewritten as follows, in which $s = (x, \theta), s' = (x', \theta')$.

$$P(s', a_H|s, a_A) = T(x'|x, a_H, a_A)\pi_H(a_H|s, a_A)\mathbb{1}(\theta = \theta') \tag{3.17}$$

In CIRL, it is hypothesized that humans behave rationally when considering an agent's future action. This means that the human's action value function is conditioned by the agent's conditional plan *sigma*. Thus, the action value function for the human is calculated as follows.

$$Q_H(s, a_H, \sigma) = \sum_{s \in \mathcal{S}} T(x'|x, a_H, a_A) \cdot \alpha_{v(a_H)}((x', \theta)) \tag{3.18}$$

Likewise, the human's policy and the probability that the agent observes the human's action is also conditioned by *sigma* (in Eqs. 3.10, 3.11).

$$P(s', a_H|s, \sigma) = T(s'|s, a_H, a_R)\pi_H(a_H|s, \sigma) \tag{3.19}$$

If the human decides their action deterministically, the policy is as follows.

$$\pi_H(a_H|s, \sigma) = \mathbb{1}(a_H = \underset{a'_H \in \mathcal{A}_H}{\operatorname{argmax}} Q_H(s, a_H, \sigma)) \tag{3.20}$$

If the human decides their action stochastically, the policy is as follows.

$$\pi_H(a_H|s, \sigma) \propto \exp(\beta Q_H(s, a_H, \sigma)) \tag{3.21}$$

The β is a hyper parameter for rationality.

The basic process is similar to POMDP. The additional computational cost is times $|\Theta|$ to calculate the max action. Thus, the total computational complexity is $O(|\mathcal{S}||\mathcal{A}^{\mathcal{A}}||V'||\mathcal{A}^{\mathcal{H}}||B||\Theta|)$.

3.4 Bayesian Modeling for Theory of Mind

As mentioned, humans have a function for inferring others' goals or intentions from their behavior called theory of mind. Our method is based on the Bayesian theory of mind.

3.4.1 Model of Human Rationality

Bayesian theory of mind assumes that if humans have goals or intentions, they will behave rationally to achieve these goals or intentions. We use Boltzmann noisy rationality [27] as the rationality of action for our model. In the definition of Boltzmann noisy rationality, $P(a|g)$ is the probability that an agent executes a for goal g in accordance with the Boltzmann distribution of the "value" of a for achieving g. Equation 3.22 is the definition of the probability. β is the temperature parameter of the Boltzmann distribution for defining rationality.

$$P(a|g) = \frac{\exp\left(\beta Q_g(a)\right)}{\sum_{g' \in \mathcal{G}} \exp\left(\beta Q_{g'}(a)\right)} \tag{3.22}$$

Here, $Q_g(a)$ corresponds to the action value function in terms of the MDP planning described Sec. 3.2.1. This rationality is also called a softmax policy in the planning area.

This rationality is the one of the basic behavioral models, that is, a logit model, in the behavioral economics area [50]. Thinking that we model human behavior using a simple model based in rational choice theory [51] and random utility theory [52], when we use a Gumbel distribution [53] as the error term for utility, the best choice of theory corresponds to the Boltzmann distribution over the utility function.

This modeling is one of the most popular models of stochastic rational autonomy. For example, it is used in many models for theory of mind and many methods [25–27, 29, 54] of inverse reinforcement learning [55–58].

3.4.2 Bayesian Theory of Mind

In recent cognitive science areas, the modeling of human reasoning using the Bayesian theorem has become popular [59]. When humans observe an event e, they infer the probability of the cause c as follows.

$$P(c|e) \propto P(e|c)P(c) \tag{3.23}$$

$P(c|e)$ is the human's posterior probability that c is the cause of an observed e, $P(e|c)$ is the likelihood that c causes e, and $P(e)$ is the prior probability that e happens independently. $P(e)$ is the human's prior knowledge such as common sense and so on. If a modeler does not want to implement such knowledge, they can use a uniform distribution for $P(e)$ and erase the term from Eq. 3.23.

Theory of mind is human's cognitive function to understand other goals, preferences, intention and so on [4]. Bayesian Theory of Mind [25–27] is the one of application of the Bayesian human reasoning model.

The objective of Theory of Mind is infering others goal is g when observing others action a.

$$P(g|a) \tag{3.24}$$

According to Eq. 3.23, the inference of the probability that goal g is the reason for action a is as follows.

$$P(g|a) \propto P(a|g)P(g) \tag{3.25}$$

$P(a|g)$ is also the probability of planning given specific goal g. Using Boltzmann rationality (Equation 3.22), we can calculate it.

Since, this apploach reversed the dependency of the variables and P is the likelihood of the planning, it is also be called Inverse Planning [25].

Chapter 4

Human-Agent Teaming

This chapter explains the practical definition of Human-Agent Teaming. Section 4.1 explains what is Human-Agent Teaming beliefly. Section 4.2 is about the definition of the Human-Agent Teaming. Section 4.3 explains the planning of the supportive agent in the Human-Agent Teaming definition.

4.1 Example of Human-Agent Teaming

We defined Human-Agent Teaming is as a collaborative task. In the task, the human and the agent are in the same environment and share the problem's objective(s) information. The success of the problem is the achieve one objective in the problem. Figure 4.1 is a simple example of Human-Agent Teaming, in which the objective of the problem is to capture the characters (shown as a face picture) which are in between theirs. The characters try to escape from the humans and the agent at the same speed as humans. Thus This problem is a kind of the task is a kind of pursuit-evasion problem [60]. To achieve the problems' objective (capture on character), the human and the agent have to approach the same character from both sides. If the human and the agent pursue other characters, there can not catch up with each character forever, and then, the problem fails because

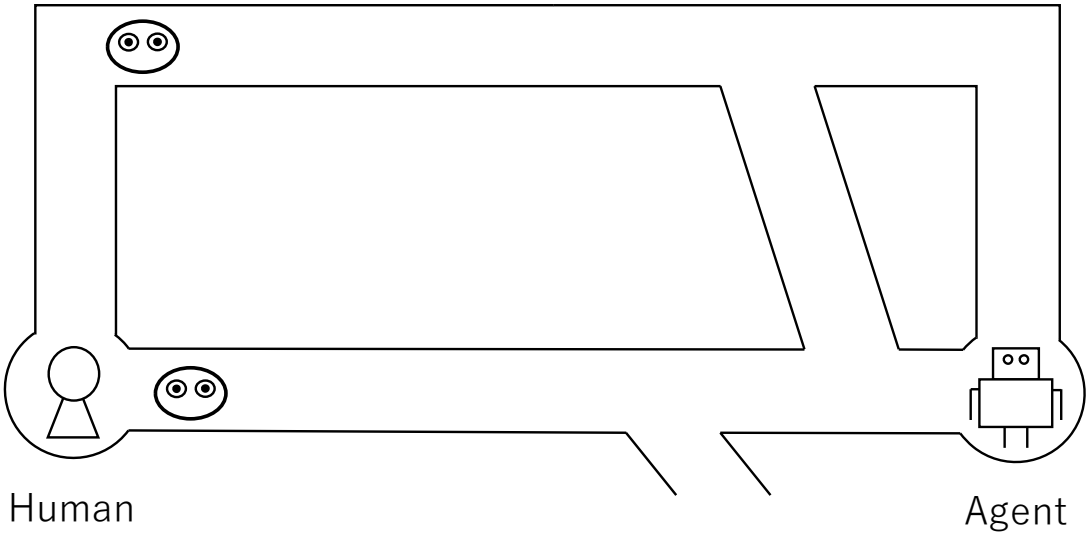

FIGURE 4.1: Example of agent with Human Agent Teaming

characters escape out through the lower bypass. Thus, the human and agent have to cooperate with sharing the character which there should approach to.

4.2 Formalization of Human-Agent Teaming

First, we model the human-agent teaming problem as a kind of Dec-POMDP by treating the human as one agent.

The current objective in a state is dealt with as an unobservable factor. In other words, the human and agent are in multiple Dec-MDP environments that have only one objective. They move over the environments at each step, but they cannot observe which environment they are in directly.

Based on the above, we model Dec-POMDP. $\mathcal{I}$ consists of an agent and a human $\{i_A, i_H\}$, so $\mathcal{A}$ consists of the human's action and agent's action; thus, it can be represented as $\mathcal{A}_A \times \mathcal{A}_H$. In human-agent teaming problems, the human and the agent cannot directly know the partner's objective. We model such a situation as states that include specific information that cannot be observed by the partner. We define each information space as Θ_H and Θ_A, respectively. We define information that can shared with one another such as each position as $\mathcal{O}$. The state space becomes $\mathcal{S} = \mathcal{O} \times \Theta_H \times \Theta_A$. This representation follows the MOMDP

representation. Moreover, observations become equal to the observable factors of states, formally, $\Omega = \mathcal{O}$. Furthermore, the objective does not affect the observable factors of states, that is, the transition function T can be factorized into observable state part $T^{\mathcal{O}} : \mathcal{O} \times \mathcal{A}_H \times \mathcal{A}_A \times \mathcal{O} \to [0..1]$ and unobservable state part $T^{\Theta_H} : \mathcal{S} \times \mathcal{A}_H \times \mathcal{A}_A \times \Theta_H \to [0..1]$ and $T^{\Theta_A} : \mathcal{S} \times \mathcal{A}_H \times \mathcal{A}_A \times \Theta_A \to [0..1]$.

Based on the above formulation, we model the human-agent teaming problem. First, the objective of the problem is shared between the agent and the human. We represent a common objective as Θ. In an actual environment, the agent cannot control a human's actions directly. Thus, the action state should consist of only the agent's action, formally $\mathcal{A} = \mathcal{A}_A$. Instead, we introduce human policy π_H like in CIRL. $\pi_H(a_H|s, a_A, \theta)$ represents the probability that the human takes action a_H after observing the agent take action a_A in state s. Furthermore, the agent and human have to share each other's objective, and we consider only the human's objective. As a result, the form of the state transition function and the reward function is converted to $T^{\mathcal{O}} : \mathcal{S} \times \mathcal{A}_A \times \mathcal{O} \to [0..1]$, $T^{\Theta} : \mathcal{S} \times \mathcal{A}_A \times \Theta \to [0..1]$ and $R : \mathcal{S} \times \mathcal{A}_A \to \mathbb{R}$, and the function can be calculated as follows.

$$
\begin{aligned}
T^{\mathcal{O}}(o'|(o,\theta), a_A) &= \sum_{a_H \in \mathcal{A}_H} \pi_H(a_H|(o,\theta), a_A) T^{\mathcal{O}}(o'|o, a_A, a_H) \\
T^{\Theta}(\theta'|(o,\theta), a_A) &= \sum_{a_H \in \mathcal{A}_H} \pi_H(a_H|(o,\theta), a_A) T^{\Theta}(\theta'|(o,\theta), a_A, a_H) \\
R((o,\theta), a_A) &= \sum_{a_H \in \mathcal{A}_H} \pi_H(a_H|(o,\theta), a_A) R((o,\theta), a_A, a_H)
\end{aligned} \tag{4.1}
$$

Note that $s = (o, \theta)$.

Here, we hypothesize that the process of the human's decision-making consists of two steps: humans decide their target and take action toward the target. Therefore, the human's policy can be factorized as follows.

$$
\pi_H(a_H|(o,\theta), a_A) = \sum_{\theta' \in \Theta} P(\theta'|(o,\theta), a_H) P(a_H|(o,\theta'), a_A) \tag{4.2}
$$

We also assume that humans behave rationally.

$$P(a_H|(o,\theta'),a_A) \propto \exp\left(\beta \cdot Q((o,\theta),a_A,a_H)\right) \tag{4.3}$$

Note that θ is the only unobservable factor for the problem. Thus, given θ, the problem is reduced to the MDP with the joint action space. Therefore, $Q((o,\theta),a_A,a_H)$ can be calculated using basic value iteration. As an intuitive explanation, if an agent knows a human's objectives in advance, the agent can make the best plan to collaborate with the human.

4.3 Planning of Supportive Agent in Human-Agent Teaming

We explain the planning of supportive agents in human-agent teaming. As we explain in Sec. 1, supportive agents infer the objective of a human and do not expect the human to change their objective. This can be modeled as follows.

$$P(\theta'|s,a_A,\theta) = \mathbb{1}(\theta = \theta') \tag{4.4}$$

The important point is that the function is not dependent on the agent's action. This means that the human's policy is independent of the agent's action. Thus, we can implement the human's policy into the state transition. As a result, human-agent teaming is reduced to a simple POMDP (MOMDP, more strictly) problem. We can solve the problem by using the existing planning algorithm for POMDP.

Chapter 5

Planning with Implicit Guidance

This chapter explains our first study for the basic framework of an agent based on implicit guidance. Section 5.1 is an overview of implicit guidance and explicit guidance. Section 5.2 explains how this guidance into implemented in agents. Section 5.3 refers to a participant experiment, and Sec. 5.1 shows the result of the experiment. Finally, there is a brief discussion in Sec. 5.5.

5.1 Implicit Guidance and Explicit Guidance

5.1.1 Implicit Coordination and Explicit Coordination

There are many specific definitions of implicit and explicit coordination. For example, some studies define implicit coordination as coordination that relies on the anticipation of information and resource needs of other team members, and explicit coordination is the transfer of information and resources in response to requests [21]. Another study defines implicit coordination as anticipating one another's information and explicit coordination involves mechanics such as prompts or requests for information amongst teammates [61]. Combining such definitions, we define implicit coordination as the transfer of information with the expectation

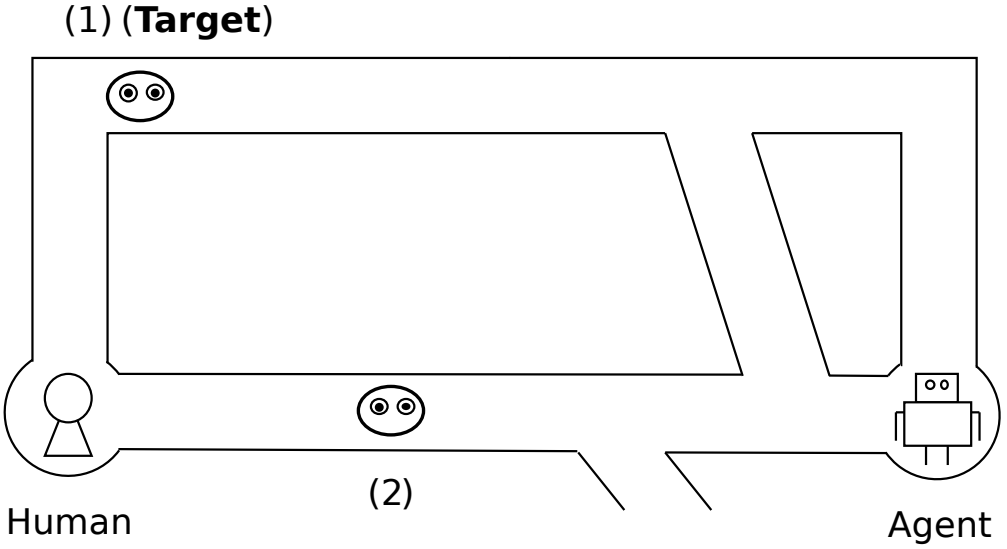

FIGURE 5.1: Example of human-agent teaming (1)

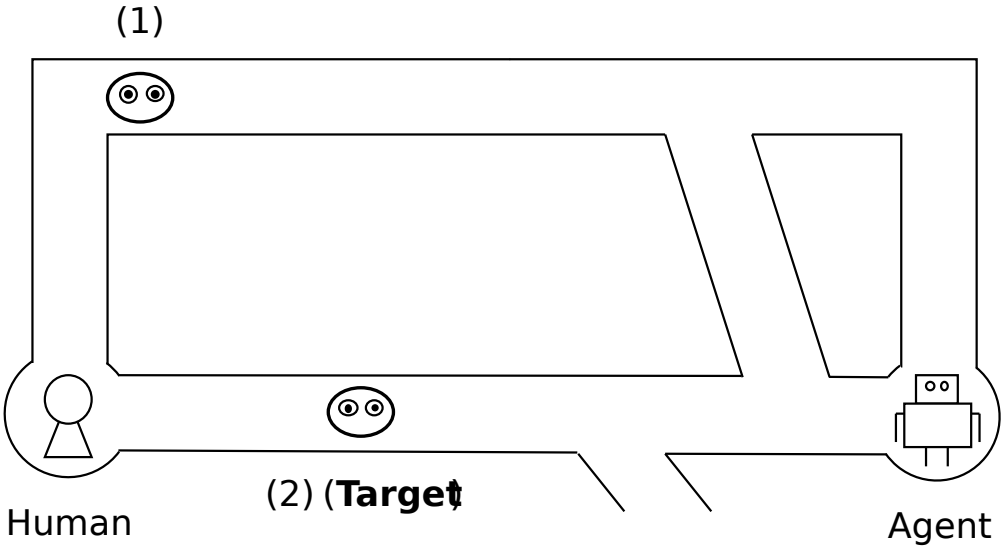

FIGURE 5.2: Example of human-agent teaming (2)

of the ability to infer others' intentions, and explicit coordination is the transfer of information using a recognizable protocol.

5.1.2 Implicit Guidance and Explicit Guidance for Human-Agent Teaming

In general, humans cannot plan optimal actions for difficult problems due to limitations in their cognitive and computational abilities. Figures 5.1 and 5.2 show an example of a misleading human-agent teaming task as one such difficult problem.

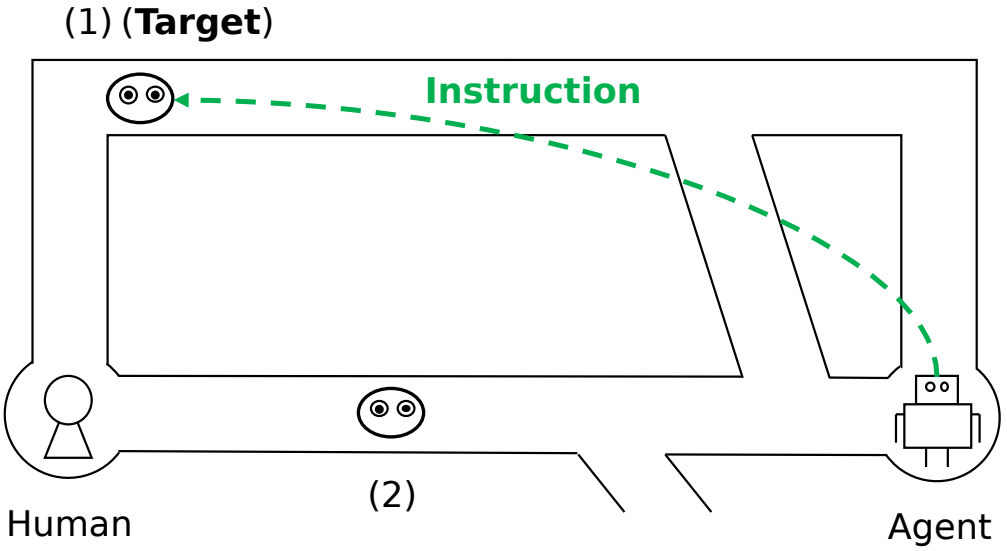

FIGURE 5.3: Example of explicit guidance agent

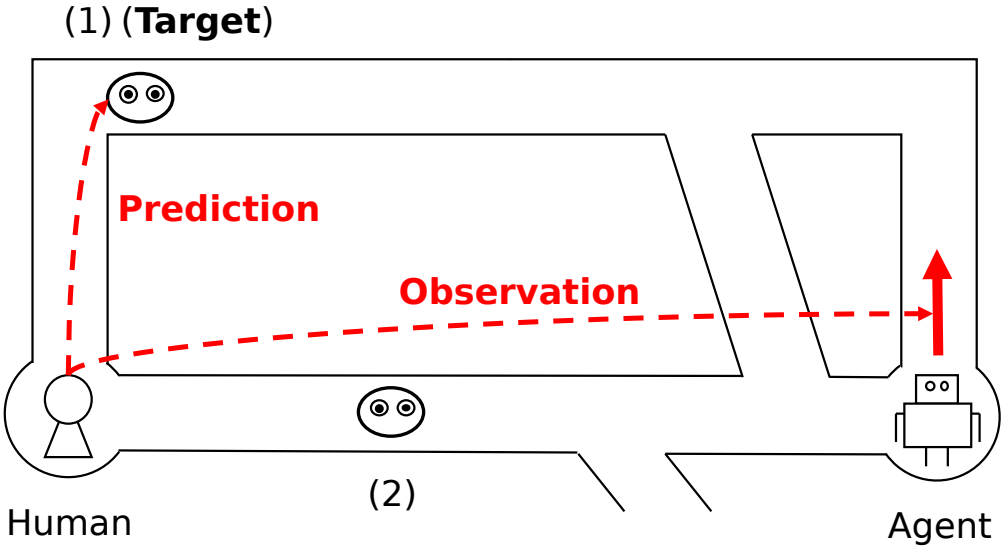

FIGURE 5.4: Example of implicit guidance agent

In Figure 5.1, there are two characters, (1) and (2), on the upper and lower roads, respectively. Since (1) is farther away, (2) seems to be a more appropriate target. However, (2) cannot be captured because it can escape via the lower bypath. On the other side, in Fig. 5.2, the agent and the human can successfully capture (2) because it is slightly farther to the left than in Fig. 5.1. Thus, the best target might change due to a small difference in a task, and this can be difficult for humans to judge. For such a situation, the guidance from an agent for a desirable character to capture is helpful.

Figure 5.3 is an example of explicit guidance for the problem in Fig. 5.1. The

agent guides the human to a desirable target directly and expects the human to follow. Figure 5.4 is an example of implicit guidance. The agent has two kinds of expectations for the human. One is that the human will infer the agent's intention from the agent's behavior, and another is that the human will make an adequate plan to collaborate with the agent on the basis of this inference. The human and the agent have to head to the same character regarding the latter. Thus, the agent expects the human to head to the character that the human infers as the agent's objective. Under these expectations, the agent concentrates on informing the human as to which character the agent wants to get regarding the former. Figure 5.4 is a movement made to inform the human of the agent's objective. When the agent moves upward, the human can infer that the agent is aiming for the upper target by observing the agent's movement. Although this is technically the same thing as the agent showing the target character explicitly, we feel that, in this case, humans would feel as though they were able to maintain autonomy by inferring the agent's target voluntarily.

5.2 Planning with Implicit Guidance and Explicit Guidance

We implement a planning algorithm for agents based on explicit guidance and implicit guidance. The difference from the supportive agent is that the agents assume that they can change the human's objective θ. Thus, the only difference is the formalization of $P(\theta'|(o,\theta),a_A)$.

5.2.1 Planning with Explicit Guidance

The explicit guidance agent guides the human toward the best target; thus, it assumes that the human knows what the best target is. We represent the best target as θ^* as follows.

$$P(\theta'|(o,\theta),a_A) = \theta^* \tag{5.1}$$

The best target is calculated as $\theta^* = \mathrm{argmax}_{\theta' \in \theta} V((o_0,\theta))$, where o_0 is the initial observable state, and $V((o_0,\theta))$ is the state value function of the problem given θ.

5.2.2 Planning with Implicit Guidance

In the current problem, we assume that the human changes their objective to match the agent's one. Thus, the probability of the human inferring the agent's target objective becomes the same as the probability of the human objective. The implicit guidance agent assumes that humans change their target by observing the agent's actions. We assume that humans infer the target of the agent on the basis of Boltzmann rationality, as suggested in earlier theory of mind studies [25] [54].

$$P(\theta'|(o,\theta),a_A) \propto P(\theta)P(a_A|o,\theta) \tag{5.2}$$

$P(a_A|o,\theta)$ is also based on Boltzmann rationality:

$$P(a_A|o;\theta) = \frac{\exp\left(\beta_2 V((T^{\mathcal{O}}(a_A),\theta))\right)}{\sum_{a'_A \in \mathcal{A}_A} \exp\left(\beta_2 V((T^{\mathcal{O}}(a'_A),\theta))\right)} \tag{5.3}$$

where β_2 is a rational parameter, and $V((T^{\mathcal{O}}(a'_A),\theta)))$ is the state value function of the problem regarding the state after a_A given θ. Finally, this calculation is completely the same as POMDP planning, the difference being the transition function. Thus, the total computational complexity is the same as that of POMDP. Considering the difference of notation, the computational time complexity is $O(|\mathcal{S}||\mathcal{A}^{\mathcal{A}}||\mathcal{V}'||\mathcal{A}^{\mathcal{H}}||B|)$ if we use a point-based value iteration approach.

5.2.3 Decide Agent Actions

By solving POMDP (MAMDP) as shown in Secs. 3.2.2 and 3.2.3, we can calculate the action value function $Q(o,\theta,a)$ for the current observable state o and belief for the objective θ. Thus, the agent can decide their best action using them.

$$a^*_A = \underset{a_A \in \mathcal{A}_A}{\operatorname{argmax}}\, Q(o,\theta,a_A) \tag{5.4}$$

b is updated on each action of a human and an agent as follows for each unobservable factor of belief $b(\theta)$:

$$b(\theta) \propto b(\theta)\pi_H(a_H|(o,\theta),a_A) \tag{5.5}$$

The initial belief is Uniform(θ) for the supportive and implicit guidance agents and $\mathbb{I}(\theta = \theta^*)$ for the explicit guidance agent.

5.3 Experiment

We conducted a participant experiment to investigate the advantages of the implicit guidance agent. This experiment was approved by the ethics committee of the National Institute of Informatics.

5.3.1 Collaborative Task Setting

The collaborative task setting for our experiment was a pursuit-evasion problem [62], which is a typical type of problem used for human-agent collaboration [60, 63]. This problem covers the basic factors of collaborative problems, that is, that the human and agent move in parallel and need to communicate to achieve the task. This is why we felt it would be a good base for understanding human cognition.

Figure 5.5 shows an example of our experimental scenario. There are multiple types of object in a maze. The yellow square object labeled "P" is an object that the participant can move, the red square object labeled "A" is an object that the agent can move, and the blue circle objects are target objects that the participant has to capture. When participants move their object, the target objects move, and the agent moves. Target objects move to avoid being captured, and the participant and the agent know that. However, the specific algorithm of the target objects is known only by the agent. Since both the participants and the target objects have the same opportunities for movement, participants cannot capture any target objects by themselves. This means they have to approach the target objects from both sides through collaboration with the agent, and the participant and the agent cannot move to points through which they have already passed. For this collaboration, the human and the agent should share with each other early on which object they want to capture. In the experiment, there were two target objects located in different passages. The number of steps needed to capture each object was different, but this is hard for humans to judge. Thus, the task will be more successful if the agent shows the participant which target object is the best. In the example in the figure, the lower passage is shorter than the upper

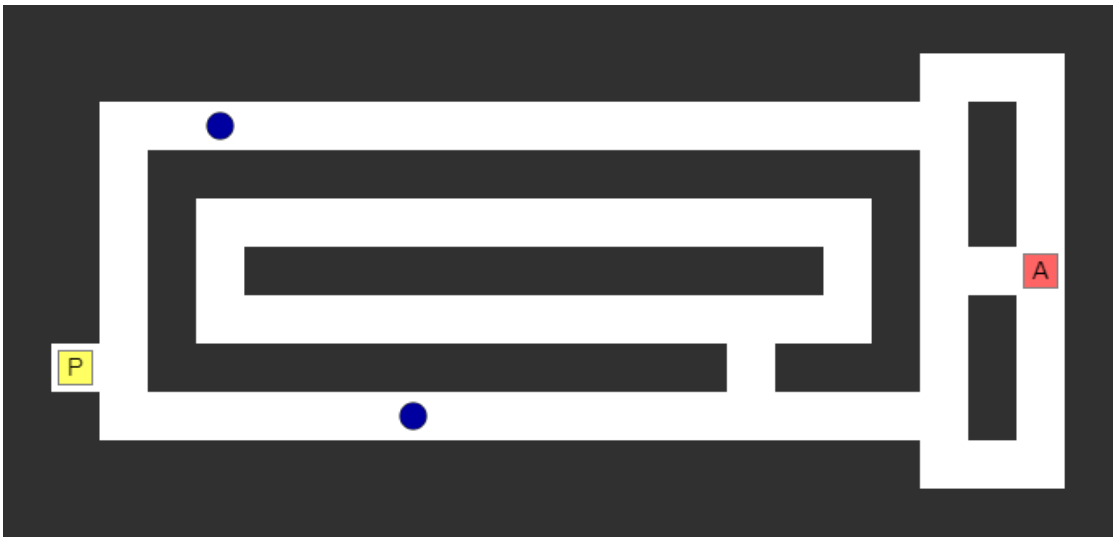

FIGURE 5.5: Example of experiment

one, but it has a path for escape. Whether the lower object can reach the path before the agent can capture it is the key information for judging which object should be aimed for. This is difficult for humans to determine instantly but easy for agents. There are three potential paths to take from the start point of the agent. The center one is the shortest for each object, and the others are detours for implicit guidance. Also, to enhance the effect of the guidance, participants and agents were prohibited from going backward.

Model

We modeled the task as a collaborative task formulation. The action space corresponded to an action for the agent, and the observable state corresponded to the positions of the participant, agent, and target objects. The reward parameter was conditioned on the target object that the human aims for. The space of the parameter corresponded to the number of target objects, that is, $|\Theta| = 2$. The reward for capturing a correct/wrong object for θ was 100, –100, and the cost of a one-step action was –1. Since a go-back action was forbidden, we could compress multiple steps into one action for a human or an agent to reach any junction. Thus, the final action space was a compressed action sequence, and the cost was $-1 \times$ the number of compressed steps. Furthermore, to prohibit invalid actions such as heading to a wall, we assigned such actions a –1000 reward. We modeled three types of collaborative agent, as discussed above: a supportive agent, an explicit

guidance agent, and an implicit guidance agent. We set the rational parameters as $\beta_1 = 1.0, \beta_2 = 5.0$, and the discount rate was 0.99.

Hypothesis

The purpose of this experiment was to determine whether implicit guidance can guide humans while allowing them to maintain autonomy. Thus, we tested the following two hypotheses.

- (H1) Implicit guidance can guide humans' decisions toward better collaboration.
- (H2) Implicit guidance can help humans maintain autonomy more than explicit guidance can.

Tasks

We prepared five tasks. Two of these tasks, as listed below, were tricks to make it hard for the participants to judge which would be the best target. All tasks are shown in the supplementary material.

- (A) There were two winding passages with different but similar lengths. There were three tasks for this type.
- (B) As shown in Fig. 5.5, there was a long passage and a short one with a path to escape. There were two tasks for this type.

Participants

We recruited participants for this study from Yahoo! Crowdsourcing. The participants were 100 adults located in Japan (70 male, 24 female, 6 unknown). The mean age of participants who answered the questionnaire we administered was 45 years.

Procedure of experiment

Our experiment was based on a within-subject design and conducted on the Web using a browser application we created. Participants were instructed on the rules of the agent behavior and then underwent a confirmation test to determine their degree of understanding. Participants who were judged to not have understood the rules were given the instructions again. After passing this test, they entered the actual experiment phase. In this phase, they were shown the environment and asked "where do you want to go?" After inputting their desired action, both the agent and the target objects moved forward one step. This process was continued until the participants either reached a target object or input a certain number of steps. When each task was finished, participants moved on to the next one. In total, they were shown 17 tasks, which consisted of 15 regular tasks and 2 dummy tasks to check whether they understood the instructions. Regular tasks consisted of three task sets (corresponding to the three collaborative agents) that included five tasks each (corresponding to the variations of tasks). The order of the sets and the order of the tasks within each set were randomized for each participant. After participants finished each set, we gave them a survey on the perceived interaction with the agent (algorithm) using a 7-point Likert scale.

The survey consisted of the questions listed below.

1. Was it easy to collaborate with this agent?
2. Did you feel that you had the initiative when working with this agent?
3. Could you find the target object of this agent easily?
4. Did you feel that this agent inferred your intention?

Item 2 was the main question, as it relates to the perceived autonomy we wanted to confirm. The additional items were to prevent biased answers and relate to other important variables for human-agent (robot) interaction. Item 1 is related to the perceived ease of collaboration, namely, the fluency of the collaboration, which has become an important qualitative variable in research on human-robot interaction in recent years [64]. Item 3 is related to the perceived inference of the agent's intentions by the human. It is one of the variables focused on the transparency of the agent, which plays a key role in constructing human trust in an

agent [65]. From the concrete algorithm perspective, a higher score is expected for guidance agents (especially explicit guidance agents) than for supportive agents. Item 4 is related to the perceived inference of the human's intentions by the agent. This is a key element of the perceived working alliance [64], and when it is functioning smoothly, it increases the perceived adaptivity in human-agent interaction. Perceived adaptivity has a positive effect on perceived usefulness and perceived enjoyment [66]. From the concrete algorithm perspective, a higher score is expected for agents without implicit guidance (especially supportive agents) than for implicit guidance agents.

5.4 Results

Before analyzing the results, we excluded any data of participants who were invalidated. We used dummy tasks for this purpose, which were simple tasks that had only one valid target object. We then filtered out the results of participants (a total of three) who failed these dummy tasks.

5.4.1 Results of Collaborative Task

Figure 5.6 shows the rate at which participants captured the best object that the agent knew. In other words, it is the success rate of the guidance of the agent based on any of the given [types of?] guidance. We tested the data according to the standard process for paired testing. The results of a repeated measures analysis of variance (ANOVA) showed that there was a statistically significant difference between the agent types for the overall tasks ($F(2, 968) = 79.9, p = 7.4e - 33$), task type (A) ($F(2, 580) = 55.9, p = 5.9e - 23$), and task type (B) ($F(2, 386) = 24.7, p = 7.5e - 11$). We then performed repeated measures t-tests with a Bonferroni correction to determine which two agents had a statistically significant difference. "*" in the figure means there were significant differences between the two scores ($p << 0.01$). The results show the average rate for the overall tasks, task type (A), and task type (B). All of the results were similar, which demonstrates that the performances were independent of the task type. The collaboration task with the supportive agent clearly had a low rate. This indicates that the task was difficult enough that participants found it hard to judge which object was best, and the guidance from the agent was valuable for improving

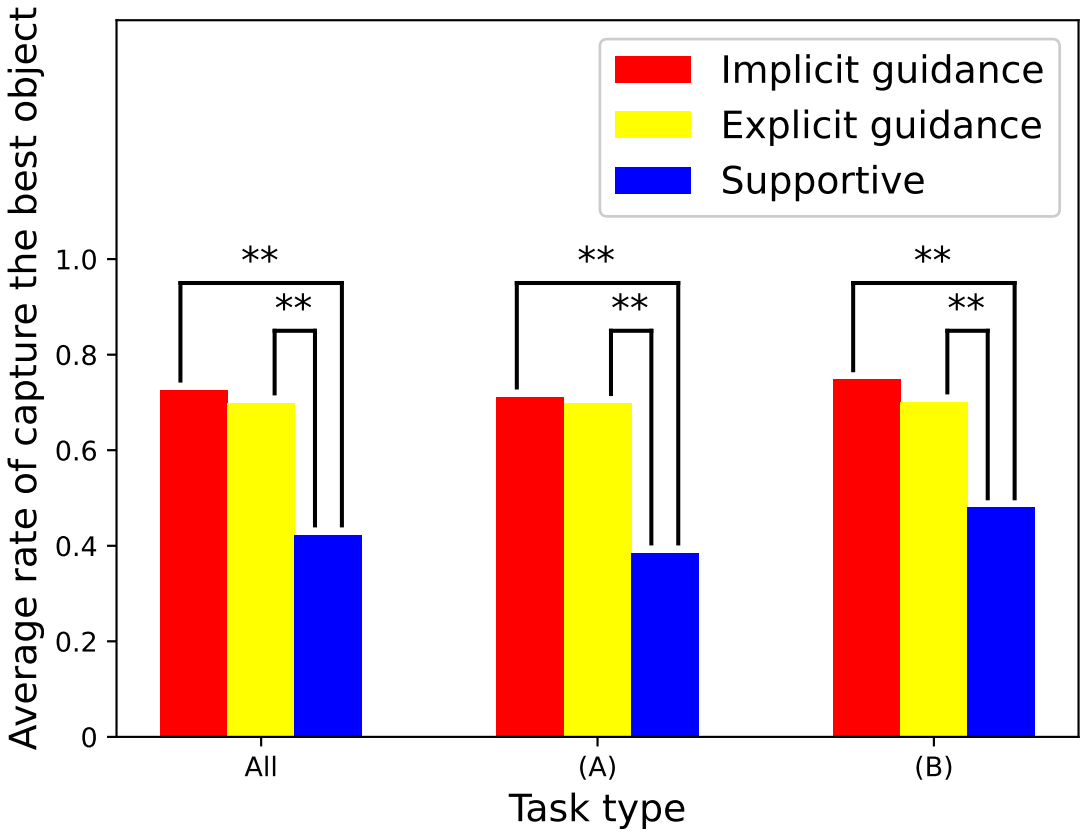

FIGURE 5.6: Average rate of capturing best object

the performance on this task. These results are strong evidence in support of hypothesis H1. As another interesting point, there was no significant difference in the rate between implicit guidance and explicit guidance. Although we did not explain implicit guidance to the participants, they inferred the agent's intention anyway and used it as guidance. Of course, the probable reason for this is that the task was so simple that participants could easily infer the agent's intentions. However, the fact that implicit guidance is almost as effective as explicit guidance in such simple tasks is quite impressive.

5.4.2 Results for Perceived Interaction with Agent

Figure 5.7 shows the results of the survey on the effect of the agent on cognition. The results of a repeated measures ANOVA showed that there was a statistically significant difference between the agent types for perceived ease of collaboration ($F(2, 192) = 29.8, p = 5.4e - 12$), perceived autonomy ($F(2, 192) = 36.4, p = 4.1e{-}14$), and perceived inference of the human's intentions ($F(2, 192) = 49.7, p =$

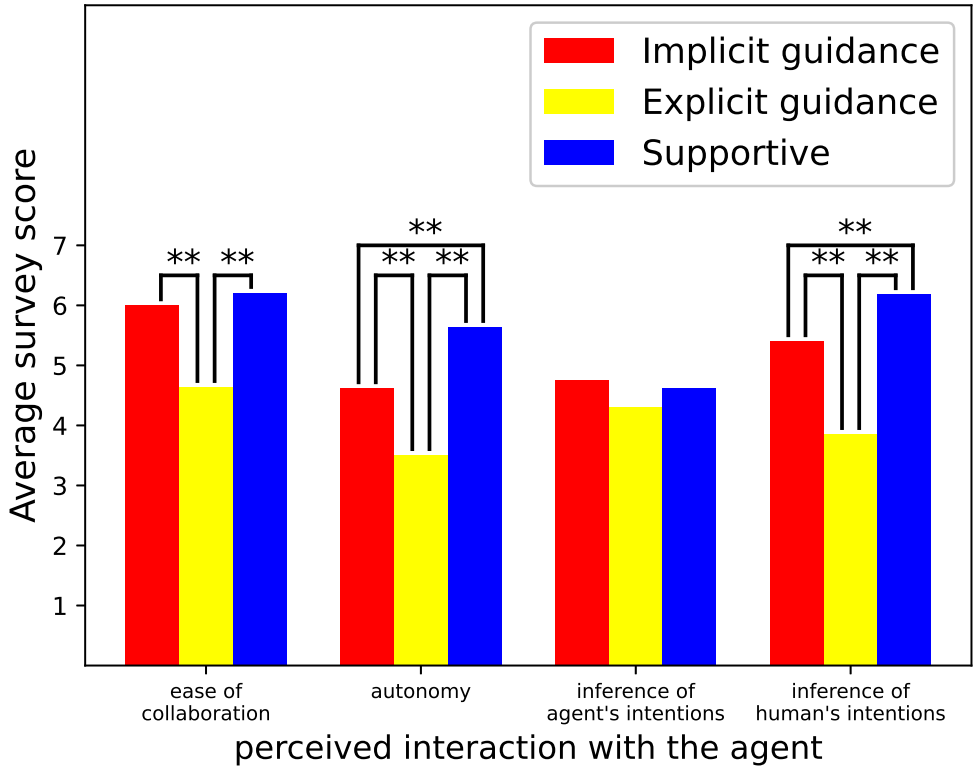

FIGURE 5.7: Average survey score for perceived interaction with agent

$4.0e-18$). In contrast, there was no statistically significant difference for the perceived inference of the agent's intentions ($F(2,192) = 1.8, p = 0.167$). We then performed repeated measures t-tests with Bonferroni correction to determine which two agents had a statistically significant difference regarding variables that had a significant difference. "*" in the figure means there were significant differences between the two scores ($p << 0.01$). The most important result here is the score for perceived autonomy. From this result, we can see that participants felt they had more autonomy during the tasks when collaborating with the implicit guidance agent than with the explicit guidance one. These results are strong evidence in support of hypothesis H2.

Although the other results do not directly concern our hypothesis, we discuss their analysis briefly. Regarding the perceived inference of the human's intentions, the results were basically as expected, but for the perceived inference of the agent's intentions, the fact that there were no significant differences among all agents was unexpected. One hypothesis that explains this is that humans do not recognize guidance information as the agent's intention. As for the perceived ease of collaboration, the results showed that explicit guidance had adverse effects on it.

Implicit guidance agents and supportive agents use exactly the same interface, though the algorithms are different, but explicit guidance agents use a slightly different interface to convey guidance, which increases the amount of information on the interface a little. We think that the burden of understanding such additional visible information might be responsible for the negative effect on the perceived ease of collaboration.

5.5 Discussion

As far as we know, this is the first study to demonstrate that implicit guidance has advantages in terms of both task performance and the effect of the agent on the perceived autonomy of a human in human-agent teams. In this section, we discuss how our results relate to other studies, current limitations, and future directions.

The results in 5.4.1 show that both implicit and explicit guidance increase the success rate of a collaborative task. We feel one reason for this is that the quality of information in the guidance is appropriate. A previous study on the relationship between information type and collaborative task performance [21] showed that "implicit coordination" improves the performance of a task more than "explicit coordination" in a cooperative task. The word "implicit" refers to coordination that "relies on anticipation of the information and resource needs of the other team members." This definition is different from ours, as implicit coordination is included in explicit guidance in our context. The study in [21] further divided implicit coordination into deliberative communication, which involves communicating objectives, and reactive communication, which involves communicating situations, and argued that high-performance teams are more likely to use the former type of communication than the latter. We feel that the quality of information in implicit guidance in our context is the same as this deliberative communication in that it conveys the desired target, which is one of the reasons the performance of our guidance can be good.

The main concern of human-agent teams is how to improve the performance on tasks. However, as there have not been many studies that focus on the effect of the agent on cognition, the results in 5.4.2 should make a good contribution to the research on human-agent teams. One of the few studies that have been done

investigated task performance and people's preference for assigning tasks in a cooperative task involving a human and an AI agent [67]; the authors mentioned the risk that a worker with a robot collaborator may perform less well due to their loss of autonomy, which is something we also examined in our work. They found that a semi-autonomous setting, in which a human first decides which tasks they want to perform and the agent then decides the rest of the task assignments, is more satisfying than manual control and autonomous control settings in which a human and robot fully assign tasks. In cooperation with the implicit guidance agent and the supportive agent in our study, the human selects the desired target object by his or herself. This can be regarded as a kind of semi-autonomous setting. Thus, our results are consistent with these ones in that the participants felt strongly that cooperation was easier than with explicit guidance agents. Furthermore, the above study also mentioned that task efficiency has a positive effect on human satisfaction, which is also consistent with our results.

This study has limitations in that the experimental environment was small and simple, the intention model was a small discrete set of target objectives, and the action space of the agent was a small discrete set. In a real-world environment, there is a wide variety of human intentions, such as target priorities and action preferences. The results in this paper do not show whether our approach is sufficiently scalable for problems with such a complex intention structure. In addition, the agent's action space was a small discrete set that can be distinguished by humans, which made it easier for the human to infer the agent's intention. This strengthens the advantage of implicit guidance, so our results do not necessarily guarantee the same advantage for environments with continuous action spaces. Extending the intention model to a more flexible structure would be the most important direction for our future study. One of the most promising approaches is integration with studies on inverse reinforcement learning [68]. Inverse reinforcement learning is the problem of estimating the reward function, which is the basis of behavior, from the behavior of others. Intention and purpose estimation based on the Bayesian theory of mind can also be regarded as a kind of inverse reinforcement learning [69]. Inverse reinforcement learning has been investigated for various reward models [55, 70–72] and has also been proposed to handle uncertainty in information on a particular reward [73]. Finally, regarding the simplicity of our experimental environment, an interesting direction for future work would be using environments that are designed according to an objective complexity factor [74] and then analyzing the relationship between the effectiveness of implicit

guidance and the complexity of the environment.

Another limitation is the assumption that all humans have the same fixed cognitive model. As mentioned earlier, a fixed cognitive model is beneficial for ad-hoc collaboration, but for more accurate collaboration, fitting to individual cognitive models is important. The first approach would be to parameterize human cognition with respect to specific cognitive abilities (rationality, K-level reasoning [75], working memory capacity [76], etc.) and to fit the parameters online. This would enable the personalization of cognitive models with a small number of samples. One such approach is human-robot mutual adaptation for "shared autonomy," in which control of a robot is shared between the human and the robot [77]. In that approach, the robot learns "adaptability," which is the degree to which humans change their policies to accommodate a robot's control.

Finally, the survey items we used to determine the effect of the agent on perceived autonomy were general and subjective. For a more specific and consistent analysis of the effect on perceived autonomy, we need to develop more sophisticated survey items and additional objective variables. Multiple consistent questions to determine human autonomy in shared autonomy have been used before [10]. As for measuring an objective variable, an analysis of the trajectories in a collaborative task would be the first choice. A good clue for the perceived autonomy in trajectories is "shuffles." Originally, shuffles referred to any action that negates the previous action, such as moving left and then right, and it can also be an objective variable for human confusion. If we combine shuffles with goal estimation, we can design a "shuffles for goal" variable. A larger number of variables means that a human's goal is not consistent, which would thus imply that he or she is affected by others and has low autonomy. In addition, reaction time and biometric information such as gaze might also be good candidates for objective variables.

Another limitation of this study is that we assumed that humans regard the agent to be rational. One approach to solving this is to use the Bayesian theory of mind model for irrational agents [78].

Chapter 6

Extension of Planning with Implicit Guidance for Complex Tasks

This chapter explains our second and third studies. Section 6.1 is about our second study, which is the extension of the model for theory of mind. Section 6.2 is about our third study, which is the extension of collaborative planning.

6.1 Extension of Theory of Mind for Complex Situations

Bayesian theory of mind, described in Sec. 3.4.2, can capture human perception in various situations. However, this approach may differ from human perception in some complex situations. In this section, we show an example of such a situation and an extension of Bayesian theory of mind to capture human perception better in the situation.

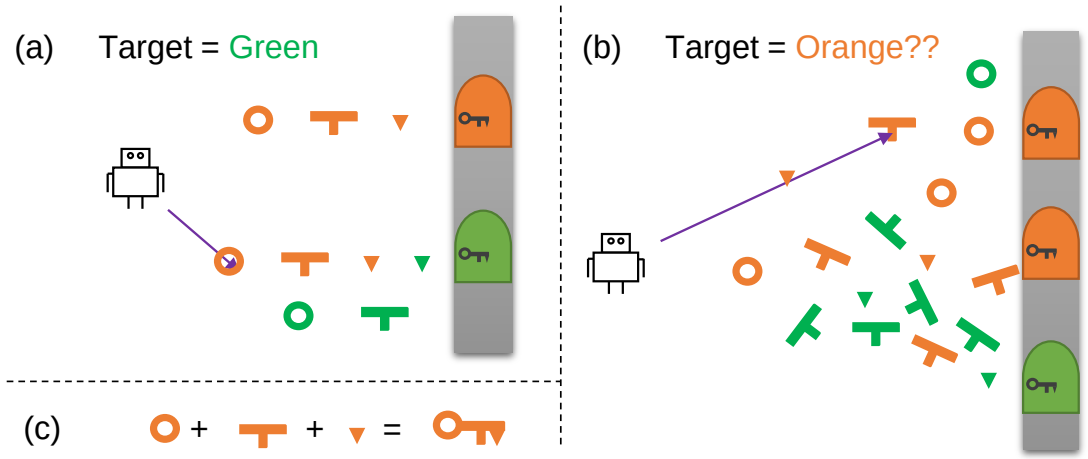

FIGURE 6.1: Complex environment in which Bayesian inverse planning is mistaken for model human inference

6.1.1 Example of Complex Situation Settings

Figure 6.1 is one example of a situation in which the existing Bayesian Theory of Mind is not suited to capturing human perception.

This difference is due to the fact that observers cannot completely recognize the rationality of actors when a problem is complicated. In the above example, there are many plans to achieve one goal, but it is difficult to evaluate all plans. For this reason, human inference is affected by a bias towards "easy to predict" plans. This means that humans only consider a few plans in which they can easily predict the plan. This is an example of bounded rationality [3].

To capture human bounded rationality, we modify the existing Bayesian theory of mind model to implement the bias to increase the probability that a plan can be easily predicted from observed actions. We call the model the plan predictability oriented model, and we call the existing model the full inverse planning model.

6.1.2 Notation and Problem Setting

We denote a set of people's goals as $\mathcal{G}$ and a set of actions as $\mathcal{A}$. Action sequences are represented as $\mathbf{a} \in \mathcal{A}^+$, and a set of all plans to achieve goal $g \in \mathcal{G}$ is represented as $\mathcal{P}_g$. Since plans are a kind of sequence, $\mathcal{P}_g \subset \mathcal{A}^+$

This objective is a posterior probability given an observed action sequence the same as Bayesian theory of mind. The difference from Eq. 3.24 is that action changes into an action sequence.

$$P(g|\mathbf{a}) \tag{6.1}$$

6.1.3 Calculation of Full Inverse Planning Model

Using Eq. 3.25, the objective can be calculated as follows.

$$P(g|\mathbf{a}) \propto P(\mathbf{a}|g)P(g) \tag{6.2}$$

We assume no prior knowledge about the other person's goal. In other words, we assume $P(g)$ as a uniform distribution. Therefore, we can ignore $P(g)$ from Eq. 6.1. To calculate this, we assume that a human makes a plan in advance and executes their actions according to the plan. Thus, we can factorize $P(\mathbf{a}|g)$ into the probability of a plan to be considered achieving g and the probability of executing $\mathbf{a}$ under the plan. The following equation is obtained when summarizing this factored probability for all available plans.

$$P(\mathbf{a}|g) \propto \sum_{p \in \mathcal{P}_g} P(\mathbf{a}|p)P(p|g) \tag{6.3}$$

$P(\mathbf{a}|p)$ and $P(p|g)$ are according to Boltzmann noisy rationality.

$$\begin{aligned} P(\mathbf{a}|p) &= \frac{\exp\left(\beta Q_{p,g}(\mathbf{a})\right)}{\sum_{p' \in \mathcal{P}_g} \exp\left(\beta Q_{p',g}(\mathbf{a})\right)} \\ P(p|g) &= \frac{\exp\left(\beta Q_g(p)\right)}{\sum_{g' \in \mathcal{G}} \exp\left(\beta Q_{g'}(p)\right)} \end{aligned} \tag{6.4}$$

$Q_g(\mathbf{a}), Q_{p,g}(\mathbf{a})$ corresponds to the action value function given g and g, p respectively. Note, p is plan to achieve specific goal. In other words, if plan decided, the corresponding goal also comes uniquely. thus we can deal $Q_{p,g}(\mathbf{a})$ as $Q_p(\mathbf{a})$

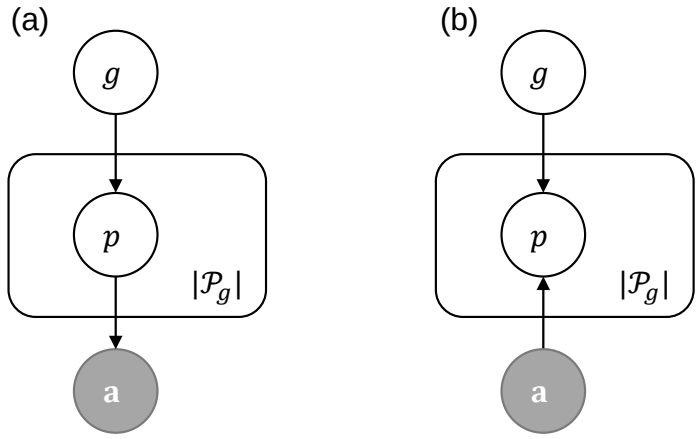

FIGURE 6.2: Graphical model of $P(\mathbf{a}|g)$ for (a) Full Inverse Planning Model and (b) Plan Predictability Oriented Model.

6.1.4 Calculation of Plan Predictability Oriented Model

In Plan Predictability Oriented Model, Eq. 6.1 is same, however the way to calculate of $P(\mathbf{a}|g)$ is different to integrate a bias that people prefer predicable plan. Predictability of plans is the probability of plan with observing action sequences, thus that is $p(p|\mathbf{a})$. We use this instead of $P(\mathbf{a}|p)$, thus we earn the following equation.

$$P(\mathbf{a}|g) \propto \sum_{p \in \mathcal{P}_g} P(p|\mathbf{a}) P(p|g) \tag{6.5}$$

Here, We use simple Boltzmann Noisy Rationality for $P(p|\mathbf{a})$. That is

$$P(p|\mathbf{a}) = \frac{\exp\left(\beta Q_p(\mathbf{a})\right)}{\sum_{\mathbf{a} \in \mathcal{A}^+} \exp\left(\beta Q_p(\mathbf{a})\right)} \tag{6.6}$$

Figure 6.2 shows the difference of a graphical model of the likelihood of the action sequence given by goal $P(\mathbf{a}|g)$. In Full Inverse Planning Model, we use the forward generative process for full rational humans to calculate the likelihood as shown as showing Figure 6.2 (a). In the model, humans decide their plan in advance, and then humans do actions according to plans. On the other hand, as Figure 6.2 (b),

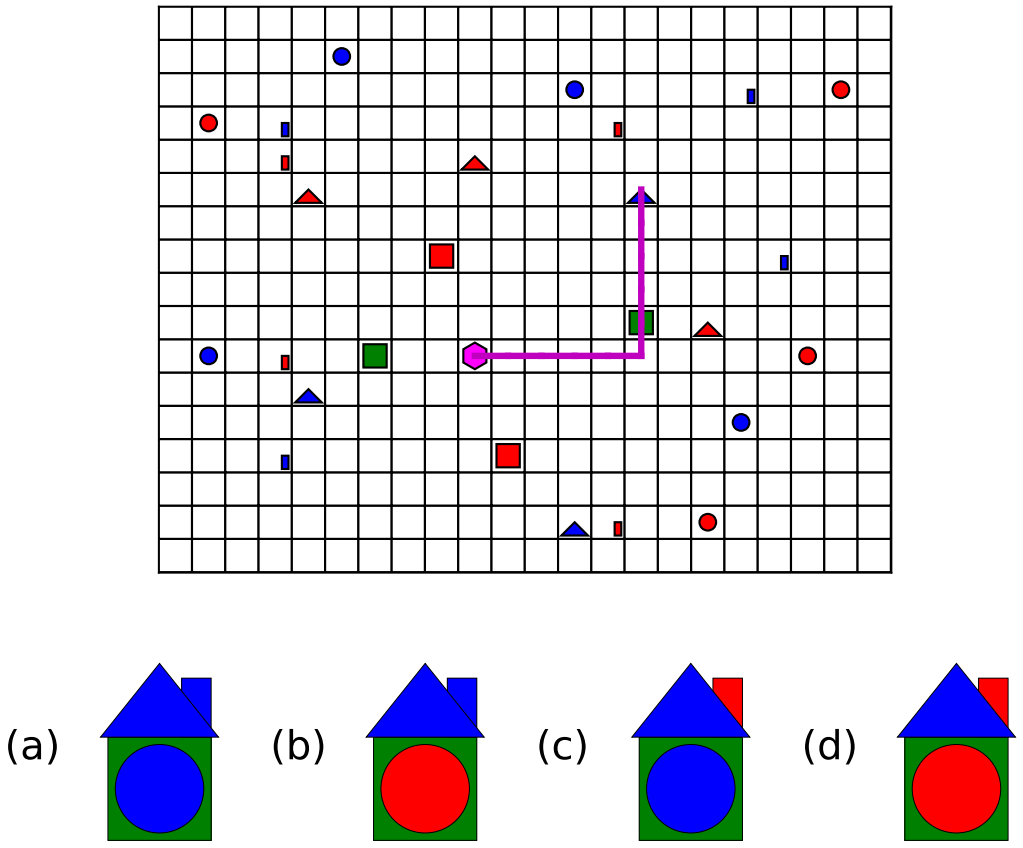

FIGURE 6.3: Example of "item creating" scenarios (for task (4, 2, 2)).

in Plan Predictability Oriented Model, there is a dependency from action sequence to plans.

We assume that humans cannot calculate others' goals on the basis of the full inverse planning model because the model needs all plans of $\mathcal{P}_g$ and it is almost impossible to enumerate all plans for humans in a complex situation. Humans consider several plans to estimate the goals of others and tend to consider plans that they can predict easily. That is the reason for the dependency of the observed action sequence into plans.

6.1.5 Experiment

To compare the full inverse planning model and the plan predictability oriented model for human cognition, we did subject experiments. For these experiments, we considered "item creating" scenarios.

Full Inverse Planning Model	0.765 ($p \ll 0.0001$)
Plan Predictability Oriented Model	0.916 ($p \ll 0.0001$)

TABLE 6.1: Pearson correlation between full Bayesian model and plan prediction oriented model with human inferences

In Fig. 6.3, the upper figure is an example of an item creating scenario. The environment in this scenario is one kind of grid world. There is one agent and several parts for items in either grid, and there is not more than one part in the same grid. Here, the agent is represented by a purple hexagon. There are four types of parts (square, triangle, small rectangle, circle), and there are two to three colors for each type.

The goal of the agent is to create a "goal product" that it wants to create. The goal product consists of two to four types of parts with only one used for each type: (square, triangle), (square, triangle, small rectangle), and (square, triangle, small rectangle, circle). The agent moves to collect the parts that are necessary for the agent's own "goal product." The agent has a priority to collect the items. The agent collects the parts in the order of square, triangle, small rectangle, and circle. There are multiple objects of the same color and the same type in the environment; thus, there is a more than one combination of objects for generating one object.

Participant

We recruited participants for this study using Yahoo! Crowdsourcing. Valid participants were 47 adults located in Japan (13 male, 29 female, 5 unknown). The mean age was 39 years old.

Procedure of experiment

Experiments were conducted on the Web via browser application we made. Subjects were instructed the rule of agent behavior then underwent a confirmation test to check their degree of understanding. In this test, participants judged to not understand the rules were given the instructions again. The participants who passed the confirmation test entered the actual experiment phase. In the actual experiment phase, participants saw the environment, part of the agent's movement

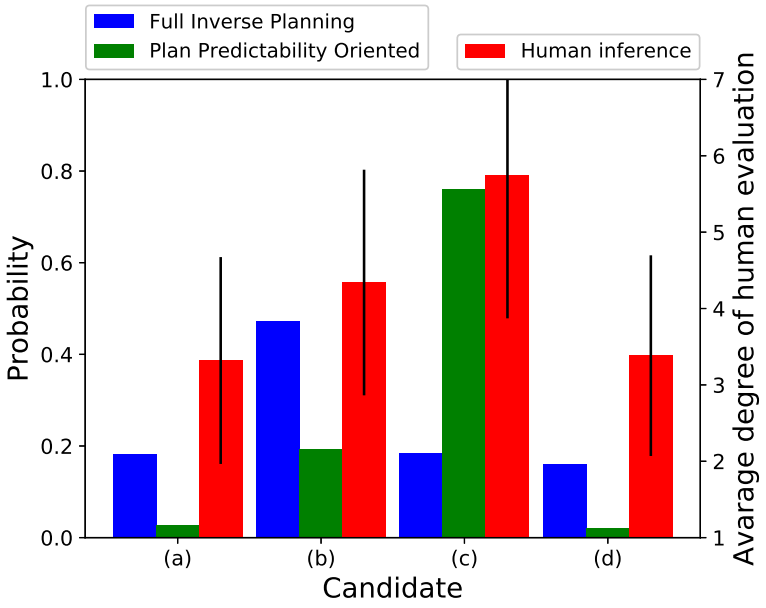

FIGURE 6.4: Inference of human and computational models for task (4, 2, 2)

Full Inverse Planning Model	-0.714 ($p = 0.03$)
Plan Predictability Oriented Model	0.116 ($p = 0.76$)

TABLE 6.2: Pearson correlation of "Pearson correlation of human inference with each models" with task complexity factor $k - n$

path collecting parts, and four target candidates for the agent's "goal product" simultaneously. The subjects selected one that they considered most likely to be the agent's "goal product" from the candidates. Also, participants scored the degree of likelihood that they estimate the candidate is the agent's "goal product" for all the candidates. We adopted a seven-degree score for the evaluation.

Before analyzing the participants' results, we excluded the results of the participants who were invalidated. We defined invalid participants as those who (a) gave the same evaluation score to all the candidates or who (b) did not give the highest evaluation score to the selected candidate as the most likely candidate.

Stimuli

We prepared nine stimuli (tasks) with different task complexities. There were three variables that affected the task complexity. k was the number of types of parts included in the agent's goal product, n was the number of types of parts included in the agent's path, and c was the number of colors of parts that were

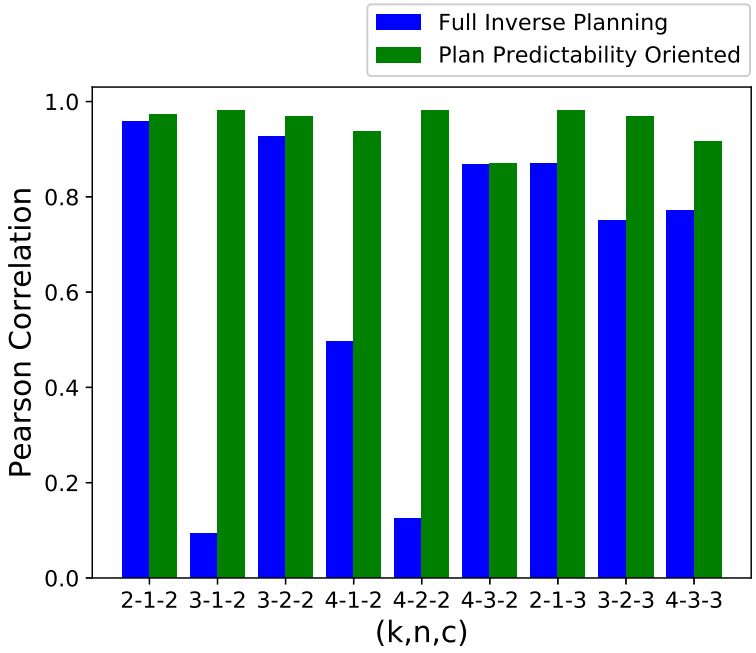

FIGURE 6.5: Pearson correlation between full inverse planning model and plan predictability oriented model with human inferences for each task

not [corrected — collected?] by the agent yet. We designed nine combinations of complexity for variables k, n, and c. There were (2, 1, 2), (3, 1, 2), (3, 2, 2), (4, 1, 2), (4, 2, 2), (4, 3, 2), (2, 1, 3), (3, 2, 3), and (4, 3, 3). In addition, we made a stimulus corresponding to each combination. In task $c = 3$, the only kind of part that was not yet collected by an agent had three colors. The example in Fig. 6.3 is task (4, 2, 2), and the purple line in the upper figure corresponds to the movement of the agent.

We designed the placement of parts within the environment and the agent's path within the task to differ in terms of the inference of the most likely goal product with the full inverse planning model and plan predictability oriented model. We chose four candidates according to four policies: (1) the most likely candidate for the full inverse planning model, (2) the most likely candidate for the plan predictability oriented model, and (3, 4) the candidate that had a low probability for both models. The lower figures in Fig. 6.3 are examples of candidates for the agent's goal product in task (4, 2, 2).

Model

In the item creating scenario, $\mathcal{G}$ corresponds to a set of goal products, and $\mathcal{A}$ corresponds to a set of individual parts. $\mathcal{P}_g$ is a set of all available combinations of parts to build a goal product g. Since there is more than one combination of objects for generating one object, $\forall g, |\mathcal{P}_g| > 1$. Here, we defined $Q_g(p)$ as $-cost(p)$. $cost(p)$ is the shortest path length of p. $Q_p(\mathbf{a})$ is $-cost(p - \mathbf{a})$. $p - \mathbf{a}$ means a remaining plan of p after $\mathbf{a}$. We set rational parameters $\beta_1 = 0.3$, $\beta_2 = 0.3$, and $\beta_3 = 0.4$.

6.1.6 Result

We evaluated the two models in a comparative experiment by comparing participant scores. For comparison, we made a human score vector and model probability vector. The human score vector consisted of the participants' scores serialized over all results (thus, the length of the vector was 36) without any normalization. We used the average vector of human score vectors for all valid participants. To make the model probability vector, we extracted the probabilities of candidate for each task and serialized them (the length of the vector was also 36). Table 6.1 is a Pearson correlation of the averaged human score vector between model probability vectors for the two models. The results show that the plan predictability oriented model had a much better correlation with human inference. Figure 6.4 is specific result for task (4, 2, 2). Blue and green are the probability calculated by each computational model, and the red bar is the average of the participants' score. This figure also shows a good correlation of human inference with the plan predictability oriented model. We also executed a significance test. First, we calculated the Pearson correlation between human score vectors and model probability vectors for all valid participants. Thus, we obtained two sets of Pearson correlations for the two models. Then, we executed t-tests on the sets. The p-value was 0.03 (< 0.05). Thus, we confirmed that there was a significant difference in the correlation with the two models.

Relation of task complexity

Next, we calculated the Pearson correlation of human inference with both models for each task. We averaged the human score and model probability vectors for

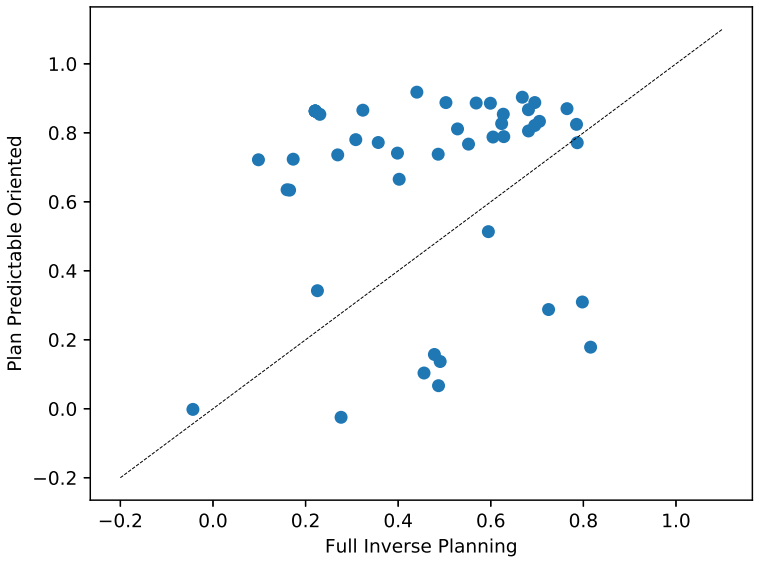

FIGURE 6.6: Scatter plot of Pearson correlation between full inverse planning model and plan predictability oriented model with human inference for each participant

each task and calculated the Pearson correlation using these vectors. Figure 6.5 is the result. First, the results show that the plan predictability oriented model had a much better correlation with human inference for all the tasks. The full inverse planning model had a low correlation with human inference in tasks (3-1-2), (4-1-2), and (4-2-2) in particular. The common factor in these tasks was that the remaining number of kinds of parts, which is represented as $k - n$, was more than one.

Table 6.2 is the Pearson correlation of human inference with each model with $k - n$. In other words, it is the correlation between the values in Fig. 6.5 and $k - n$. The full inverse planning model had a strongly negative correlation with $k-n$. $k-n$ is strongly related to the future available paths of an agent. This means that this model is not effective for tasks that have many future available paths. This matches with the intuition that humans may think bounded-rationally, not full-rationally, in complex situations. The plan predictability oriented model did not have such negative correlation with $k - n$. This means that this model is not affected by task complexity.

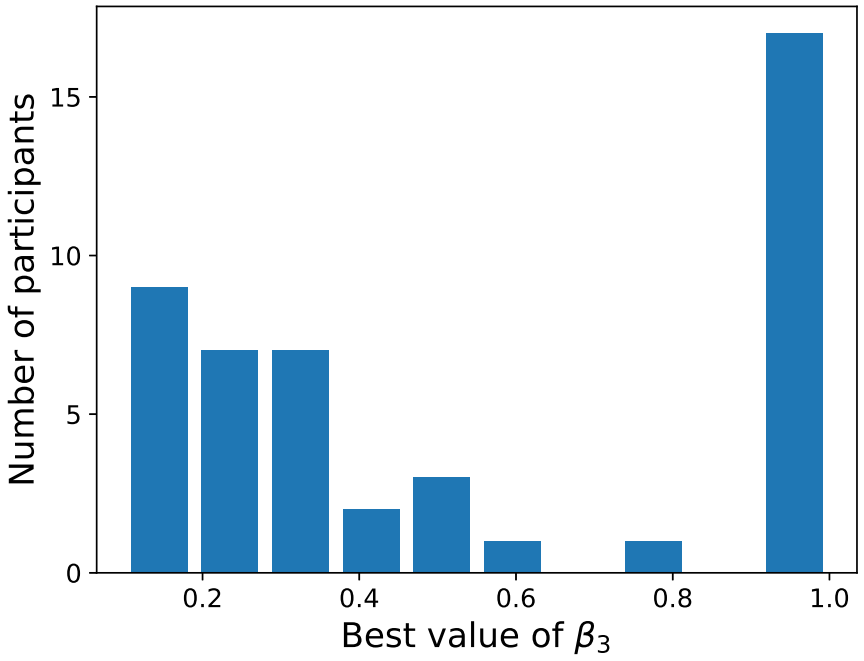

FIGURE 6.7: Histograms for number of participants for each best predictability bias

Confirmation of individual differences

We calculated the Pearson correlation of human inference with both models for each participant. Figure 6.6 is a scatter plot of the results. The dotted line on the plot shows the boundary where the correlation between both models was equal. Most of participants are in the upper left of the plot. This means that most of the participants' inferences had a good correlation with the plan predictability oriented model. Some participants are in the lower right, which means that this model could not model these participants. These participants could rationally recognize other people's intentions completely in this experiment, so they had a good correlation with the full inverse planning model. These results suggest that there are individual differences in peoples' bounded rationality.

We made multiple plan predictability oriented models that had different β_3. β_3 is the parameter for the plan predictability bias. The range of β_3 was from 0.0 to 1.0 increments 0.1. Figure 6.7 shows histograms of the number of participants who had the best correlation with the model with β_3. It shows that many participants had a strong bias, but some had a small bias or no bias. Table 6.3 is the average Pearson correlation of each participant's result with the models. Here, (same)

Full Inverse Planning Model	0.513
Plan Predictability Oriented Model (same)	0.638
Plan Predictability Oriented Model (individual)	0.738

TABLE 6.3: Average Pearson correlation of human inference between models for each participant

means we used the same value as β_3, and (individual) means we used the best value as β_3 for individual users. The individual setting had a higher correlation with humans than the same correlation. This suggests that the adaptation of the bias can improve our model.

6.1.7 Discussion

Essentially, the full inverse planning model differs from human cognition in situations in which there is a difference in the rationality of the actor and the observer. Since forward planning to take action towards a particular goal is generally easier than inverse planning to infer causes from observed actions, these situations might always happen. Therefore, we think that the model is useful for many situations. Additionally, our computational model is based on the inverse planning model and simple plan prediction, so our model has the potential to adapt to various situations. However, we showed the actual effectiveness only under an item creating scenario. This scenario is one example environment that has a grid geometric rationality and sequential planning. Explicitly sequential planning is the most basic planning, and a lot of human planning is based on sequential planning. Grid geometric rationality and sequential planning are often used for theory of mind [54]. Therefore, we think that our model can be used in a broad range of situations.

Understanding how human infer others' intentions is useful for considering good actions for collaboration with others. In the cognitive science area, there is research on how humans behave when they want to communicate their goal or purpose [28]. In the artificial intelligence and robotics areas, research on collaborative planning is more popular and important. For example, "legibility" was proposed [79]. This is a measure of the human expectation toward robots' intentions or goals based on their behavior. Additionally, there are works on using legibility for planning [80].

The expansion of our model for larger and more complex tasks would a very interesting direction for our future work. Introducing hierarchical planning is

a promising approach. Our model can be considered one type of hierarchical modeling in which the inference of plans is regarded as an intermediate layer. The hierarchical predictive coding framework [81] is one example of a hierarchical model for human cognition. This model has multiple inference layers with different abstraction levels, and it executes step-by-step inference using MAP estimation. Similarly, our model can be expanded with multiple planning layers. Checking whether such a model would be better for modeling human cognition would be an interesting next research step.

A deeper analysis of individual rationality would also be interesting. We just demonstrated that human rationality differs from person to person. However, there might be some factors that decide the degree of bias. Seeking such factors and improving our model by implementing them would be a very valuable study.

6.2 Extension of Planning Algorithm

In a realistic problem, a human might have their goal or preference for action, which is hidden from the agent. In this case, the agent lacks the information to calculate the best plan. Thus, the agent has to infer information from the human from the behavior of the human. We extend the human-agent teaming problems described in Chapter 4 by implementing the information.

From a different point of view, human-agent teaming is a problem in which the human lacks information on the reward. Therefore, we can model the extended human-agent teaming problem as a problem in which the agent and human have their own reward information. To calculate the best plan, they have to know each other's informatio

Furthermore, we add one assumption for human behavior. In the setting described in Chapter 5, we assumed that humans act to achieve their current likely goal. However, in realistic situations, humans tend to act to share their action for better collaboration [47, 80]. This is the same concept as our implicit guidance. Thus, we assume that humans also decide their action in consideration of sharing their information through their behavior. All assumptions considered, human-agent teaming can be formalized as the human and the agent exchanging their information with each other through their behavior.

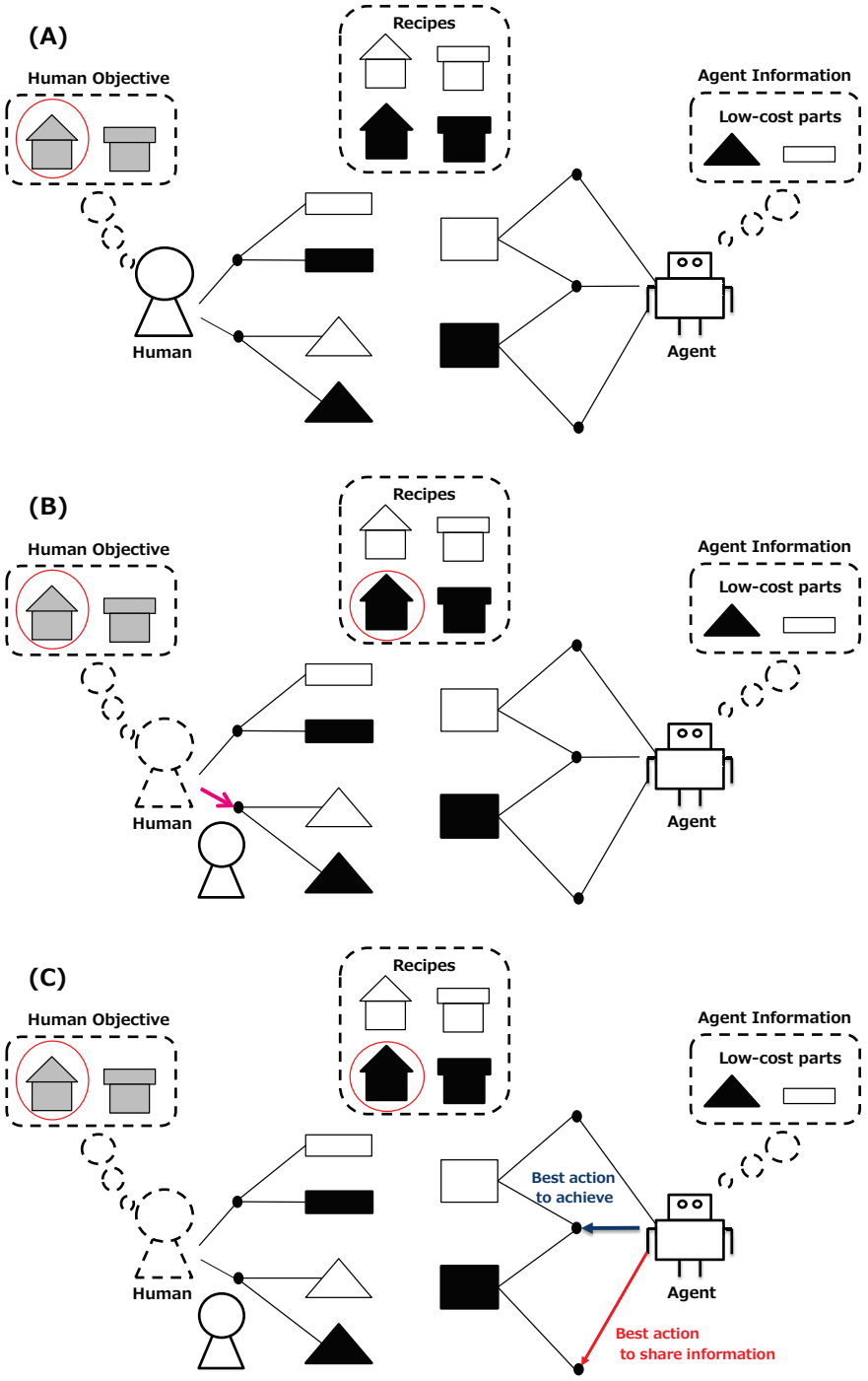

FIGURE 6.8: Example of extended human-agent teaming scenario

Figure 6.8 (A) is an example of the extended human-agent teaming scenario. This task is to gather parts to build one object from among recipes. The human has an intended object shape, and the agent does not have this information. Instead, the agent has information on which parts are low in cost. Figure 6.8 (B) is a situation in which the human moves down to make their object, at which point

the agent discovers the human's objective and chooses the best recipe. Figure 6.8 (C) shows the option for the agent to take action. If the agent considers only their own performance, it should choose the blue path because it is the shortest path to get the required parts. However, if the agent chooses this path, the human cannot determine which parts the agent wants to get, and this may cause the human to get the wrong parts for the best recipe. To avoid this, the agent should take the red path in order to reveal the information that the agents want to get in the black parts.

6.2.1 Updated Problem Setting

We extended the CIRL model for our human-agent teaming problem by adding a factor for specific reward information to the agent. Formally, the human-agent teaming problem is represented as a tuple $< \mathcal{X}, \{\mathcal{A}_H, \mathcal{A}_A\}, T, \{\Theta_H, \Theta_A, R\}, \gamma >$. Θ_A is a set of reward parameters observable only by the agent, and the reward function considers these parameters. Formally, this is $R(x, a_H, a_A, \theta_H, \theta_A)$. Other notations are the same as CIRL.

6.2.2 Bellman update for Human-Agent Team

We modify the Bellman update of CIRL in Sec. 3.3.2 further. In our problem, the human cannot access the reward parameters of the agent, so the human cannot plan their best actions. Therefore, we assume that the human infers the agent's intention from the agent's action and uses the inference to calculate their best action. We introduce such a human inference function into calculating a value for the human. The value is updated as follows.

$$Q_H(x, a_H, \sigma, \theta_H) = \sum_{\theta_A \in \Theta_A} p(\theta_A | x, a_H, \sigma, \theta_H) Q_H(x, a_H, \sigma, \theta_H, \theta_A) \tag{6.7}$$

$Q_H(x, a_H, \sigma, \theta_H, \theta_A))$ is the value with fixing agent's reward parameter with θ_A. In other words, it is the value when the human knows the agent's preference is θ^A. This is the same problem as CIRL, so we can calculate it for all $theta_A$. This value is calculated by using the Q-value and the Bellman update we describe in Sec. 3.3.2.

$p(\theta_A | x, a_H, \sigma, \theta_H)$ is the human's inference function.

We can use any probability function in which the probability correlates with the value as the human's inference function. First, we use the function based on inverse planning in Sec. 3.4. We first assume that the human infers the agent's reward parameter using only the first action in a conditional plan; then,

$$p(\theta_A|x, a_H, \sigma, \theta_H) \quad \propto \quad p(a_A|x, \theta_A, a_H, \theta_H) \tag{6.8}$$

To calculate $p(a_A|x, \theta_A, a_H, \theta_H)$, the human needs to know the values of the agent's plan; however, the plan needs the agent's belief for the human's reward parameter. This causes recursive calculation infinitely; thus, we implement the optimistic assumption that a human assumes that an agent can infer the human's reward parameter perfectly in one step. Under this assumption, the human thinks that the agent know human's reward parameter at the next time step. In summary, the probability of the agent's action is calculated as follows.

$$p(a_A|x, \theta_A, a_H, \theta_H) = \frac{\exp(\beta Q(b^*, x, a^H, \theta_A)}{\sum_{a_A \in \mathcal{A}_A} \exp(\beta Q(b^*, x, a^H, \theta_A)} \tag{6.9}$$

$$b^*(a_H) = \mathbb{1}(\theta_H = \theta'_H) \quad \forall \theta'_H \in \Theta_H \tag{6.10}$$

Note that the value function has additional dependance about the agent's reward parameter θ_A.

The basic process is like POMDP, and the additional computational cost is times $|\Theta^H|$ to calculate the max action. The total computational complexity is $O(|\mathcal{X}||\Theta^A||\Theta^H||\mathcal{A}^{\mathcal{A}}||V'||\mathcal{A}^{\mathcal{H}}||B|)$.

6.2.3 Applying Predictability Bias

As mentioned in Section 6.1 we can change the human inference function to other probability distributions. We use an inference function inspired by plan predictability bias as an alternative human inference function. The core idea is omitting one Bayesian inference layer and considering the value only of the action observed. In our problems, the layer to remove is the upper equation in Eq. 6.8. Thus, we can obtain a human inference function based on predictability bias for our problems.

$$p(a_A|x, \theta_A, a_H, \theta_H) = \frac{\exp(\beta Q(b^*, x, a^H, \theta_A)}{\sum_{\theta_A \in \Theta_A} \exp(\beta Q(b^*, x, a^H, \theta_A)} \tag{6.11}$$

6.2.4 Experimental

Scenario

To evaluate our method for a human-agent team with human cognition, we executed experiments with participants. For these experiments, we considered the "human rescue" scenario. This scenario was a metaphor for a situation in a disaster. The situation assumed that human-agent teaming was based on implicit guidance since it is difficult to give fixed and explicit guidance in such situations.

Figure 6.9 is an example of the scenario. There are two fields represented as graphs. The participant is in the left field, and the autonomous agent is in the other. The lines represent roads, and the gray nodes represent step points along the roads. The objects in the bottom half (yellow and pink) indicate participants and agents. The objects in the upper side indicate victims. There are two types of victims (blue and red). The objects on the lines indicate pollution on the road. There are two types of pollution (brown and green).

The purpose of this scenario was for the human and agent to rescue as many specific types of victims as possible. The objective on which type of victims should be rescued was given to the participants. This corresponds to the human knowing "What to do" for the problem. The scenario had another factor for evaluating the quality of how the participant and agent performed a rescue, that is, the pollution. The participants and agents had to keep their pollution as low as possible. There were two types of pollution, and the degree of pollution was different. The agent had information on the degree of pollution as the agent preference. This corresponds to the agent knowing "How to do it" for the problem. Rescuing many victims took priority over keeping pollution low. This means that "What to do" took priority over "How to do it" in the problem.

One step of action is moving from node to node or to victims. The human and agent take action in turns. They can move only to the upper side. This means that the participant cannot go back and retry choosing a path. Thus, for collaboration to be good, predicting the other's information is an important factor.

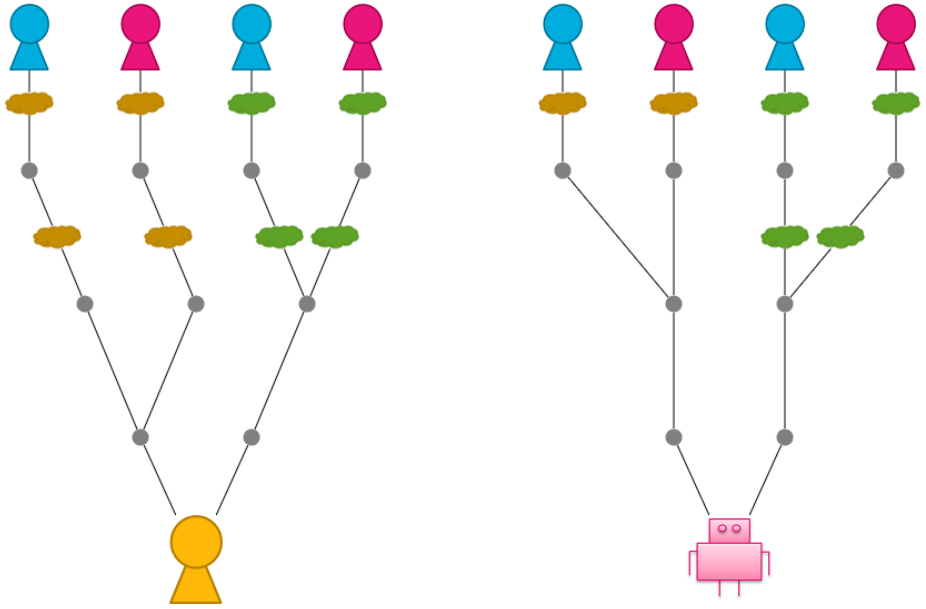

FIGURE 6.9: Example of "human resque" scenario.

Model

We modeled each scenario as a human-agent teaming formulation. The action space corresponded to one-step actions, the state space corresponded to a combination of past trajectories of humans and agents, the reward parameter for humans corresponded to an objective for the target victims that should be rescued, and the reward parameter for the agents corresponded to the preference on the degree of pollution. The reward for rescuing target victims was 400, and the rewards for passing through awful and normal pollution were -100, -10.

Types of agents

We prepared four types of agents that had other collaborate algorithms. Two of the algorithms were based on our method, and the other two were alternative algorithms.

(A) **Our method with predictability bias**: This agent used the modified Bellman update for CIRL with a Q-value including human inference for human-agent teams and used predictability bias as a human inference function. We adopted a

perfect rational policy as human policy π^H. The policy was the same as agents (B) and (C).

(B) **Our method without predictability bias**: This agent used a modified Bellman update for CIRL with a Q-value including human inference for human-agent teams and used inverse planning as the human inference function.

(C) **Optimistic CIRL**: This agent used a modified Bellman update for CIRL with a Q-value without human inference. To calculate this, an agent reward parameter was needed. We used grand truth, which the agents had. This means that agents expected humans to infer the information of the agents. This is too optimistic for comparison with reality.

(D) **Simple Min-Max** : A simple heuristic algorithm that assumes that humans take the worst action for the agents in all possible human's goal. Formally,

$$a_A = \pi_{minmax}(x) = \text{argmax}_{a_A \in \mathcal{A}_A} Q_{min}(x, a) \tag{6.12}$$

In this equation, $Q_{min}(x, a)$ can be calculated as follows.

$$Q_{min}(x, a) = \min_{a_H \in \mathcal{A}_H} Q_{min}(x', \pi_{minmax}(s')) \tag{6.13}$$

x' is the next state after x.

Stimuli

We prepared three types of stimuli (graphs) that needed different collaborative scenarios as follows.

(1) **Agent waits for information on human objective first**: The agent should know the human's objective before giving their preference. Therefore, the agent should behave in such a way as to be able support any kind of objective, and the human should behave in such a way as to reveal their objective as soon as possible.

(2) **Agent shows their preference first**: The human should know the agent's preference before selecting the best path. Therefore, the agent should behave in such a way as to reveal their preference as soon as possible even if the action is sub-optimal for the agent.

(3) **Agent shows wrong preference to encourage human behavior**: The agent should know the human's objective as soon as possible. However, to reveal the human's objective, the human should go to a less preferable side as per the agent's preference. Therefore, the agent should behave in such a way as to reveal the wrong preference. Figure 6.9 is an example of collaborative scenario (3). In this figure, the human has to rescue the blue victims, and brown pollution is more awful than green pollution. For the best collaboration, the agent should know the human's objective before the human takes a third action. To reveal the objective to the agent at an appropriate timing, the human should be go to the left side. To encourage the human to go left, the agent had better go left even though the brown pollution is more awful.

We made three types of graphs carefully in which each collaborative scenario had the best behavior. Additionally, for (1) and (2), we made two routes in which the human was told of the agent's preference. One route was preferred by predictability bias, and the other route was preferred by the inverse planning model. Therefore, planning with predictability bias and without was different in (1) and (2).

Participants

We recruited participants for this study using Yahoo! Cloudsourcing. Valid participants were 71 adults located in Japan (36 male, 21 female, 14 unknown). The mean age of participants who answered a questionnaire was 41 years old, and 60 participants answered the questionnaire for objective evaluation.

Procedure of experiment

Experiments were conducted on the Web via a browser application we made. Participants were instructed on the rules of agent behavior and then underwent a confirmation test to confirm their degree of understanding. In this test, participants who were judged to not understand the rules were given the instructions again. The participants who passed the confirmation test entered the actual experiment phase. In this phase, they saw the environment and clicked the "start" button first. After clicking button, the agent took one step forward. Then, the participants were asked "where do you want to go?" After the participants decided their step, the agent went forward one step again. Continuing this process,

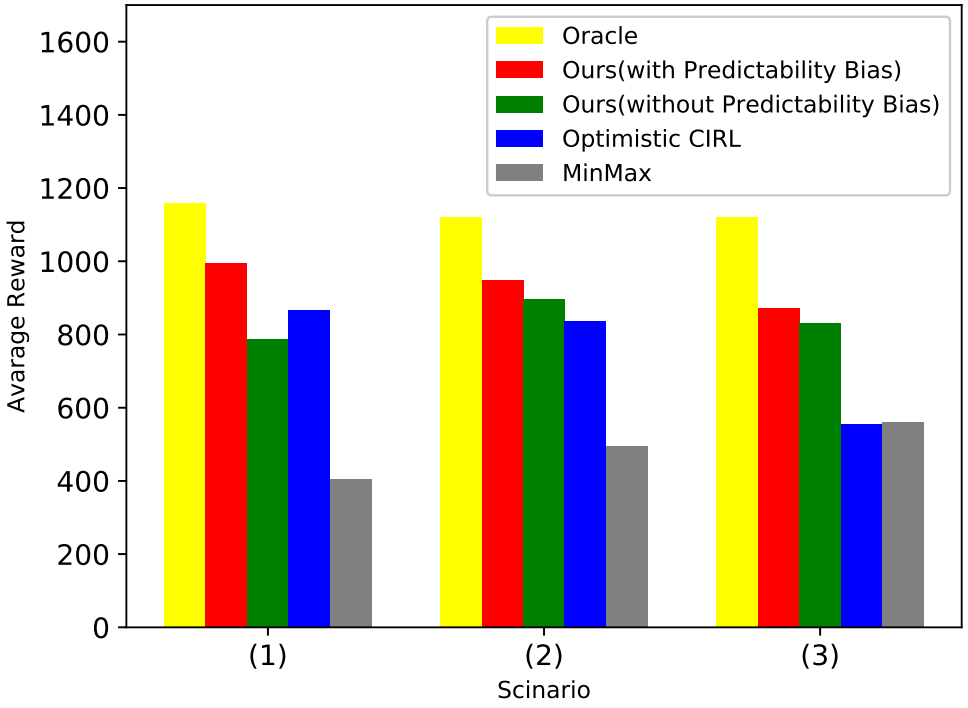

FIGURE 6.10: Average reward of collaborative task

when participants reached the top of the field, they got the next scenario. They then took a survey on the agent by using a questionnaire for subjective evaluation. Participants were asked how easy it was to collaborate with the autonomous agent for each agent. We adopted a seven-degree score for the survey.

Before analyzing the participants' results, we needed to exclude the results of the participants who were invalidated. We defined invalid participants as participants who took the same action each time. In this case, there were no invalidated participants.

6.2.5 Result

Objective evaluation

First, we evaluated the performance of the collaboration. We summarized the actual reward for the collaboration task for each stimuli and agent and averaged it. Figure 6.10 is the result. Oracle means the performance of the best route, which is that in which the human and agent had all on the information on the reward. In other words, this is the theoretical best score for the scenario.

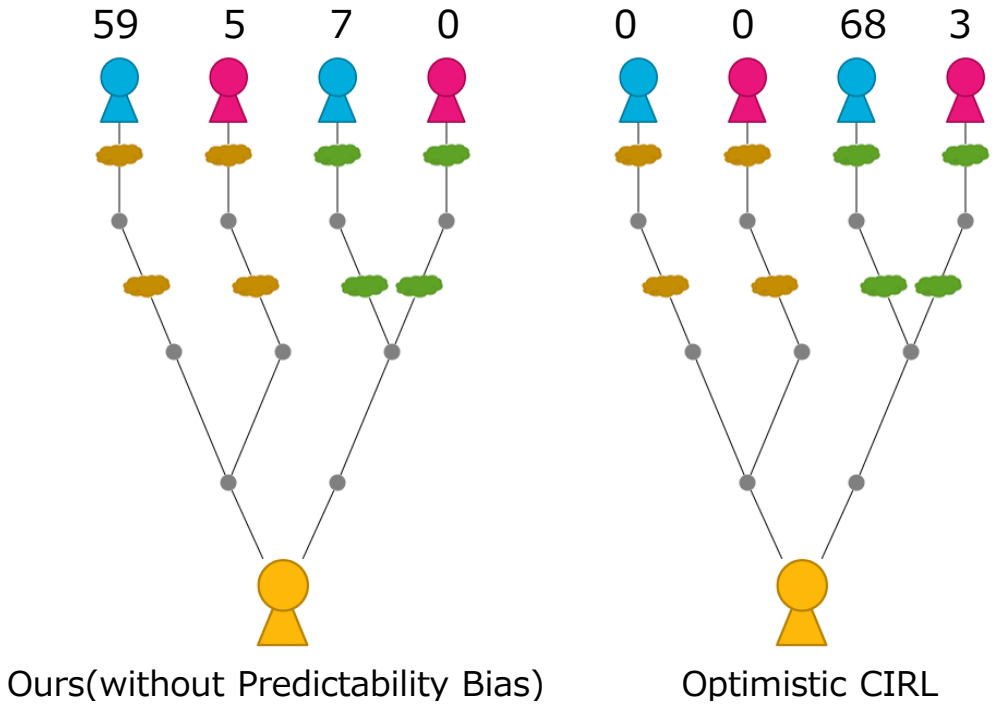

FIGURE 6.11: Difference in human destinations between our method without predictability bias and optimistic CIRL

Figure 6.9 shows that our method with predictability bias had the best performance in all collaboration scenarios. In particular, the result of scenario (3) shows that the two agents based on our method had significantly better performance than the optimistic CIRL. In scenario (3), our method revealed the wrong preference to humans to get their objective as soon as possible. However, optimistic CIRL did not do this because CIRL does not assume that humans infer agents' preferences. To more concretely analyze this, we show concrete human behavior in collaboration with our method without predictability bias and optimistic CIRL in Fig. 6.11. The numbers above victims indicate the number of participants who went to the victims. Figure 6.11 shows that participants tended to go to the left during collaboration with our method. The agents based on our method went to the left first in Fig. 6.9 despite the fact that the agents wanted to avoid brown pollution. This means that our method revealed the wrong preference to the participants to encourage them to take a more preferable way to perform the task. In comparison, CIRL does not assume that humans infer the preferences of agents, so the agents took the best action (go right). As a result, the participants indicated the right side, so the agents could not meet their objective at a proper timing. This is one example in which our method can lead to good collaborative

	(1)	(2)	(3)
Ours (with Predictability Bias)	296.3	378.4	266.2
Ours (without Predictability Bias)	447.8	440.6	315.0
Optimistic CIRL	405.6	376.4	349.4
Min-Max	315.7	297.1	306.2

TABLE 6.4: Standard deviation of average rewards of collaborative task

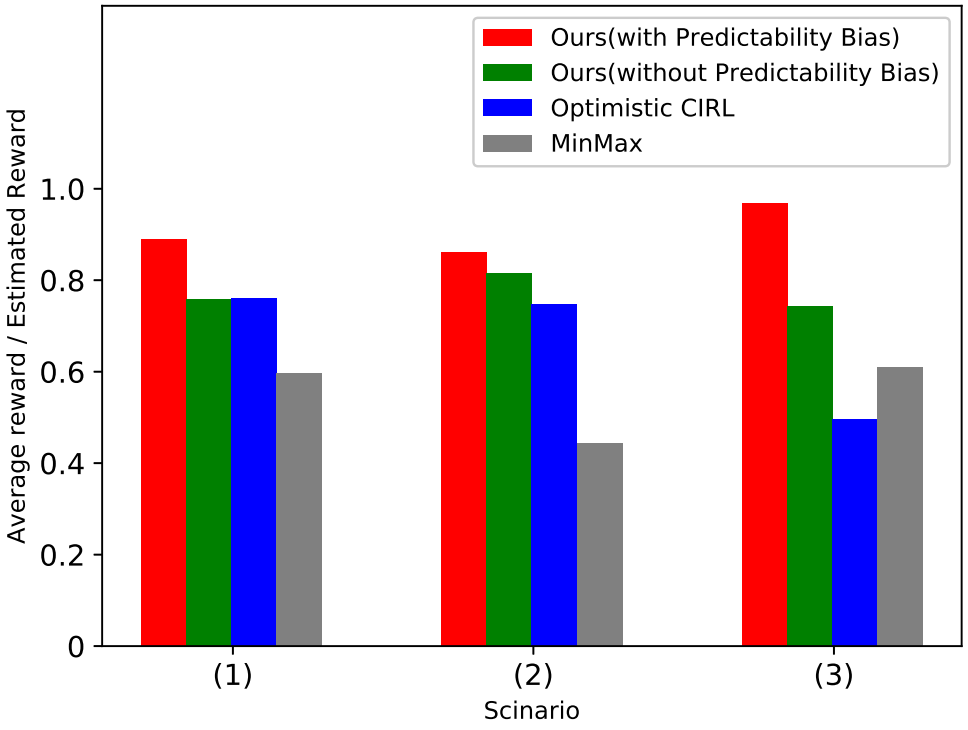

FIGURE 6.12: Ratio between expected and real rewards for each stimuli and agent

planning, which cannot be done with optimistic CIRL.

Next, we evaluated the stability of the collaboration. Table 6.4 is the standard deviation of the rewards of the collaborative task. This result shows that our method with predictability bias had lower variance than our method without predictability bias and optimistic CIRL. We think that the reason is that our method can encourage human behavior well by revealing the agent's preference, so the variance became low. The Min-Max algorithm also had low variance. The reason is that the algorithm is less adaptive, so the agents tended to take the same action.

To evaluate the quality of encouraging humans more deeply, we calculated the ratio between the expected reward of each algorithm and the real reward. For our

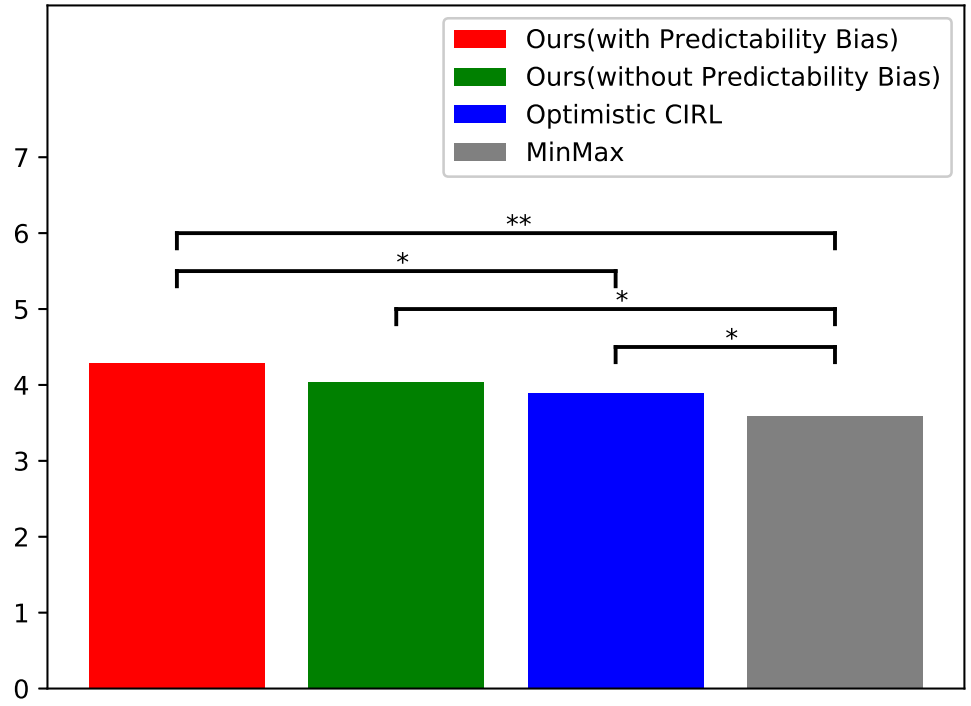

FIGURE 6.13: Statistical summary of participant's evaluation score for each agent

method and CIRL, the estimated reward was the value at the start point. The estimation was done by considering the prediction of human behavior, so if the ratio is near 1, it means that the agent can infer the human's future behavior accurately. In other words, the agent can encourage human behavior accurately. The result is shown in Fig. 6.12. This figure shows that our method with predictability bias had the best ratio, meaning that our method can encourage human behavior well. That is one piece of evidence that using a cognitive model encourages humans well.

Subjective evaluation

We also executed a subjective evaluation by using the survey results. We summarized the evaluation score from the participants for each agent. Figure 6.13 is the average of the participants' evaluation score for each agent. This shows that our method, especially with predictability bias, had a higher score than the other methods. This indicates that the cognitive model could improve peoples' impression of collaboration. In addition, we checked for a significant difference using

the Bonferroni method with a t-test. The result is also in Fig. 6.13. "*" represents $p < 0.05$, and "**" represents $p < 0.01$. There was no significant difference between our method without predictability bias and optimistic CIRL. However, there was significant difference between our method with predictability bias and optimistic CIRL. This means that we can make an agent that is more subjectively collaborative when combined with a cognitive model. This would be a positive result that bridges artificial intelligence and cognitive science.

6.2.6 Discussion

The biggest limitation of our method is that the class of agent preferences is a discrete set. This means that the human knows all candidates of agent preferences and has details on the reward parameter for all candidates. This is non-realistic in almost all situations. In particular, "how to do it" is difficult to represent as a discrete set, so this limitation should be a big problem.

The scalability of the algorithm comes next as a limitation. Since our method is based on a simple value iteration of POMDP, we can use a promising approximate solution such as SARSOP [82] and POMCP [83] to solve huge problems. However, POMDP planning generally has high computational costs, so our method is limited by the size of the application. State abstraction methods for POMDP [84] are an approach to reducing computational cost. In recent years, the deep neural network has been used to solve POMDP problems. Essentially, the state of POMDP is a history of observations. Thus, [they represent the information of history is represented by using an RNN structure and culculated using a deep reinforcement learning approach [85]. This approach uses an approximate method [86]. Furthermore, there are studies on modeling the overall planning process as a neural network [87]. These approaches are useful for approximation and abstraction.

Inference depth might be also limitation of our method. Speaking in the context of I-POMDP, our method considers only one level nest of a belief structure. Cutting off the nest of a belief structure might decrease performance in a collaboration. However, in recent research, human inference is not that deep [18, 19]; thus, we think this is not much of a problem.

Chapter 7

General Discussion

This chapter is a general discussion. Section 7.1 refers to the primary limitations of our current method. Section 7.2 discusses which situation is effective for implicit guidance in human-agent teaming. Section 7.3 introduces potential applications for the future.

7.1 Limitation

As described in the above discussions, the experiments in this dissertation were simple and small; thus, it would be difficult to apply our method to larger and more complex situations such as the natural environment. Thus, more study is needed to extend the problem coverage and increase the calculation efficiency (approximation algorithm, abstraction of the problem, etc.).

Furthermore, there is correspondence for the dynamic and unknown situation as a current essential challenge. We assume the agent knows the problem setting ultimately then calculate the likelihood of a human's decision and inference in advance. However, in real problems, the situation changes every second. Thus, it is difficult to precalculate using enough time. For a stable agent, it needs bounding

pre-calculation time. One solution is anytime algorithm [88] which returns approximative solution in fix time. Anytime algorithm is studyed for POMDP[46, 83], I-POMDP[17]. Reusing precalculated results is another solution. The change of environment is loose, not drastic in most real situations. Thus it is likely able to use the precalculated result for the previous environment. There is a study about a POMDP-based agent to use the plan cache for noisy environments [89]. That is a clue to reuse previous results. Moreover, abstraction using the recent deep learning technique is a good direction. Recently, some studies to implement ad-hoc robust support agents based on neural network model [90]. For example, we can consider the improvement of our implicit guidance agent more robustly using incremental learning based on precalculated policy.

7.2 Valuable Situation for Implicit Guidance Agent

This dissertation mainly focused on balancing human autonomy and improving task performance as the advantages of implicit guidance for human-agent teaming. Moreover, we also showed that implicit guidance is valuable for improving task performance in complex situations. However, there are many other situations where implicit guidance works to enhance communication.

The most typical case is when it is difficult to build a stable protocol or mechanism to communicate. Most communication is through predetermined protocols and mechanisms. Similar to machine-to-machine communication, it goes without saying that even human-to-human communication uses such protocols, that is, language. Of course, humans can use non-verbal communication such as body language; however, there needs to be common agreement in the culture, country, community, and so on for the communication to be correct. Although our implicit guidance is a part of non-verbal communication, it expects the human's cognitive functioning to be the source of communication. We think that this functioning is less influenced by the difference in culture etc. Next, implicit guidance does not need a specific mechanism for communication since the communication is based on? observing others' behavior. Furthermore, it does not need to observe others' actions directly; it can infer their actions indirectly by observing changes in the environment. Observing the environment is an essential function for an autonomous agent. Thus, it is shown that implicit guidance does not need a specific

mechanism. Therefore, implicit guidance is valuable in situations that do not have a specific communication protocol.

Another situation is when there is trouble communicating. In human-agent teaming, humans and agents have an actual purpose to achieve. In most cases, the types of action for communicating differ from the actions toward the actual purpose. Furthermore, the actions for communicating might be time-consuming. In such cases, the need for explicit communication may become a burden to humans. Implicit guidance uses the same types of actions to achieve an actual purpose; thus, it could relieve humans of such a burden.

7.3 Potential Applications

We mainly focused on the balance between human autonomy and task performance as the advantages of implicit guidance. From this perspective, the most promising application domain is video games. Video games have many non-player characters (NPCs) that assist or hinder the game player. In particular, an NPC that assists the player is the agent in human-agent teaming. Human autonomy is an essential factor for a player to enjoy video games. Likewise, playing better is also important. Thus, a balance between performance and human autonomy is what the player needs to have fun. Recently, some machine learning methods have been developed in the video game domain for single play [91, 92] and team play [93, 94]. Recently, video game environments have also been the focus of studies on ad-hoc human-agent teaming. The game most focused on is "Overcooked! 2" [95], which is a cooperative cooking simulation video game developed by Team17. In the game, teams cooperatively prepare and cook orders in absurd restaurants. Players gather, chop, and cook ingredients, combine them on plates, serve dishes, and wash dishes. Players have to cooperate quickly and intelligently to play the game skillfully. Our idea of implicit guidance is one way for smart cooperation. A multi-agent testbed based on the game was developed [96], and Fig. 7.1 is a screenshot. This game has many essential factors for good human-agent teaming, so many studies have used simple environments based on this game situation [8, 11].

The application for education is also interesting, especially, a serious game [97] which is a combination of education and game is an excellent potential application. In a serious game, students learn specific topics through game playing. Some

FIGURE 7.1: Video game testbed for Overcooked! 2

players may deviate from the proper learning path, but force adjusting students' playing has the risk of decreasing students' motivation. Implicit guidance has the potential of crucial technology to avoid this risk. A serious game is also one focused environment for multi-agent system study [98].

As described above, this study is a specific case of multi-agent communication, especially ad-hoc coordination. Thus, it has potential for the same applications as them. For example, autonomous driving is one of the hottest applications in multi-agent planning problems [99]. Another advantage of implicit guidance is that we do not need a specific protocol mechanism. This feature is important for environments that include heterogeneous agents since it is difficult to build communication protocols that can implement each other. A traffic environment that has both autonomously driven cars and human driven cars is one typical environment.

This study is closely related to "nudges" [100] in terms of guiding people to make more rational choices instead of making wrong choices based on human irrationality. The balance between performance and autonomy targeted in this study is very close to the idea of "libertarian paternalism" [101] that underlies the idea of nudges. Therefore, nudge technology has potential problems, such as the risk that users will be led to the result that the designer wants. To avoid these risks, it is

important to consider AI ethics to apply our methodology to practical applications, for example, avoiding using the technology for choices that are made on the basis of personal values

Chapter 8

Conclusion

In this work, we demonstrated that a collaborative agent based on "implicit guidance" is effective at providing a balance between improving a human's plans and maintaining the human's autonomy. Implicit guidance can guide human behavior toward better strategies and improve the performance in collaborative tasks. Furthermore, our approach makes humans feel as though they have autonomy during tasks, more so than when an agent guides them explicitly. We implemented agents based on implicit guidance by integrating the Bayesian theory of mind model into the existing POMDP planning. Furthermore, we extended the framework for more realistic problems. First, we implemented a kind of egocentric "bounding rationality" factor, "plan predictability bias," to the existing Bayesian theory of mind model. Next, we implemented implicit guidance into the existing collaborative planning algorithm for inferring humans' reward information. This approach can make an agent use implicit guidance in a problem where the human can infer the specific intention of the agent through its behavior. We ran a behavioral experiment in which humans performed simple tasks with autonomous agents, and we confirmed that our proposed approach has an advantage compared with the existing approach. Our experiment environment was syntactic and small. Thus, our approach might not be adopted for real applications soon. However, the method we proposed under this theory is related to guiding humans' behavior in a better way naturally. Recently, life has been improved by artificial intelligence. AI is

quite convenient, but a not small number of people are anxious toward a life that is controlled by AI. We believe that our method can be a factor in improving our world and be a key factor in manually helping the relationship between humans and AI. There would be no greater happiness than if our thesis becomes one part that improves the relationship between humans and artificial intelligence.

Bibliography

[1] Tarek Taha, Jaime Valls Miró, and Gamini Dissanayake. A pomdp framework for modelling human interaction with assistive robots. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, pages 544–549. IEEE, 2011.

[2] Owen Macindoe, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Pomcop: Belief space planning for sidekicks in cooperative games. In *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2012.

[3] Herbert A Simon. Administrative behavior. *AJN The American Journal of Nursing*, 50(2):46–47, 1950.

[4] Daniel Clement Dennett. *The intentional stance*. MIT press, 1987.

[5] Kimberly Stowers, Lisa L Brady, Christopher J MacLellan, Ryan Wohleber, and Eduardo Salas. Improving teamwork competencies in human-machine teams: Perspectives from team science. *Frontiers in Psychology*, 12:1669, 2021.

[6] Matthew Johnson, Jeffrey M Bradshaw, Paul J Feltovich, Catholijn M Jonker, M Birna Van Riemsdijk, and Maarten Sierhuis. Coactive design: Designing support for interdependence in joint activity. *Journal of Human-Robot Interaction*, 3(1):43–69, 2014.

[7] Ali Dorri, Salil S Kanhere, and Raja Jurdak. Multi-agent systems: A survey. *Ieee Access*, 6:28573–28593, 2018.

[8] Sarah A Wu, Rose E Wang, James A Evans, Joshua B Tenenbaum, David C Parkes, and Max Kleiman-Weiner. Too many cooks: Bayesian inference for coordinating multi-agent collaboration. *Topics in Cognitive Science*, 13(2): 414–432, 2021.

[9] DJ Strouse, Max Kleiman-Weiner, Josh Tenenbaum, Matt Botvinick, and David Schwab. Learning to share and hide intentions using information regularization. *Advances in Neural Information Processing Systems*, 31:10270–10281, 2018.

[10] Yuqing Du, Stas Tiomkin, Emre Kiciman, Daniel Polani, Pieter Abbeel, and Anca Dragan. Ave: Assistance via empowerment. *Advances in Neural Information Processing Systems*, 33:4560–4571, 2020.

[11] 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*, 32:5174–5185, 2019.

[12] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *Proceedings of the thirty-sixth international conference on Machine learning*, pages 3040–3049. PMLR, 2019.

[13] Peter Stone, Gal A Kaminka, Sarit Kraus, and Jeffrey S Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proceeding of Twenty-Fourth AAAI Conference on Artificial Intelligence*, page 1504–1509, 2010.

[14] Piotr J Gmytrasiewicz and Prashant Doshi. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research*, 24:49–79, 2005.

[15] Prashant Doshi and Dennis Perez. Generalized point based value iteration for interactive pomdps. In *Proceeding of Twenty-Third AAAI Conference on Artificial Intelligence*, volume 320, pages 63–68, 2008.

[16] Prashant Doshi and Piotr J Gmytrasiewicz. Monte carlo sampling methods for approximating interactive pomdps. *Journal of Artificial Intelligence Research*, 34:297–337, 2009.

[17] Brenda Ng, Carol Meyers, Kofi Boakye, and John Nitao. Towards applying interactive pomdps to real-world adversary modeling. In *Twenty-Second IAAI Conference on Artificial Intelligence*, pages 1814–1820, 2010.

[18] Harmen De Weerd, Rineke Verbrugge, and Bart Verheij. How much does it help to know what she knows you know? an agent-based simulation study. *Artificial Intelligence*, 199:67–92, 2013.

[19] 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.

[20] J Alberto Espinosa, F Javier Lerch, and Robert E Kraut. Explicit versus implicit coordination mechanisms and task dependencies: One size does not fit all. 2004.

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

[22] Vaibhav V Unhelkar and Julie A Shah. Contact: Deciding to communicate during time-critical collaborative tasks in unknown, deterministic domains. In *Thirtieth AAAI Conference on Artificial Intelligence*, page 2544–2550, 2016.

[23] Graeme Best, Michael Forrai, Ramgopal R Mettu, and Robert Fitch. Planning-aware communication for decentralised multi-robot coordination. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation*, pages 1050–1057. IEEE, 2018.

[24] Ece Kamar, Ya' akov Gal, and Barbara J Grosz. Incorporating helpful behavior into collaborative planning. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. Springer Verlag, 2009.

[25] Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

[26] Chris Baker, Rebecca Saxe, and Joshua Tenenbaum. Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the 33rd annual meeting of the cognitive science society*, volume 33, page 2469–2474, 2011.

[27] Chris L Baker and Joshua B Tenenbaum. Modeling human plan recognition using bayesian theory of mind. *Plan, activity, and intent recognition: Theory and practice*, 7:177–204, 2014.

[28] 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.

[29] Mark K Ho, Michael Littman, James MacGlashan, Fiery Cushman, and Joseph L Austerweil. Showing versus doing: Teaching by demonstration. *Advances in neural information processing systems*, 29:3027–3035, 2016.

[30] Jan Pöppel and Stefan Kopp. Egocentric tendencies in theory of mind reasoning: An empirical and computational analysis. In *Proceedings of the 41th annual meeting of the cognitive science society*, pages 2585–2591, 2019.

[31] Irina Rabkina and Kenneth D Forbus. Analogical reasoning for intent recognition and action prediction in multi-agent systems. In *Proceedings of the Seventh Annual Conference on Advances in Cognitive Systems*, pages 504–517, 2019.

[32] Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *Proceedings of the thirty-fifth international conference on Machine learning*, page 4218–4227. PMLR, 2018.

[33] Anca D Dragan. Robot planning with mathematical models of human state and action. *arXiv preprint arXiv:1705.04226*, 2017.

[34] Anca D Dragan, Kenton CT Lee, and Siddhartha S Srinivasa. Legibility and predictability of robot motion. In *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction*, pages 301–308. IEEE, 2013.

[35] Dorsa Sadigh, Shankar Sastry, Sanjit A Seshia, and Anca D Dragan. Planning for autonomous cars that leverage effects on human actions. In *Robotics: Science and Systems*, volume 2, pages 1–9, 2016.

[36] Tathagata Chakraborti, Sarath Sreedharan, and Subbarao Kambhampati. Human-aware planning revisited: A tale of three models. In *Proc. of the*

IJCAI/ECAI 2018 Workshop on EXplainable Artificial Intelligence (XAI)., 2018.

[37] Matthew Gombolay, Anna Bair, Cindy Huang, and Julie Shah. Computational design of mixed-initiative human–robot teaming that considers human factors: situational awareness, workload, and workflow preferences. *The International journal of robotics research*, 36(5-7):597–617, 2017.

[38] Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, 6(5):679–684, 1957.

[39] Edward Jay Sondik. *The optimal control of partially observable Markov processes*. Stanford University, 1971.

[40] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[41] Sylvie Ong, Yuri Grinberg, and Joelle Pineau. Mixed observability predictive state representations. In *Proceeding of Twenty-Seventh AAAI Conference on Artificial Intelligence*, volume 27, 2013.

[42] Harold William Kuhn and Albert William Tucker. *Contributions to the Theory of Games*, volume 2. Princeton University Press, 1953.

[43] H. W. Kuhn. *Extensive games and the problem of information*. Princeton University Press, Princeton, NJ, 1953.

[44] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[45] Anthony Rocco Cassandra. *Exact and approximate algorithms for partially observable Markov decision processes*. PhD thesis, Brown University, 1998.

[46] Joelle Pineau, Geoff Gordon, and Sebastian Thrun. Point-based value iteration: An anytime algorithm for pomdps. In *Proceeding of the 18th International Joint Conference on Artificial Intelligence*, volume 3, pages 1025–1032, 2003.

[47] Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. *Advances in neural information processing systems*, 29:3916—3924, 2016.

[48] Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840, 2002.

[49] Dhruv Malik, Malayandi Palaniappan, Jaime Fisac, Dylan Hadfield-Menell, Stuart Russell, and Anca Dragan. An efficient, generalized bellman update for cooperative inverse reinforcement learning. In *Proceedings of the thirty-fifth international conference on Machine learning*, pages 3394–3402. PMLR, 2018.

[50] Stephen M Stigler. *The history of statistics: The measurement of uncertainty before 1900*. Harvard University Press, 1986.

[51] Kenneth J Arrow and Gerard Debreu. Existence of an equilibrium for a competitive economy. *Econometrica: Journal of the Econometric Society*, pages 265–290, 1954.

[52] Louis L Thurstone. A law of comparative judgment. *Psychological review*, 34(4):273, 1927.

[53] Emil Julius Gumbel. *Statistics of extremes*. Columbia university press, 1958.

[54] Chris L Baker, Julian Jara-Ettinger, Rebecca Saxe, and Joshua B Tenenbaum. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):1–10, 2017.

[55] Jaedeug Choi and Kee-Eung Kim. Hierarchical bayesian inverse reinforcement learning. *IEEE transactions on cybernetics*, 45(4):793–805, 2014.

[56] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Proceeding of Twenty-Third AAAI Conference on Artificial Intelligence*, volume 8, pages 1433–1438, 2008.

[57] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *Proceeding of the 20th International Joint Conference on Artificial Intelligence*, volume 7, pages 2586–2591, 2007.

[58] Gergely Neu and Csaba Szepesvári. Apprenticeship learning using inverse reinforcement learning and gradient methods. pages 295–302, 2012.

[59] Joshua B Tenenbaum, Charles Kemp, Thomas L Griffiths, and Noah D Goodman. How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022):1279–1285, 2011.

[60] Rene Vidal, Omid Shakernia, H Jin Kim, David Hyunchul Shim, and Shankar Sastry. Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. *IEEE transactions on Robotics and Automation*, 18(5):662–669, 2002.

[61] Julie Shah and Cynthia Breazeal. An empirical analysis of team coordination behaviors and action planning with application to human–robot teaming. *Human factors*, 52(2):234–245, 2010.

[62] Luca Schenato, Songhwai Oh, Shankar Sastry, and Prasanta Bose. Swarm coordination for pursuit evasion games using sensor networks. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2493–2498. IEEE, 2005.

[63] Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *Proceedings of The 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 66–83. Springer, 2017.

[64] Guy Hoffman. Evaluating fluency in human–robot collaboration. *IEEE Transactions on Human-Machine Systems*, 49(3):209–218, 2019.

[65] Michael Lewis, Huao Li, and Katia Sycara. Deep learning, transparency, and trust in human robot teamwork. In *Trust in Human-Robot Interaction*, pages 321–352. Elsevier, 2021.

[66] Dong-Hee Shin and Hyungseung Choo. Modeling the acceptance of socially interactive robotics: Social presence in human–robot interaction. *Interaction Studies*, 12(3):430–460, 2011.

[67] Matthew C Gombolay, Reymundo A Gutierrez, Shanelle G Clarke, Giancarlo F Sturla, and Julie A Shah. Decision-making authority, team efficiency and human worker satisfaction in mixed human–robot teams. *Autonomous Robots*, 39(3):293–312, 2015.

[68] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Proceedings of the seventeenth international conference on Machine learning*, volume 1, pages 663–670. PMLR, 2000.

[69] Julian Jara-Ettinger. Theory of mind as inverse reinforcement learning. *Current Opinion in Behavioral Sciences*, 29:105–110, 2019.

[70] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pages 1–8. PMLR, 2004.

[71] Sergey Levine, Zoran Popovic, and Vladlen Koltun. Nonlinear inverse reinforcement learning with gaussian processes. *Advances in neural information processing systems*, 24:19–27, 2011.

[72] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888*, 2015.

[73] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart Russell, and Anca Dragan. Inverse reward design. *Advances in Neural Information Processing Systems*, 31:6768—-6777, 2017.

[74] Robert E Wood. Task complexity: Definition of the construct. *Organizational behavior and human decision processes*, 37(1):60–82, 1986.

[75] Rosemarie Nagel. Unraveling in guessing games: An experimental study. *The American Economic Review*, 85(5):1313–1326, 1995.

[76] Meredyth Daneman and Patricia A Carpenter. Individual differences in working memory and reading. *Journal of Verbal Learning and Verbal Behavior*, 19(4):450–466, 1980.

[77] Stefanos Nikolaidis, Swaprava Nath, Ariel D Procaccia, and Siddhartha Srinivasa. Game-theoretic modeling of human adaptation in human-robot collaboration. In *Proceedings of the 12th ACM/IEEE International Conference on Human-Robot Interaction*, pages 323–331, 2017.

[78] Tan Zhi-Xuan, Jordyn L Mann, Tom Silver, Joshua B Tenenbaum, and Vikash K Mansinghka. Online bayesian goal inference for boundedly-rational planning agents. *Advances in Neural Information Processing Systems*, 33: 19238–19250, 2020.

[79] Anca Dragan and Siddhartha Srinivasa. Integrating human observer inferences into robot motion planning. *Autonomous Robots*, 37(4):351–368, 2014.

[80] Jaime F Fisac, Monica A Gates, Jessica B Hamrick, Chang Liu, Dylan Hadfield-Menell, Malayandi Palaniappan, Dhruv Malik, S Shankar Sastry, Thomas L Griffiths, and Anca D Dragan. Pragmatic-pedagogic value alignment. In *Robotics Research*, pages 49–57. Springer, 2020.

[81] Mark Blokpoel, Johan Kwisthout, and Iris van Rooij. When can predictive brains be truly bayesian? *Frontiers in psychology*, 3:406, 2012.

[82] Hanna Kurniawati, David Hsu, and Wee Sun Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and systems*, volume 2008, 2008.

[83] David Silver and Joel Veness. Monte-carlo planning in large pomdps. volume 23, pages 2164—2172, 2010.

[84] Joelle Pineau and Sebastian Thrun. An integrated approach to hierarchy and abstraction for pomdps. *Technical report CMU-RI-TR-02-21*, 2002.

[85] Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *2015 AAAI fall symposium series*, pages 29–37, 2015.

[86] Peter Karkus, David Hsu, and Wee Sun Lee. Qmdp-net: Deep learning for planning under partial observability. *arXiv preprint arXiv:1703.06692*, 2017.

[87] Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value iteration networks. *Advances in neural information processing systems*, 29:2154—2162, 2016.

[88] Shlomo Zilberstein. Using anytime algorithms in intelligent systems. *AI magazine*, 17(3):73–73, 1996.

[89] Gavin Rens and Deshendran Moodley. A hybrid pomdp-bdi agent architecture with online stochastic planning and plan caching. *Cognitive Systems Research*, 43:1–20, 2017.

[90] Mark Woodward, Chelsea Finn, and Karol Hausman. Learning to interactively learn and assist. In *Thirty-fourth AAAI Conference on Artificial Intelligence*, pages 2535–2543, 2020.

[91] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[92] Ruben Rodriguez Torrado, Philip Bontrager, Julian Togelius, Jialin Liu, and Diego Perez-Liebana. Deep reinforcement learning for general video game ai. In *IEEE Conference on Computational Intelligence and Games*, pages 1–8. IEEE, 2018.

[93] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[94] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemys law Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[95] Ghost Town Games. Overcooked! 2. 2018.

[96] Justin Bishop, Jaylen Burgess, Cooper Ramos, Jade B Driggs, Tom Williams, Chad C Tossell, Elizabeth Phillips, Tyler H Shaw, and Ewart J de Visser. Chaopt: a testbed for evaluating human-autonomy team collaboration using the video game overcooked! 2. In *2020 Systems and Information Engineering Design Symposium*, pages 1–6. IEEE, 2020.

[97] Tarja Susi, Mikael Johannesson, and Per Backlund. Serious games: An overview. *Technical report HIS-IKI-TR-07-001*, 2007.

[98] Elaachak Lotfi, Amine Belahbib, and Mohammed Bouhorma. Towards a system of guidance, assistance and learning analytics based on multi agent system applied on serious games. *International Journal of Electrical and Computer Engineering*, 5:344–354, 04 2015.

[99] Praveen Palanisamy. Multi-agent connected autonomous driving using deep reinforcement learning. In *2020 International Joint Conference on Neural Networks*, pages 1–7. IEEE, 2020.

[100] Richard H Thaler and Cass R Sunstein. Nudge: improving decisions about health, wealth, and happiness. *Newhaven: Yale*, 2009.

[101] Richard H Thaler and Cass R Sunstein. Libertarian paternalism. *The American Economic Review*, 93(2):175–179, 2003.

List of Publication

1. Ryo Nakahashi and Seiji Yamada. Modeling Human Inference of Others' Intentions in Complex Situations with Plan Predictability, In *Proceeding of 40th Annual cognitive Science Society Meeting (CogSci 2018)*, pp. 2147—2152, July 2018

2. Ryo Nakahashi, Seiji Yamada,Balancing Performance and Human Autonomy With Implicit Guidance Agent, In *Frontiers Artificial Intelligence,* 4: 736321, Oct. 2021

3. Ryo Nakahashi, Seiji Yamada, Framework for Human-Agent Team using Implicit Information Showing via Behavior, *The 33th Annual Conference of the Japanese Society for Artificial Intelligence*, June. 2019