Enabling Collaboration With Intention Inference Using Partially Observable Markov Decision Processes

MASTER OF SCIENCE DISSERTATION

Iordan Tchaparov
1068874

Supervised by Dr. Benjamin Rosman

October 1, 2021

A dissertation submitted in fulfilment of the requirements for the degree of Master of Science

in the

School of Computer Science & Applied Mathematics

UNIVERSITY OF THE WITWATERSRAND

Declaration of Authorship

I, Iordan Tchaparov, declare that this dissertation titled, "Enabling Collaboration With Intention Inference Using Partially Observable Markov Decision Processes" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this dissertation has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this dissertation is entirely my own work.
- I have acknowledged all main sources of help.
- Where the dissertation is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: October 1, 2021

UNIVERSITY OF THE WITWATERSRAND

Abstract

School of Computer Science & Applied Mathematics

Master of Science

Enabling Collaboration With Intention Inference Using Partially Observable Markov Decision Processes

by Iordan Tchaparov

Efficient collaboration between two humans jointly solving a task is often only possible if both humans know the other's intention. They could communicate their intention verbally or simply observe one another until they have reasoned about the other's intention before selecting their own approach to completing the task. This is relatively easy for humans to do due to their Theory of Mind - their ability to reason about other's behaviour to develop a suitable response. Imbuing a robot with the same ability to enable collaboration with a human on those same tasks is complex as the robot lacks a Theory of Mind and so this system needs to be developed to allow the robot to infer a human's intention.

In this dissertation, a reasoning mechanism akin to Theory of Mind for a robot to reason about intentions is developed called an Action Trajectory Encapsulated Partially Observable Markov Decision Process (ATE-POMDP). This is modelled from the action trajectories of the human, whereby the agent aims to infer the current intention of the human based on observing their actions. Once their intention is found, the robot selects a task to complete itself, thereby collaborating with the human by completing some appropriate sub-components of the task.

The ATE-POMDP is tested in simulation with simulated humans against other collaborative teams in gridworld environments to establish the effectiveness of this approach to intention inference. The results show that the ATE-POMDPs provide performance comparable to the other teams when there is a small collision time cost but are the best option for intention inference when collision time costs are large. This makes using ATE-POMDPs suitable to enable collaboration through intention inference.

Acknowledgements

This dissertation is the product of two and a half difficult yet rewarding years. I would like to extend thanks and appreciation to my supervisor Dr. Benjamin Rosman for that meeting in late 2018 that kicked off this entire journey and for not only taking a chance on a student who just wanted to work in the field of reinforcement learning because it sounded "*cool*" but also for his wealth of knowledge, continued support and excellent advice.

A heartfelt thank you to my family who had the patience and willingness to support me on this journey for the past years. To my parents, I remain eternally grateful for your love and wisdom, supporting me even while uncertain about whether the decisions I made were the best ones. To my brother, who has been my inspiration, confidant and biggest critic, I don't think I would have gotten this far without your support and compromise.

Lastly, I want to thank all my friends who have helped me on this journey, whether directly or indirectly. Whether it be the members of "*Clen*" (the colloquial name of my friend group), or my old friends at the school of electrical engineering or to the new friends I made in the Rail lab and school of computer science - you all served as motivation for me to see this journey through to the end and for that, a massive thank you.

Contents

List of Figures

List of Tables

List of Abbreviations

ToM	**T**heory **o**f **M**ind
HRC	**H**uman-**R**obot **C**ollaboration
RL	**R**einforcement **L**earning
MDP	**M**arkov **D**ecision **P**rocess
TD	**T**emporal **D**ifference
POMDP	**P**artially **O**bservable **M**arkov **D**ecision **P**rocess
HRL	**H**ierarchical **R**einforcement **L**earning
IRL	**I**nverse **R**einforcement **L**earning
ATE-POMDP	**A**ction **T**rajectory **E**ncapsulated **P**artially **O**bservable **M**arkov **D**ecision **P**rocess

Chapter 1

Introduction

1.1 Motivation

The premise of the phrase *"two heads are better than one"* argues that having two (or more) people working on a problem is better than one person. The word *better* is subjective with respect to the problem, usually taking the form of developing a higher quality solution using less time. This is the main purpose of collaboration - breaking down a large problem into smaller components, each solved by one or more participants, together completing the whole problem.

Consider a breakfast scenario whereby a single human wants to make fried eggs with a cup of coffee. The human can decide on the order, either making the eggs or coffee first. The tasks themselves can be completed in a variety of different ways, with some possible variations of coffee making including boiling water in a kettle or using a pot, using one, two or three spoons of coffee, adding sugar or milk, using a small or big mug, stirring with a spoon or even a fork. All of these depend on the human's own preference for making coffee.

Introducing a second human into the same environment would allow tasks to be completed by dividing the work - one human could make the eggs while the other makes the coffee in the breakfast scenario. The characteristic that allows humans to understand one another is called the Theory of Mind (ToM) - the ability to reason about others' mental states to be able to formulate suitable responses [Byom and Mutlu (2013)]. A human's mental state encapsulates their current thoughts, behaviours and goals - knowing another human's mental state allows the use of appropriate responding behaviour in the form of verbal communication or physical actions. The absence of a functioning ToM leads to difficulty in understanding and interacting with both the world and people who live in it [Baron-Cohen et al. (1985)].

The division of work is achieved through collaboration between humans, enabled through ToM. By understanding the intentions and behaviour of others, a human can select their own actions (and therefore behaviour) that would aid the collaboration effort. In general, there are three different categories of collaboration: shared common plan collaboration, on-the-fly collaboration and isolated collaboration.

Shared common plan collaboration involves participants coming together and creating an explicit common plan prior to starting [Bauer et al. (2008)]. Participants share their intentions amongst one another, creating an explicit common plan that contains the steps for completing different tasks in a scenario as well as assigning participants to tasks. Following this plan usually results in the most efficient completion of the scenario at the cost of taking extra time to develop the plan by

assessing available skills and preparing resources as these impact the quality of collaboration.

On-the-fly collaboration is geared towards scenarios whereby completion speed is vital and there exists no preparatory time. Participants select and complete tasks one by one, communicating their intention using a variety of techniques including speech, gesture, gaze and non-verbal cues [Green et al. (2008)]. This information in conjunction with the humans' own ToM establishes an implicit common plan among participants. Participants only have partial knowledge of the implicit plan, knowing their own intentions with complete certainty but only being able to assume their team members' intentions unless actively communicated to them.

Lastly, there is isolated collaboration - a naive method whereby participants complete tasks in the scenario without establishing any common plan, explicit or implicit. Participants do not attempt to gain information as to the task intention of other participants so that they can use this information to guide their behaviours, instead opting to work in isolation in the hopes that the overall problem will be completed faster. Without a plan, the same task can be completed by multiple participants or participants can even start completing in-progress tasks, hindering their future progress. This causes the effects of the collaboration to be minimal or even worse than one participant completing the overall problem themselves.

Continuing with the breakfast scenario - two parents plan to make breakfast for their children tomorrow - agreeing on an egg omelette, which they research how to make. This is an example of a shared common plan. Tomorrow comes and the parents have overslept! Mom wakes up first and upon seeing the time immediately begins to make fried eggs - no time for sophisticated omelettes or waking up Dad. Dad, upon smelling fried oil, wakes up and joins in by making the coffee. This is an example of on-the-fly collaboration. Once breakfast is made, they call their children - only to find them eating toast in the lounge as they didn't know their parents would make breakfast that day. This is an example of isolated collaboration - the children aided the breakfast scenario collaboration unknowingly.

Robots are machines that can complete complex tasks, do not tire and can be cheaper than hiring humans over a long period of time. Replacing the second human with a robot requires a number of systems to enable the same level of collaboration as was possible with the team of two humans. Assuming that the robot had the correct actuators for movement and manipulation, a system is required that would be able to reason about the human's behaviour as well as decide on its own actions, even if no common plan or communication exists. In the breakfast scenario, the human moving towards and using the kettle can be used to conclude that the human's intention is to make coffee and so the robot can then make fried eggs in parallel.

The breakfast scenario only comprised of two tasks - making coffee and eggs - while in reality there are many tasks in everyday life and humans have preferences for the completion order of tasks and the components of some tasks also have a required order (such as boiling the kettle before pouring the water). Explicit communication can be difficult to achieve as both parties have to agree on a common communication medium, and there could be a need for time to learn how to use it, especially in the case of a human. To alleviate the use of explicit communication, the robot could observe the human's actions to infer their intention. This remains a difficult

problem as many tasks share similarities that make it difficult to distinguish them from one another when only observing actions.

Human-robot collaboration (HRC) is a field that aims to develop robotic systems that can be used for collaboration with humans in the real world. There has been work in using robots in a variety of collaborative settings such as nursing homes [Pineau et al. (2003)], rescue missions [Gilbert and Beebe (2010)], industry [Gleeson et al. (2013)] and outer-space [Bluethmann et al. (2003)]. In each of these works, the robots are highly specialised for a particular application and limited in their ability to be generalised.

The goal of this dissertation is to focus on the collaboration category of on-the-fly collaboration without explicit communication, instead observing the human's actions to infer their intention towards a task and thereby using this information to enable collaboration. In the case of the breakfast scenario, the robot should be able to determine the human's intention for either the coffee making task or the fried eggs making task so that they can do the other in parallel thereby reducing not only the amount of work of the human, but the overall time of the breakfast scenario.

1.2 Problem Statement

Consider a human that has been the only actor in their environment for some length of time and they have developed their own behaviours used to complete tasks with a behaviour being defined as a sequence of consecutive actions. A robot assistant is introduced to aid the human in the completion of their tasks with a focus on preventing an additional burden on the human by requiring the human to learn a new communication medium, adapting their preferred behaviours or adapting their preferred task completion order. The robot must thus infer the human's intentions to be able to collaborate by observing the human.

Creating a system with the ability to reason about the behaviour of a human will function as an artificial ToM. This means it will aim to understand the intention of the human (their mental state). The mental state of the human is represented by their performed actions and these can be used as evidence to find their intention towards a specific task in their environment. The aim lies in creating a system that can infer the human's intention based on their actions. The problem statements are:

"(1) Can the intention of a human be inferred using a Partially Observable Markov Decision Process?"

and

"(2) If the intention of the human is found, how can collaboration then be enabled with that human?"

Inference is the process of combining prior knowledge as well as evidence to draw a conclusion about something - in this case the human's intention towards completing a task. Prior knowledge indicates that there needs to be enough useful information from previous experiences that is relevant to understanding the current evidence and while humans can efficiently use prior experiences to understand

new information, robots cannot do this easily. It requires associating actions with tasks and since actions are not unique to tasks, many tasks share the exact same actions. The more actions observed over time that belong to a specific task, the more accurate the inference.

Once the intention has been inferred, it can be used to influence the robot's behaviour around the human's current behaviour. In the case of an environment containing only two tasks, the robot can simply do the other task that the human appears to not currently be completing. However as the number of tasks grows, there are more intentions to infer by the robot and these affect both the accuracy and speed of the inference as well as the efficiency of the collaboration as now there exists more choices for the robot with regards to tasks to select and complete.

Reinforcement learning (RL) is a framework that uses the principle of reward based learning by having an agent perform actions under specific conditions that result in positive or negative rewards that are used to influence the agent's behaviour. Using this framework will allow an agent to learn based on the human's own behaviour. However, intentions are not directly observable in reality - only the human can know their own intention with complete certainty. This means that the standard RL technique of modelling problems as Markov Decision Processes (MDP) should not be used as these assume that the agent can observe everything including intentions. An extension to MDPs, Partially Observable Markov Decision Processes (POMDPs) are better suited to reflect these real world conditions as they do not allow direct access to the intention information, instead requiring the agent to hold a belief about the human's current intention that changes based on the observations received. Therefore information must be gathered by the robot to update its belief about the underlying human intention, and subsequently its own actions.

The contribution of this dissertation lies in creating POMDPs based on the behaviour of simulated humans for the purpose of intention inference solely using their actions. Additionally, a method is devised to deal with large POMDP models as they can quickly scale to intractability by splitting them into separate models that can be solved. The robot needs to minimise disruption of the human so that there is less of a burden on the human as they do not need to adapt their behaviours to accommodate the robot. The use of POMDPs as reasoning mechanisms is then tested in simulation with a simulated human against multiple other teams and varying robots to establish the effectiveness of using POMDPs for intention inference and whether this inference can be used to enable collaboration.

1.3 Outline

The structure of the dissertation is as follows: **Chapter 2** presents the background theory of reinforcement learning, covering Markov Decision Processes, Partially Observable Markov Decision Processes and the solutions to both. **Chapter 3** examines the methodology including assumptions and design decisions used in the development of POMDPs to infer intentions. **Chapter 4** presents the experiments, their evaluations and discussions around the results. **Chapter 5** reviews human-robot collaboration literature relevant to this research. **Chapter 6** concludes the dissertation by summarising it and providing future recommendations.

Chapter 2

Background

This chapter presents and explains the theory of reinforcement learning by giving a general overview to the field in section 2.1 before going into the details of formalising problems as Markov Decision Processes (MDPs) in section 2.2. Thereafter the theory of value functions and the Q-learning algorithm are examined in subsections 2.2.1 and 2.2.2 respectively, to illustrate the manner in which MDPs can be solved. Section 2.3 describes the theory of Partially Observable Markov Decision Processes (POMDPs), an extension to MDPs. Thereafter subsection 2.3.1 details the belief update of POMDPs while subsection 2.3.2 details POMDP value functions, both of which are used to solve POMDPs. Subsection 2.3.3 details the incremental pruning algorithm that finds a vector representation of the value function for a POMDP.

2.1 Reinforcement Learning

Reinforcement learning is the field of machine learning concerned with learning through interaction [Sutton and Barto (2018)]. The basic setting of reinforcement learning problems consists of an environment and agent, whereby the agent has a goal to achieve. This goal is represented through rewards that the agent receives during its interaction with the environment and these influence the agent's behaviour and decision making. The agent has no direct understanding of the goal, instead only aiming to maximise the cumulative reward it receives during its interaction duration. Generally, executing actions that maximise the cumulative reward results in the agent achieving its goal.

Solely maximising immediate rewards during the agent's interaction does not always achieve the goal in complex environments. The agent must leverage *exploitation* and *exploration* in order to effectively achieve its goal [Kaelbling et al. (1996)]. The agent could *exploit* immediate rewarding actions repeatedly or *explore* new actions that could eventually lead to greater future rewards. These would never have been found if it solely exploited its actions and therefore creates a trade-off of foregoing immediate rewards for future rewards. A combination of these two approaches would result in the optimal behaviour of the agent towards achieving the goal, although sub-optimal behaviours that also achieve the goal could be found.

Reinforcement learning problems are formalised as Markov Decision Processes (MDPs). MDPs are models that encapsulate the different aspects of a discrete sequential decision process that is controlled by a decision maker with the ability to interact directly with the process [Puterman (1990)].

2.2 Markov Decision Processes

A Markov Decision Process consists of a five element tuple of the form:

$$MDP = \langle S, A, T, R, \gamma \rangle$$

Where:

- S - set of states
  A collection of all possible states s of the environment. The current state is represented by s while the future successor state (also known as state prime) is represented by s'.
- A - set of actions
  A collection of all the possible actions a available to the agent.
- T - Transition function with the form $T(s' \mid a,s)$
  A probability function detailing the environment's state transitions. It fully describes the environment's dynamics by giving the probability that taking action a in state s will result in state s'.
- R - Reward function with the form $R(s,a,s')$
  A function detailing the reward given based on taking action a in state s and transitioning to state s'.
- γ - Discount factor
  A factor used to limit the reward by decaying it over the duration of the agent's interaction with the environment. It takes a value between $0 < \gamma \leq 1$. Values closer to 0 indicate that rewards lose value over time so they should be prioritised early as opposed to later in the agent's decision making while values closer to 1 indicate that rewards retain their value over a longer duration.

Figure 2.1 illustrates the relationship of the elements in MDPs. An agent selects an action a that changes the environment's state from s to s'. This results in reward r and state s (the environment's new state s') being returned to the agent. The agent uses this information to select its next action and this relationship repeats until either a terminal (ending) state is encountered or a number of fixed steps have passed (duration ends). The duration of the agent's interaction with the environment is called an *episode*, consisting of a number of time steps.

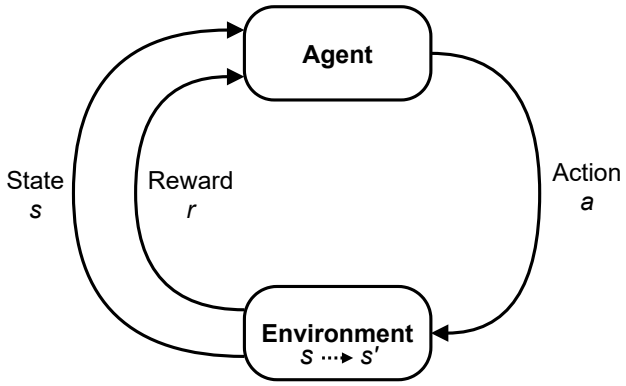


FIGURE 2.1: Markov Decision Process Loop

MDPs are *Markovian* meaning that future states are independent of the past history of states if the present state is given [Zhang et al. (2010)]. This allows the agent to select optimal actions only for the current state without regard to its previous interactions in the environment. A policy π is a mapping of states to actions such that $\pi : S \times A \rightarrow [0,1]$, thus dictating the behaviour of an agent according to the states it encounters during an episode. The optimal policy is represented by π_*, resulting in the maximum cumulative reward an agent can receive during an episode and is known as the solution to the MDP. Other policies do exist that may or may not achieve the goal of the agent.

2.2.1 Value Functions for MDPs

The return G of an episode is described as the total discounted cumulative reward R an agent receives at each time step t:

$$G = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{2.1}$$

An optimal policy π_* seeks to maximise the expected return G. This policy can be obtained using the *state-value* function that relates the worth of being in a specific state and the *action-value* function that relates the worth of an action in a specific state [Kochenderfer (2015)]. The *state-value* function $V_\pi(s)$ estimates the expected return G given state s at time t and following policy π:

$$V_\pi(s) = \mathbb{E}_\pi \left[G | s_t = s \right] \tag{2.2}$$

$$= \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | s_t = s \right] \tag{2.3}$$

The *state-value* function $V_\pi(s)$ can be further expressed as the expectation $\mathbb{E}$ of the immediate reward R_t and the future return G_{t+1} while following policy π:

$$V_\pi(s) = \mathbb{E}_\pi \left[R_t + \gamma G_{t+1} | s_t = s \right] \tag{2.4}$$

The expectation $\mathbb{E}$ accounts for the stochastic nature of the MDP dynamics given by its transition function $T(s'|a,s)$ as well as the stochastic nature of the policy π being followed from state s with $\pi(a|s)$ giving the probability of action a given state s. The *state-value* function must be the summation of each action of the policy π and each future state given by $T(s'|a,s)$:

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s'} T(s'|a,s) \left[R(s,a,s') + \gamma \, \mathbb{E}_\pi \left[G_{t+1} | s_{t+1} = s' \right] \right] \tag{2.5}$$

$$= \sum_a \pi(a|s) \sum_{s'} T(s'|a,s) \left[R(s,a,s') + \gamma V_\pi(s') \right] \tag{2.6}$$

Equation (2.6) is known as the recursive *Bellman equation* for the *state-value* function and solving this equation would result in a *state-value* function that could be used to determine the optimal policy [Bellman (1957)]. It can be seen that the equation requires the *state-value* function of the future state, making it recursive such that it

eventually converges to the optimal value function. Similarly, the *action-value* function $Q_\pi(s,a)$ estimates the expected return given state s and action a at time t and following policy π thereafter:

$$Q_\pi(s,a) = \mathbb{E}_\pi\left[G|s_t = s, a_t = a\right] \tag{2.7}$$

$$= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}|s_t = s, a_t = a\right] \tag{2.8}$$

$$= \mathbb{E}_\pi\left[R_t + \gamma G_{t+1}|s_t = s, a_t = a\right] \tag{2.9}$$

In equation (2.9), the action is fixed as given by $a_t = a$ and so the expectation does not need to sum over all actions of the policy π for the current iteration of the equation. This results in:

$$Q_\pi(s,a) = \sum_{s'} T(s'|a,s)\left[R(s,a,s') + \gamma\,\mathbb{E}_\pi\left[G_{t+1}|s_{t+1} = s'\right]\right] \tag{2.10}$$

$$= \sum_{s'} T(s'|a,s)\left[R(s,a,s') + \gamma V_\pi(s')\right] \tag{2.11}$$

Equation (2.11) is the *Bellman equation* of the *action-value* function. Finally the relationship between the *state-value* function $V_\pi(s)$ and the *action-value* function $Q_\pi(s,a)$ are related by substituting equation (2.11) into equation (2.6):

$$V_\pi(s) = \sum_a \pi(a|s) Q_\pi(s,a) \tag{2.12}$$

Substituting equation (2.12) into equation (2.11) results in:

$$Q_\pi(s,a) = \sum_{s'} T(s'|a,s)\left[R(s,a,s') + \gamma \sum_{a'} \pi(a'|s') Q_\pi(s',a')\right] \tag{2.13}$$

The optimal equations of both the *state-value* function and the *action-value* function can be found using:

$$V_*(s) = \max_\pi V_\pi(s) \tag{2.14}$$

$$Q_*(s,a) = \max_\pi Q_\pi(s,a) \tag{2.15}$$

Finally the optimal policy π_* can be found by selecting the action a that results in the largest reward as given by:

$$\pi_*(a|s) = \arg\max_a Q_*(s,a) \tag{2.16}$$

The *Bellman equations* act as a basis for many *Temporal-Difference* (TD) learning algorithms. These are algorithms that seek to combine the *Bellman equations* with experience directly from interacting within an environment [Sutton (1988)]. This means that the results of the environment, received reward r and resulting state s', are used with the *Bellman equations* to obtain a policy.

2.2.2 Q - learning

Q-learning is a common model-free TD learning algorithm for solving an MDP [Watkins and Dayan (1992)]. Model-free means that the policy for a given MDP can be computed without knowing the transition function $T(s' \mid a,s)$ or reward function $R(s,a,s')$ of the model, instead relying on experience. The steps of Q-learning are shown in Algorithm 1.

Algorithm 1: Q-learning

```
1  initialise Q for all combinations of s and a
2  initialise α
3  initialise γ
4  s ← s_0
5  while Episode do
6      if Random <= ε then
7          a ← Random a                          ▷ Select random action
8      else
9          a ← max_a Q(s,a)                      ▷ Select most rewarding action
10     end
11     Execute a
12     Observe r and s'
13     Q(s,a) ← (1 - α) · Q(s,a) + α · [r + γ max_a Q(s,a)]      ▷ Update rule
14     s ← s'
15 end
```

The Q-learning algorithm aims to evaluate the quality (hence the Q) of the actions available to the agent by executing them and experiencing their effects. In **lines 6 - 10**, the agent selects a random action a with probability ε, otherwise selecting the action a with the highest Q value in the already computed $Q(s,a)$ function from the previous iteration. This is known as *Epsilon Greedy*. The agent then executes this action a and observes the reward r and the resulting future state s'. **Line 13** is whereby the agent updates the Q value based on these factors using the update rule given as:

$$Q(s,a) \leftarrow (1-\alpha) \cdot Q(s,a) + \alpha \cdot \left[r + \gamma \max_a Q(s,a)\right] \tag{2.17}$$

Where s, s', a, r and γ are the state, future state (state prime), action, reward and discount factor respectively. The learning rate α is a parameter that controls the speed of convergence to select the actions that result in the largest rewards. Q-learning is known as an off-policy algorithm as the algorithm will be updated to converge towards the optimal policy regardless of the underlying initial policy.

2.3 Partially Observable Markov Decision Processes

MDPs make the assumption that the agent can fully observe the current state at any time step. This means that the agent has full access to the current state information allowing it to easily plan and interact with the environment. In reality, the sensors of the agent could be faulty, the environment might not be fully measurable or the variable cannot be observed directly. This means that the agent cannot completely perceive the current state leading to uncertainty [Spaan (2012)]. The agent could now perceive multiple states simultaneously and as its policy is a mapping of a single state to a single action, it cannot decide on an action. The Partially Observable Markov Decision Process (POMDP) is an extension to MDPs whereby the agent no longer has direct access to the current state information so it must instead maintain a belief about its current state. This belief is updated using information received based on the agent's interaction with the environment. A POMDP consists of the tuple:

$$POMDP = \langle S, A, T, \gamma, R, Z, O \rangle$$

Where:

- S, A, T, γ and R represent the set of states, set of actions, the transition function, discount factor and the reward function respectively. These all represent the same information in POMDPs as they do in MDPs.
- Z - set of observations
  A collection of all observations z available to the agent in the environment.
- O - Observation function with the form $O(z \mid s', a)$
  A probability function that gives the probability of receiving observation z after having taken action a and transitioned to state s'.

Figure 2.2 illustrates the relationship of the elements of POMDPs. The agent initially starts with a belief b about its current state s in the form of a probability distribution over all possible starting states. The agent then selects an action a that changes the environment from state s to state s' and receives reward r and observation z. The observation and action are both used to update the agent's current belief b to its new belief b'. This process repeats until either a terminal state is reached or the duration of the episode ends.

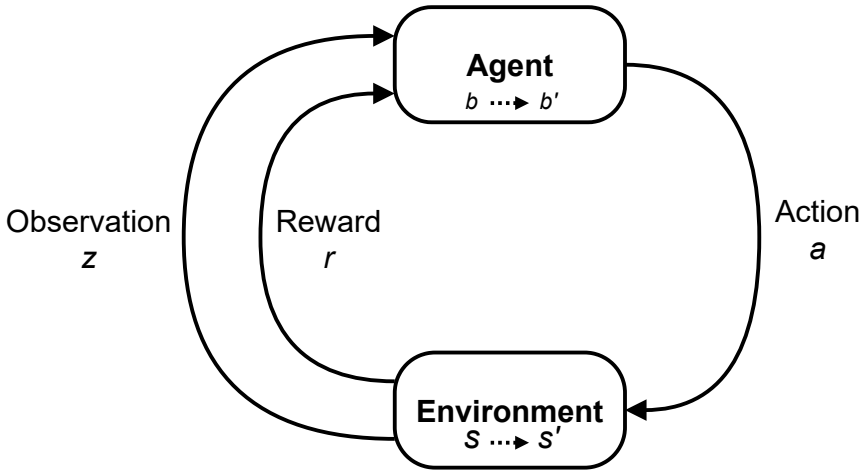


FIGURE 2.2: Partially Observable Markov Decision Process Loop

2.3.1 Belief Update

The belief space b is a probability distribution over all states with $b(s)$ giving the probability of the true state being state s. When acting under uncertainty, a history of previous experiences would aid the agent in its decision making in the future. Keeping a history however would violate the *Markov* property and so this is circumvented by using a continually updated probability distribution after each action that summarises the history of the agent. Given action a, observation z and belief $b(s)$, the new belief $b(s')$ is constructed using:

$$b(s') = \frac{1}{p_z} O(z|s', a) \sum_{s \in S} T(s'|s, a) b(s) \tag{2.18}$$

Dividing by the likelihood p_z (probability of observation z) normalises the resulting belief state to ensure that the belief space probabilities sum to 1 [Littman (2009)]. The likelihood p_z is given by:

$$p_z = \sum_{s \in S} b(s) \sum_{s' \in S} T(s'|s, a) O(z|s', a) \tag{2.19}$$

2.3.2 Value Functions for POMDPs

The solution to a POMDP is also called a policy π and it is a mapping of belief distributions of states to actions such that $\pi : b(S) \times A \rightarrow [0,1]$ with the optimal policy represented by π_*. Like MDPs, policies for POMDPs can also be found through the use of value functions that have been adjusted to include the belief state instead of direct state information. The expected reward for taking action a in belief space b is given by:

$$r_a = \sum_{s \in S} b(s) \sum_{s' \in S} R(s, a, s') \tag{2.20}$$

A horizon is defined as a finite length of time for an agent to act. In a 1-step horizon, the agent can only take a single action and be subsequently rewarded. The expected reward in a 1-step horizon is therefore the immediate expected reward it receives having taken action a in belief space b:

$$Q_1(b, a) = \sum_{s \in S} b(s) \sum_{s' \in S} R(s, a, s') \tag{2.21}$$

In equation (2.21), the expected reward for action a is a linear function of the belief b. The best action a to choose in belief b is the one that results in the highest value of $Q_1(b, a)$. The value function for the 1-step horizon is thus defined as:

$$V_1(b) = \max_{a \in A} Q_1(b, a) \tag{2.22}$$

$$= \max_{a \in A} \left[\sum_{s \in S} b(s) \sum_{s \in S} R(s, a, s') \right] \tag{2.23}$$

The function $V_1(b)$ is therefore piecewise linear and convex - the maximum values over a set of linear functions of $Q_1(b, a)$ [Kaelbling et al. (1998)]. This value function for the 1-step horizon can now be used to construct the value function for a 2-step

horizon - the cumulative reward the agent can expect if it can take two actions. However, at the end of the 1-step horizon, the agent not only receives a reward given by *R(s,a,s')* and observation z with probability p_z, the underlying state has changed to s' through *T(s' | a,s)*. Additionally, the belief b has to be updated using equation (2.18). The expected reward for selecting the second action is now the sum of the reward from the 1-step horizon with the reward from the current selected action:

$$Q_2(b,a) = r_a + \gamma \sum_{z \in Z} p_z V_1(b') \tag{2.24}$$

Substituting equation (2.19) and equation (2.20) into equation (2.24) results in:

$$Q_2(b,a) = \sum_{s \in S} b(s) \left[\sum_{s' \in S} R(s,a,s') + \gamma \sum_{z \in Z} \left[\sum_{s' \in S} T(s'|s,a) O(z|s',a) \right] V_1(b') \right] \tag{2.25}$$

Again equation (2.25) is a combination of linear functions and maximising over this set of linear functions results in the piecewise linear and convex value function given by:

$$V_2(b) = \max_{a \in A} Q_2(b,a) \tag{2.26}$$

This process of computing the value function V_{t+1} from the value function V_t continues until eventually converging to the optimal value function V'. The more states, actions and observations in a POMDP, the more linear equations will be needed to be calculated, making the solution method of finding exact value functions only applicable to small POMDP models.

2.3.3 Incremental Pruning

Incremental pruning is an exact value function solving algorithm that calculates a set of vectors that together represent the value function of a POMDP, prunes (or filters) those vectors that are dominated by others before using the remaining set to generate the next set that represents the value function of the next iteration, repeating this process until eventually converging to the optimal value function [Zhang and Liu (1996)]. By pruning the dominated vectors, that is vectors whose effects (and values) are hidden due to the presence of vectors with larger values over the same belief space b, the total number of computations decreases. Being piecewise linear and convex, the optimal value function $V^*(b)$ can be calculated by:

$$V_*(b) = \max_{\alpha \in D} b \cdot \alpha \tag{2.27}$$

Where b is the belief space, D is a finite set of vectors that are used to represent a value function and α refers to a vector inside this set. The vector α itself acts as a value function corresponding to a specific action a. The optimal set D' can be described by:

$$D' = purge\left[\bigcup_{a \in A} D^a\right] \tag{2.28}$$

$$D^a = purge\left[\bigoplus_{z \in Z} D^a_z\right] \tag{2.29}$$

$$D^a_z = purge(\tau(\alpha, a, z) | \alpha \in D) \tag{2.30}$$

Where $\tau(\alpha, a, z)(s)$ is the $|S|$ - vector given by:

$$\tau(\alpha, a, z)(s) = \frac{1}{|Z|} r^a(s) + \gamma \left[\sum_{s' \in S} T(s'|s, a) O(z|s', a)\right] \alpha(s') \tag{2.31}$$

The $purge(\cdot)$ operator is used to reduce a set of vectors to a unique minimum form, $\cup$ represents the union of the sets D^a for all actions a in the action space A, $\oplus$ represents the *cross sum* of the sets D^a_z for all observations z in the observation space Z (with a *cross sum* of two vectors given by $A \oplus B$ = $[a+b \,|\, a \in A,\ b \in B]$) and the $\setminus$ operator defines set subtraction [Cassandra et al. (2013)]. Calculating the set D^a_z can be done using equation (2.31) that attempts to calculate the value of τ based on fixed α vector, action a, observation z and state s. Once the sets D^a_z have been calculated, Algorithm 2 can be applied to do incremental pruning.

Algorithm 2: Incremental Pruning

```
1 Input D^a_{z_1},..., D^a_{z_k}
2 W ← FILTER(CrossSum(D^a_{z_1}, D^a_{z_2}))        ▷ See Algorithm 3
3 for i ← 3 to k do
4 |   W ← FILTER(CrossSum(W, D^a_{z_i}))            ▷ See Algorithm 3
5 end
```

Algorithm 2 works by firstly doing the cross sum of the first two sets $D^a_{z_1}$ and $D^a_{z_2}$, filtering the results and storing them in W (**line 2**). The process of filtering is used to remove the vectors that are dominated (also known as pruning). Following this, the algorithm continues to iterate over the remaining sets from $D^a_{z_3}$ to $D^a_{z_k}$ (**line 3**), doing the cross sum of the next set $D^a_{z_i}$ with W before being filtered (**line 4**). This repeats until all the sets D^a_z have been cross summed and filtered. The resulting set W is comprised of vector representations w that each are associated with a singular action a. The policy can therefore be obtained using:

$$\pi(a|b) = \arg\max_a \left[\max_w b \cdot W\right] \tag{2.32}$$

The filtering is done by Algorithm 3 to prune vectors in the set that are dominated by other vectors. The domination check is done by Algorithm 4, a linear program that solves for the belief state b where taking the dot product $(\cdot)$ with α results in the largest value from all vectors in A.

Algorithm 3: FILTER

```
1  Input set of vectors F
2  Initialise W to Null set        ▷ initialise a set to store dominating vectors
3  for each s in S do
4  |  w ← arg max_{φ∈F} e_s · φ
5  |  W ← W ∪ {w}                                  ▷ add to the set W
6  |  F ← F \ {w}                                  ▷ Remove from set F
7  end
8  while F != Null do
9  |  for φ in F do
10 |  |  b ← DOMINATE(φ, W)                        ▷ See Algorithm 4
11 |  |  if b == Null then
12 |  |  |  F ← F \ {φ}
13 |  |  else
14 |  |  |  w ← arg max_{φ∈F} b · φ
15 |  |  |  W ← W ∪ {w}                            ▷ add to the set W
16 |  |  |  F ← F \ {w}                            ▷ Remove from set F
17 |  |  end
18 |  end
19 end
20 return W
```

In Algorithm 3, the set W is filled with vectors w, regions where they have the largest values by taking the dot product of vector e_s and vector ϕ with $\phi \in F$ (**line 4 - 5**). The vector e_s has zeros at all positions except at $e_s(s) = 1$, acting as a representation of the final belief b of the agent for a state s. This acts as the starting place for the algorithm as each belief state b would have an associated vector ϕ (and therefore associated action a). The algorithm then iterates while F is not empty meaning that there are still vectors to be checked for domination (**line 8**). The vectors ϕ in F are then checked for domination based on the computed set W (**line 10**). If they are, they are excluded from the set F (**line 12**) and if they are not, they are added to the set W (**line 14 - 15**).

Algorithm 4: DOMINATE

```
1  Input α, A
2  L ← LP(variables: b, δ; objective:max δ)
3  for each α′ in A \ {α} do
4  |  ADDCONSTRAINT(L, b · α ≥ δ + b · α′)
5  |  ADDCONSTRAINT(L, b · 1 = 1)
6  |  ADDCONSTRAINT(L, b ≥ 0)
7  |  if L Infeasible then
8  |  |  return Null
9  |  else
10 |  |  (b, δ) ← SolveLP(L)
11 |  |  if δ then
12 |  |  |  return b
13 |  |  else
14 |  |  |  return Null
15 |  |  end
16 |  end
17 end
```

In Algorithm 4, due to α being a vector and A a set of vectors, there could be a

number of values of the belief b where α is the maximum value. By upholding the constraints given in **lines 4 - 6** when calculating the value of L, a valid belief b can be found, if one exists. The function `LP` denotes linear programming and the function `SolveLP` denotes the process of solving that linear program. Linear programming is a method that finds a solution based on the requirements given by linear relationships [De Farias and Van Roy (2003)].

2.4 Conclusion

This chapter introduced the field known as reinforcement learning, a framework of learning through interaction with an environment. Markov Decision Processes were presented - models that formalise reinforcement learning problems. These models assume the decision maker (agent) knows with complete certainty the current state and so finding the correct associated action for the state is a matter of solving for the optimal *state-value* or *action-value* functions. These are used to find policies, the solutions to Markov Decision Processes. The commonly used Q-learning algorithm was presented and explained as well. The theory was then expanded to Partially Observable Markov Decision Processes, a variation that can deal with uncertainty as it maintains a belief distribution over all states instead of having access to the current state. The belief update that changes the agent's belief based on its selected action and received observation is presented along with the value function of Partially Observable Markov Decision Processes that is used to find solutions. The incremental pruning algorithm is presented as one common case of a Partially Observable Markov Decision Process solver.

Chapter 3

Methodology

The overall goal of this work lies in developing a system capable of inferring the intention of a human to allow a robot to collaborate with that human. This chapter presents the methodology to design, create and execute this system by using the reinforcement learning theory from Chapter 2.

Section 3.1 presents the human's behaviour in their environment and the considerations made in modelling them. Section 3.2 lists the considerations of introducing the robot into the environment. Section 3.3 examines the interaction process between the human and robot. Section 3.4 introduces the idea behind using Partially Observable Markov Decision Processes (POMDPs) for intention inference. Section 3.5 presents the design of the POMDP, with section 3.5.1 detailing a method to manage the scale of large POMDPs by approximating the functionality using smaller models. Section 3.5.2 presents an alternative design of the POMDP model with its own advantages and disadvantages. Section 3.6 presents the manner in which these POMDPs will be executed in practice. There is a running example of a human and a robot throughout this chapter that is used to precisely illustrate the different sections of the methodology.

3.1 Human Behaviour in an Environment

In this work, a *task* is defined as a problem that requires a specific behaviour to solve it while a *behaviour* is defined as a number of sequential actions whose execution would complete the task (although behaviours that do not complete tasks can also exist). For example, a coffee making task has the following behaviour: turning a kettle on, fetching a mug, opening coffee tin, fetching a spoon, scooping coffee powder into the mug twice, pouring hot water from the kettle into the mug, fetching milk, pouring milk into the mug and stirring the mixture with the spoon thus completing the task. Removing the tools or ingredients used, the above behaviour could be described by primitive actions such as *pickup*, *use*, *put down* and directional movement. The behaviour to complete a task is the human equivalent to the policy a reinforcement learning agent would use to complete a task, named as such to distinguish between them.

A single human has a number of tasks that they need to complete in their environment in their everyday life. They construct a mental plan, their internal proposal for completing their tasks including the required behaviours (and subsequently actions) for these tasks. This mental plan therefore encapsulates all possible behaviours that the human has for all their tasks called their intentions. Mental states cannot be observed directly by any outside actor, only the human knows with complete

certainty their own intention unless they explicitly communicate it to another actor, which the human has no need to as they are the only actor in their environment. Once the human is satisfied with their mental plan, they follow it, selecting a task and executing the necessary actions as given by the behaviours. The human is assumed to be able to complete all tasks without any assistance.

The human develops a consistent mental plan that they follow to complete their tasks with either a strict or variable order depending on some conditions (potentially following a different order depending on the day of the week for example). An assumption is made that the human refuses to adapt their behaviours and so will accept assistance in the form of another task being completed as long as it does not impact their mental plan by disturbing task order or changing the actions the human will execute for their current task. In the absence of verbal communication, the human's performed actions are the most accurate pieces of information available to infer their intention. However some parts of behaviours are not unique to specific tasks as they use the same primitive actions (and sequences) introducing uncertainty as to the true intention of the human. An example of this is the making of coffee or tea - the behaviours are almost identical except for the primitive action sequence that controls which substance is added to the mixture (teabag or coffee powder).

In this research, the environment of the human is simulated using a gridworld - a rectangular sized grid of $m \times n$ blocks. Inside are located a number of objects that must be collected by the human with the process of moving to and collecting an object being called a task. The human can move in the directions of North, East, South and West and interact with objects by collecting them. The environment is considered complete when all tasks are completed - that is when all objects have been collected. This means that the environment itself is technically a task composed of many sub-tasks. The uncertainty arises from the fact that objects may be located in similar positions and require similar methods making distinguishing them difficult.

Running Example: Sample Gridworld

Figure 3.1 shows the *Sample* gridworld chosen to illustrate the human's capabilities. The human's starting position is the middle of the grid (denoted by the **H** and the colour blue) and is tasked with collecting the four labelled objects (denoted by **A**, **B**, **C** and **D** respectively and the colour red). The five available actions to the human include `up`, `down`, `left`, `right` (represented by arrows) and `collect` (represented by a cross). The first four actions would move the human in the corresponding direction while action `collect` collects the object, leaving the human in the same location. From this starting position, this human has their chosen behaviours to collect each object given in Figure 3.2. These behaviours are the preferred action sequences that this specific human uses to complete the tasks from their starting position. If this grid approximated the coffee making task - objects **A**, **B**, **C** and **D** could denote the boiled water, coffee powder, sugar and spoon respectively. If the human started holding a mug, they could simply move to collect the spoon and afterwards move about in any order to collect the remaining objects, eventually making their coffee.

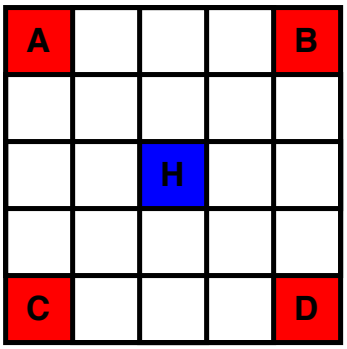


FIGURE 3.1: *Sample* gridworld

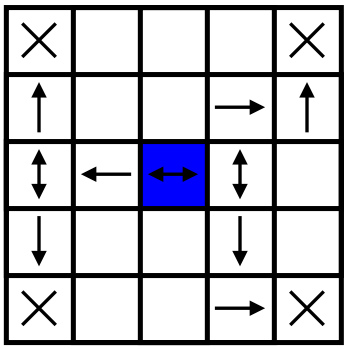


FIGURE 3.2: The human's behaviour to collect objects

3.2 Robot Behaviour Based on the Human

A robot is introduced into the same environment with the same human with the aim of assisting in completing the same tasks. For this robot to collaborate with the human on their tasks, it needs a common plan to be developed so it can provide efficient collaboration and not hinder the task completion. To prevent the human having to plan around the robot and therefore spend more time forming a plan or adapting behaviours, the robot needs to accommodate the human, making this fall into the category of on-the-fly collaboration. The manner that the human functions is dependent on tasks while the manner that the robot functions is dependent on the inferred intentions of the human.

The robot is split into two modular components known as the *thinking* component and the *acting* component, with the core focus of this research lying on the *thinking* component of the robot. This component needs to reason about the human's behaviour by inferring their intentions towards tasks as it allows the robot to act proactively as opposed to reactively once it has enough information about the human's intention. This enables better collaboration from the robot as it anticipates the correct responding actions while the task is being completed instead of waiting for the completion of the task and then responding.

To be able to infer the intentions of the human based on the behaviours they are using for their tasks, the robot needs to know the set of behaviours that could be executed by the human. Therefore these behaviours need to be learned (or modelled) by the *thinking* component in an earlier offline stage prior to introducing the robot into the environment. The assumption is made that the human's behaviour is stationary and so the robot should avoid disturbing the human as disturbing the human could cause them to adapt their behaviours for tasks online, rendering the model unable to be used. The robot needs to be inserted into the everyday life of any human to collaborate by completing tasks and therefore reducing the overall work (of the human) and completion time. This component is created using the POMDP framework and is covered in section 3.5.

While the *acting* component of the robot is responsible for enabling the robot to complete the tasks it selects, this component is assumed to complete any task satisfactorily as the sole focus of this research lies on the *thinking* component -

the intention inference. Tasks can be simple such as *"Pick up object"* to complex such as *"Prepare breakfast for two people"*, meaning that the robot needs to not only learn primitive actions for completing tasks but also develop skills that group these primitive actions together. The *acting* component can be developed using hierarchical reinforcement learning (HRL), an extension to reinforcement learning that aims to solve complex problems by breaking them up into a set of interrelated problems that are solved separately and then combining the solutions to create a solution to the original complex problem [Hengst (2012)]. Alternatively the use of inverse reinforcement learning can also lead to the development of the skills needed to complete tasks as it aims to recover a reward function based on observed behaviour [Ranchod et al. (2015)]. Designing the *acting* component is left as a future recommendation as generally, a robot only needs to be perform actions well enough to complete a task, even if the actions are not optimal.

Decoupling the intention inference and the actions (and skills) needed to complete tasks into *thinking* and *acting* components respectively allows these to be improved or replaced in isolation to the other component. The *thinking* component has both an offline and online phase. The offline phase involves the construction of the POMDP model and solving it to obtain a policy while the online phase involves using this model and policy to infer the human's current intention.

Running Example: Inserting Robot into Environment

A robot is introduced into the *Sample* gridworld shown in Figure 3.1. To isolate the effects of the *acting* component, it is assumed that the human and robot cannot physically collide and the robot starts in the same location as the human. Relaxing these two assumptions is a matter of configuring the *acting* component without changing anything about the *thinking* component. The robot has the same objects to collect as the human and needs to prevent itself from disturbing the human by attempting to collect the same object that the human is currently attempting to collect. For this reason it needs to be able to infer the intention of the human towards the current task. Should there be an order to the collection of the objects, the robot would need to also follow this order. For example if the human always collects objects in the order **A**-**B**-**C**-**D** and the robot infers that the human is currently collecting object **B**, then it would need to collect object **C** to uphold the order.

3.3 Human and Robot Interaction

The interaction between the human and robot can be described by the relationship of an *observer* and the *observed* or *subject*. The *observer* entity (robot) watches the *observed/subject* entity (human) to reason about their behaviour. As the human has a mental plan that was established in isolation from the robot, the human is driving the interaction - that is the human completes tasks according to their mental plan and the robot aims to assist them.

Once the intention of the human is inferred, the robot selects a task to complete, therefore the collaboration is enabled based on the inference. In the case of incorrect inference, the robot may select an incorrect task (one completed previously or that does not conform to the required order) that then hinders the collaboration effort as

the human would need to indicate to the robot that it has made an error. In reality, the human cannot be cognisant of the robot all of the time and so by the time the human indicates that the robot has made an error, the robot potentially has spent some time performing the wrong actions that can negatively impact the human and their task completion making correct intention inference vital.

When the human completes a task, it is assumed they can observe the status of all possible tasks, whether they are completed, in-progress or uncompleted and select their next task based on this information. The robot, once it finishes a task, again observes the human, reasoning about their behaviour and upon inferring their new task intention, selects a new task to complete. This repeats until all the tasks have been completed in the environment.

3.4 POMDPs for Intention Inference

Humans execute a sequence of actions (behaviours) to complete tasks. A robot must be able to infer the intention of the human based on the executed human actions for their current trajectory. Even if all action trajectories are known through modelling, matching the human's executed actions to infer the current task they are completing is difficult as action sequences can be shared by multiple trajectories. Therefore the uncertainty between multiple trajectories must be reduced by reasoning about the actual (or true) current trajectory of the human. Consider the three arbitrary action trajectories shown in Figure 3.3 that aim to move towards (given by directional arrows) and collect (given by the cross) a specific object.

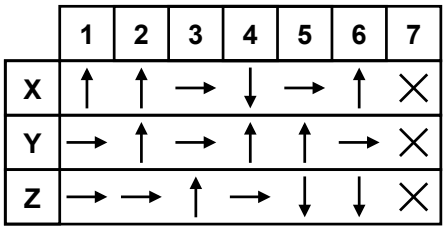


FIGURE 3.3: Three arbitrary action trajectories

The three trajectories all represent a different task but they all share similar action sequences to one another. Depending on the actions observed, one or more of these trajectories are possible behaviours of the human, illustrated by Figure 3.4. The green highlighted actions indicate the sequences that match the observations - these are the behaviours (and subsequently the tasks) that are being inferred. For the first observation, all three trajectories have the up action and so all three of them are candidates for the human's intention. The second observation now creates a two action sequence to compare to each trajectory although it does not lower the uncertainty - all three tasks are still possible candidates. The third observation eliminates trajectory **Y** as the current intention of the human as the observed action sequence no longer matches this task. The fourth and final observation creates a sequence that only trajectory **X** matches and thus this is the current task the human is completing.

Partially Observable Markov Decision Processes (POMDP) are decision processes whereby the agent does not have direct access to the state information, instead

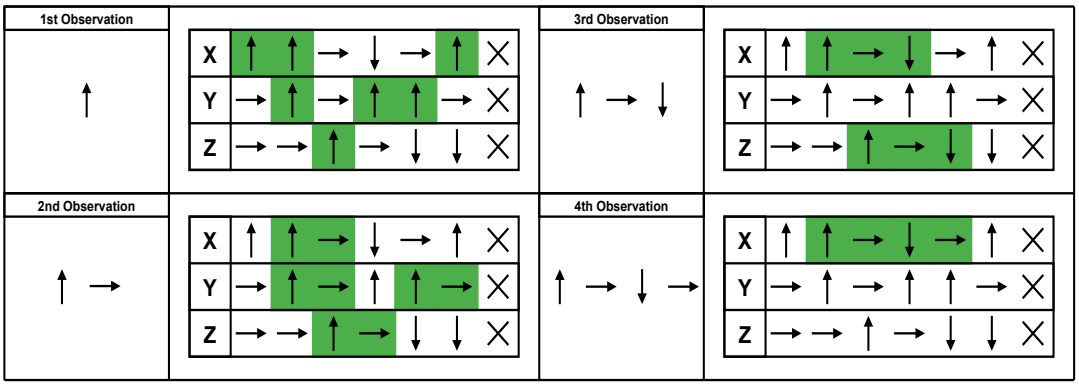


FIGURE 3.4: Trajectories indicated based on observed actions

holding a belief as to the current state and selecting actions based on this belief, receiving observations that can be used to update its belief to be more accurate as to the current state. They are suitable for situations whereby there is uncertainty or some hidden information must be found through reasoning and can thus be used to infer the human's intention. As the human continues to execute actions sequentially, the agent would observe these actions and use them to update its belief as to which intention or task the current state belongs to. Once it reaches a suitable level of confidence, it will indicate the task that is the human's intention.

We now introduce the **Action Trajectory Encapsulated Partially Observable Markov Decision Process** or **ATE-POMDP** that has a model based on the action trajectories of the actor that is being modelled (human in this case). By modelling all the trajectories in their entirety, that is all the actions, the model can detect the underlying state and intention at any point in the human's behaviour subject to the actions observed and how differentiated the trajectories are. Encapsulating all these trajectories with a model is similar to the common method of holding a *plan library* - an outline of all the possible goals of a modelled agent - to be able to reason about the goal of that agent [Albrecht and Stone (2018)]. This is the POMDP that can be used to satisfy the first research question. Knowing the human's intention and knowing the remaining tasks, the robot could then select one of these tasks to complete, thus planning its behaviour around the human's, satisfying the second research question. Since it models the different behaviours of the human, if the human eventually has new tasks or refines a behaviour for a task, the model needs to be augmented with this new task.

3.5 Offline: ATE-POMDP Design

Formally, a human has n tasks with associated action trajectory t_i with $i = 1...n$. The model design is restricted to only using actions as observations, although including other contextual information (such as locations, tools or materials) can be included to have a more descriptive observation function that aims to use more information to differentiate action trajectories from one another. The goal of the agent is to indicate the correct intention by firstly reducing uncertainty as to the underlying state as each state represents part of an intention of the human.

Tasks can be identified based on the received observations. As was shown in Figure 3.4, it is difficult to find the true intention after only one observation and so we expect the accuracy of the inference to improve with more observations. The agent itself has n *indicating* actions (corresponding to the n tasks of the human) and a single *information gathering* action. The information gathering action is used to obtain more observations without indicating the task of the human, while the indicating actions establish the human's intention and are terminal resulting in the completion of an episode. The reason for being terminal is that this marks the end of half of the *thinking* component of the robot - it has determined the intention of the human and now that information must be used to select a task (other half of the *thinking* component) before passing control over to the *acting* component to complete the selected task.

The ATE-POMDP agent starts in an unknown state s with belief b with this belief being a distribution over the possible intentions of the human. Information gathering action `Wait` will transition the agent to the next state and receive a human action as an observation - this is equivalent to waiting for the human to act to observe their action. The agent's other actions are used to indicate the intention that the human must be following. They are called indicating actions as they specify the current task of the human such that this information can be used externally to the ATE-POMDP to update the robot's knowledge of the remaining tasks and to select a task to complete itself. The state transitions are deterministic to model the human moving with complete certainty in regards to the task they want to complete and their actions being successful at each time step towards completing that task. This means they will not change their minds halfway through the current task to another new task nor will they make a mistake when it comes to completing their current task that will cause them to deviate from their known behaviour. After reaching a belief b whereby the agent knows with complete certainty the human's intention, it is indicated, thus ending the episode for the agent. The idea of waiting was chosen as this will leave the human undisturbed and is the most passive possible behaviour of the robot that will not hinder the human. The human, having executed their last action and completed their task, will now select a new task at which point the ATE-POMDP must be re-initialised to infer their new intention.

An Action Trajectory Encapsulated Partially Observable Markov Decision Process is defined by:

- **S** - The state space represents the progress of the human in a task's action trajectory. The states are therefore representations of the human executing a certain action in a particular action trajectory with the entire state space acting as a progress indicator across all trajectories. This model allows the human's intention to be inferred at any point in any trajectory. The state space **S** has a size equal to the total number of actions from all trajectories. If n is the number of tasks with associated action trajectory t_n, the function $A(t_n)$ gives the number of actions of task n. Thus the number of states is equal to:

$$|S| = \left[\sum_n A(t_n) + 1 \right] + 1$$

  The additional state per trajectory t_n acts as the starting state for the agent prior to receiving any observations - the human has not yet executed an action.

There is one additional state that acts as the terminal state for the entire model.

- **A** - The action space is a set of actions that are used to indicate the current intention of the human with one additional action that is used to gather information. Intention refers to a task that an action trajectory is associated with. The action set **A** has a size equal to $n + 1$ where n is the number of tasks that the human has. There needs to be as many actions as there are tasks to be able to indicate the current intention that the underlying state is part of. The additional action `Wait` is used to gather more observations if the agent is uncertain.
- **T** - The transition function defines deterministic state transitions. If the agent needs more information, it performs the information gathering action `Wait` (waiting for the human to execute their next action) and transitions to the next state where the human's action is received as an observation. Intention indicating actions move the agent to the terminal state. Gathering information in the state whereby the last observation has been received will transition to the terminal state as this means the human has finished their task and there are no more actions to observe.
- γ - The discount factor. It has a value of 1 so rewards will not decay over time.
- **R** - The reward function gives a penalty for every information gathering action to discourage gathering more information than needed and taking too long. Positive rewards are given when the correct indicating action is executed by the agent - it has determined the true intention of the human. Negative rewards are given for indicating the incorrect intention.
- **Z** - The observation set is equal to the overall number of distinct actions from all trajectories.
- **O** - The observation function defines deterministic observations given transitioning to state s' from state s having selected action a.

Running Example: Constructing the ATE-POMDP

To illustrate the ATE-POMDP, it will now be designed based on the human in the *Sample* gridworld given in Figure 3.1. The action trajectory for only object **A** in this environment is given by Figure 3.5.

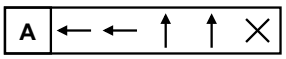


FIGURE 3.5: Action trajectory for object **A**

The **A** trajectory is made up of 5 consecutive actions: `left`, `left`, `up`, `up`, `collect`. There are only three unique actions - `left`, `up` and `collect` - that make up the observation set Z for the ATE-POMDP created specifically for only this trajectory. There are a total of 7 states (5 human actions plus 2 states, the start and terminal states) and 2 actions (the information gathering action `Wait` and the only indicating action possible `Human_A` as this model only includes a single action trajectory). The full ATE-POMDP for this single trajectory is given in Figure 3.6.

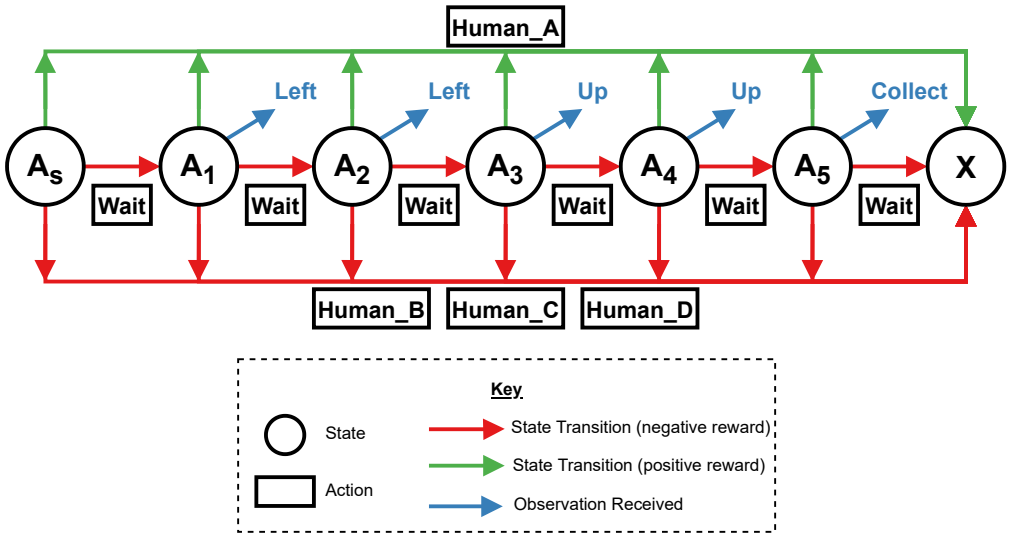


FIGURE 3.6: ATE-POMDP model based on trajectory **A**

Each circle represents a state of the ATE-POMDP with $\mathbf{A_S}$ representing the starting state of the agent and the cross ($\times$) representing the terminal state. The agent's actions include `Human_A`, `Human_B`, `Human_C`, `Human_D` - four actions used to indicate the human's underlying intention for a task in the *Sample* gridworld - and `Wait`, the information gathering action. There should be one indicating action per trajectory and since only the **A** trajectory is used, only the `Human_A` action should be present, however all indicating actions for each possible intention are shown for completeness to show the effects of executing the incorrect indicating action. For this particular ATE-POMDP model, the optimal policy would be to execute action `Human_A` regardless of the belief - it is the only action that results in a positive reward. The full ATE-POMDP design (for all 4 action trajectories) of *Sample* gridworld is given in **Appendix A**. These indicating actions are the driving force behind the inference - executing these actions is equivalent to the robot deciding that the intention referenced by the human's actions is the true intention and based on this information, the robot can now collaborate as it has found the intention of the human and it can now select its own task to complete and the *acting* component can then be used.

3.5.1 Dealing with ATE-POMDP Scaling

Generally, a human does not need to return to their starting location for every task - they would simply select their next task and move towards that new object based on the ending location of the previous task. This changes the point of view of objects as seen relative to the human depending on the task completion order. For example, the human starts in a position whereby object **A** is to their left and object **B** to their right. By moving left to collect object **A**, the relative location of **B** has changed for the human - it is now behind them and so to collect object **B**, the human must now move backwards (or turn around and move forward). This changing point of view relative to the human is known as the human's perspective and it is based on their position in the gridworld. While the location of the remaining objects has not changed, the human's new location needs its own behaviours to reach and collect the other objects.

All these perspectives give rise to a large number of possible action trajectories for the same task. This leads to a large state space that can quickly scale the ATE-POMDP model to intractability if all of these perspectives are combined into one ATE-POMDP model as then the model needs to encapsulate the movement to any object from any other object. The belief space itself will now initially consider all action trajectories for all tasks at the same time and that will require many observations and actions until the belief space is updated to reflect the true underlying state (and therefore human intention) with complete certainty. This means that the agent would consider action trajectories for tasks that may be blocked until other tasks are completed prior. The functionality of this large ATE-POMDP can be approximated by splitting the ATE-POMDP into multiple ATE-POMDPs based on grouping action trajectories together based on perspective, alleviating the complexity of solving one large POMDP model. Each smaller perspective ATE-POMDP can be initialised correctly by having the indicating action of the previous model act as the identifier as to the next ATE-POMDP model to be initialised based on perspective - achieving the same autonomy that would have been provided by one large ATE-POMDP model.

Running Example: Multiple Perspectives of the Human

As the human does not return to their starting location in the *Sample* gridworld, the human's possible different perspectives need to be considered, requiring ATE-POMDPs designed for them. There are as many perspectives for a human as there are tasks, with an additional one for the starting location. Figure 3.7 shows the action trajectories of the human (their behaviour to collect the remaining objects) from the four new starting locations, based on whether they collected object **A**, **B**, **C** or **D** first.

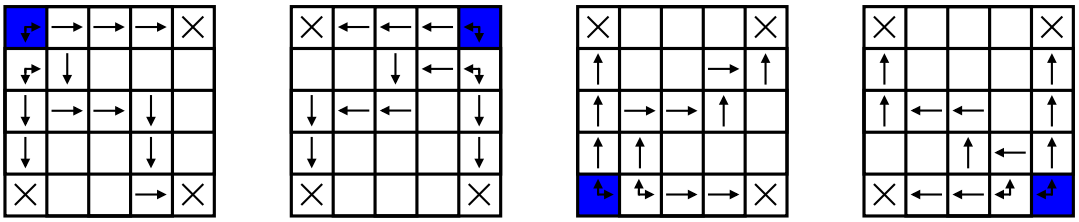


FIGURE 3.7: Task trajectories from perspective **A**, **B**, **C** and **D**

The four objects (**A**, **B**, **C** and **D**) can each be collected by 3 different action trajectories shown in Figure 3.7 with an additional action trajectory based on the starting location given by *Sample* gridworld (Figure 3.2). This means that a total of 16 different trajectories need to be modelled in the ATE-POMDP leading to a state space of 96 states - already computationally intractable based on the solution technique given in section 2.3.3. Creating five ATE-POMDPs based on the different perspectives results in four ATE-POMDP models with 19 states each and one model with 20 states that can be solved to obtain a policy. These five ATE-POMDP models function in the same way as the large ATE-POMDP except being computationally tractable. Avoiding one large ATE-POMDP model is further motivated based on whether the human has multiple starting locations and this is shown in **Appendix B**.

3.5.2 Static ATE-POMDP Design

The ATE-POMDP attempts to infer the human's intention for a task by observing the human's actions as evidence for their current intention. The uncertainty in the inference comes from the fact that multiple action trajectories share similar action sequences, making it difficult to establish the current trajectory. The more observations, the more the uncertainty is reduced. In Figure 3.4, by the third observation, two trajectories are still possible candidates that the human was following. However, by that third observation, trajectory **B** has been eliminated as a possible candidate for the human's intention. Instead of requiring a fourth observation to establish the human's intention, the robot could simply have chosen to complete the task it was certain that the human was not completing - task **B**.

We now propose an alternative method in using an ATE-POMDP based on the fact that it is quicker to eliminate tasks not being completed by the human than finding tasks that the human is completing. The model remains the same as described in the previous sections except that the *indicating* actions are replaced with *assisting* actions with the reward function being adjusted for these actions. The assisting actions explicitly dictate the task that the robot should complete itself as opposed to the indicating actions that are used to specify the human's current intention so this information can be used externally to the ATE-POMDP model. For the sake of clarity, the original ATE-POMDP model is called the *dynamic* version with designation (D) while the alternative ATE-POMDP model is called the *static* version with designation (S). The reason for this naming convention is that the dynamic ATE-POMDP can infer the intention of any combination of tasks at the cost of taking more time and observations while the static ATE-POMDP can only infer the intention for certain combination of tasks using less time and less observations.

The dynamic ATE-POMDP or ATE-POMDP (D) uses its actions to indicate the intention of the human once the uncertainty has been reduced using its updated belief space b. Once the intention is indicated, this information is used by the robot to eliminate the current task from the internal list of the possible tasks of the human could be completing. With the list being reduced, the robot would then select a task to complete. Since the robot has a choice as to the task it can select to complete - the next set of ATE-POMDP models (based on perspective) need to include every possible task trajectory that can scale the models making them larger and harder to solve. By including every task trajectory, this robot has the ability to infer the human's intention for any combination of tasks.

The static ATE-POMDP or ATE-POMDP (S) uses its actions to assist the human by grouping trajectories with similar action sequences under the same assisting action through the reward function. This means that even though the agent may be uncertain between a number of trajectories, if they all have the same assisting action - it can be executed without the need to further reduce the uncertainty. As observations are received and uncertainty is reduced, the agent attempts to assist the human based on its internal preference of tasks to complete. If the human's current intention is not indicating the first preferred task, then this agent will select that as the task to complete. This use of ATE-POMDP models restricts the possible combinations of tasks that the human can do hence being called static. For example the agent could always select task **B** if the human never shows intention of completing it. This would result in the trajectory of task **B** being absent from any

subsequent ATE-POMDP models for the different perspectives as it will always be complete from the human's new perspective.

The ATE-POMDP (D) is easier to implement since all action trajectories can be included and this allows it to infer the intention of the human for all task order variations whereas ATE-POMDP (S) needs more care in its implementation to make sure that the specific combinations are covered by the ATE-POMDP (S) models based on the perspectives later on in the collaboration setting. This also means that the ATE-POMDP (D) has more states and actions making models larger and more difficult to solve compared to the ATE-POMDP (S) that has less states and actions. The ATE-POMDP (S) also functions in less time steps than the ATE-POMDP (D) at the expense of not being able to handle all task order variations, meaning that if the human has a preferred order of tasks, the ATE-POMDP (S) can not be used to infer the human's intention to select the next task that adheres to that order.

3.6 Online: ATE-POMDP Execution

Having obtained ATE-POMDP models, they now need to be used to infer the human's intentions. Applying the models effectively requires a simulation standard - a set of rules that dictate the interaction with the ATE-POMDP and to dictate the interaction between the human and the robot.

3.6.1 Interacting with the ATE-POMDP

The simulation standard for interacting with the ATE-POMDP is identical to the one utilised by the POMDP framework in [Egorov et al. (2017)]. This POMDP standard defines a simple method to initialise the model and interact with it at every time step. The simulation standard is given in Algorithm 5.

Algorithm 5: ATE-POMDP Simulation Standard

```
1  initialise ATE-POMDP                          ▷ model created offline
2  initialise π                                  ▷ policy solved offline
3  initialise γ
4  s ← s_0                          ▷ chosen according to current intention
5  b ← b_0                          ▷ distribution over possible intentions
6  r_tot ← 0
7  while episode do
8      a ← π(b)                                  ▷ action selected using policy
9      s', z, r ← ATE-POMDP(s, a)        ▷ executing action in ATE-POMDP model
10     b ← Update(b, a, z)                       ▷ belief updated using b, a, z
11     r_tot ← γ * r
12     s ← s'
13 end
```

The simulation standard proceeds as follows: The ATE-POMDP model and its policy π are initialised, having been both created in the offline phase (**lines 1 - 2**). Depending on the current discrete point in time of the intention, the underlying state s_0 is set as s (**line 4**) along with an initial distribution for the possible intentions

of the human (**line 5**).

The interaction continues until the terminal state is reached. Utilising policy π and the current belief b, the best action a for the agent is found and executed. This results in the transition to the underlying state s' (unobservable intention of the human) and returning the observation z and the reward r to the agent. The agent updates its belief b and stores its reward r. The reward value is generally unused except as a metric to evaluate the effectiveness of the ATE-POMDP model itself - low reward means that the model may be using a sub-optimal policy. Note that while the variables s and s' are shown in the Algorithm 5, the agent still does not have access to this information. Using the observation z, the action a and the belief b, the belief is updated. The next iteration then begins with the agent using their new belief with the policy to find their new action and executing it. This continuously loops until eventually the terminal state is reached.

3.6.2 Human-Robot Collaboration Simulation Standard

The human-robot collaboration standard is the manner in which the two teammates will act to collaborate in completing the tasks. The human drives the interaction by going first, allowing their actions to be observed. The simulation standard is shown in Figure 3.8, with both teammates being their own separate entities. For ease of simulation, they are assumed to both be comprised of *thinking* and *acting* components. The *acting* component could use different action trajectories for the human and robot to complete the same task although this research assumes that the action trajectories are the same to complete the same task. The *thinking* component is whereby they differ with the human selecting tasks based off of a list that details the available tasks in the environment while the robot uses the ATE-POMDP to infer the human's intention and then selecting tasks itself.

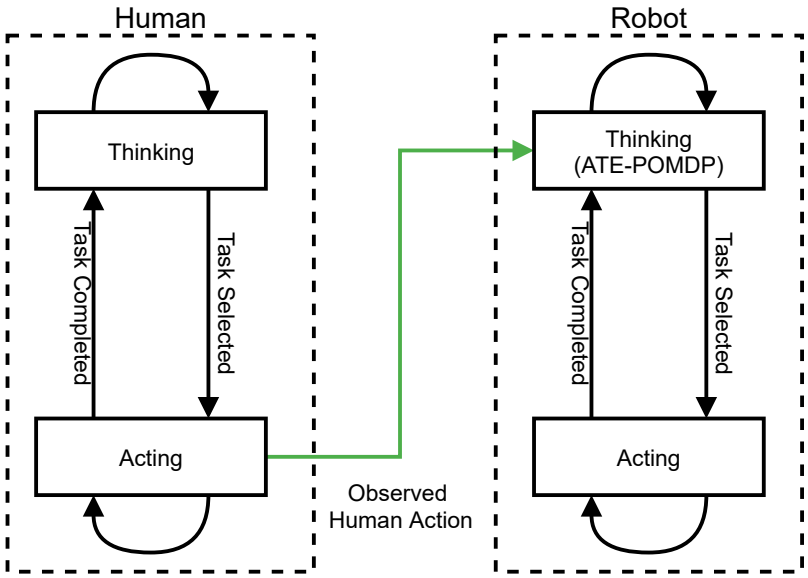


FIGURE 3.8: Human-robot collaboration simulation standard

The HRC simulation standard in Figure 3.8 works as follows: The POMDP and human are both initialised as standalone actors. The human selects tasks in their *thinking* component which are then completed by the *acting* component executing

the required actions. The robot initialises the ATE-POMDP in the *thinking* component based on the simulation standard given in Algorithm 5. Once an indicating (or assisting) action is executed and the agent estimates the current intention of the human, a task is selected to be completed with control being passed to the *acting* component that executes the necessary actions, one per time step, to complete the chosen task. The green line represents the human action information that the robot is observing using its ATE-POMDP (although this is encapsulated by the model, there is no direct communication between the human or the robot). Once tasks are completed, the control is passed back to the *thinking* component and after a new task is selected, the *acting* component is again used. This repeats until all tasks have been completed.

3.7 Conclusion

This chapter described the human and their behaviour in their environment to solve tasks. Introducing a robot into the same environment requires a model of the human before it can infer the human's intention for a task so that it can assist them. The robot will observe the human's actions and infer their intention for a task, then select another task to complete itself. This is accomplished through the use of an Action Trajectory Encapsulated Partially Observable Markov Decision Process (ATE-POMDP) that has a design based directly on the action trajectories of the human. Due to computational constraints as these ATE-POMDP models scale, we segment the model into multiple smaller models based on the perspective of the human, that is their changing point of view based on their movement in the gridworld but achieving the same functionality that the large ATE-POMDP model has. Two versions of the model are described - the ATE-POMDP (D) with its ability to infer the intention of the human based on any combination of tasks at the cost of more time steps and the ATE-POMDP (S) with its ability to infer the intention of the human based on a subset of combinations of tasks using less time steps.

Chapter 4

Experiments and Discussion

This chapter details the experiments conducted to evaluate the use of the ATE-POMDPs as the intention inference mechanism in enabling human-robot collaboration. Section 4.1 presents the setup of the simulations of the ATE-POMDPs with Section 4.1.1 explaining the metrics that will be collected from each simulation to evaluate each team and Section 4.1.2 detailing the different teams that will be used in the simulations. Section 4.2 presents the various experiments simulated, the *Sample* gridworld, Alternative gridworlds and the Larger gridworlds in section 4.2.1, section 4.2.2 and section 4.2.3 respectively. Section 4.3 presents the discussion around the results of the experiments.

4.1 Experimental Setup of Simulations

The designed ATE-POMDPs need to be tested in simulation to determine their effectiveness for intention inference and whether this inference will aid and allow the robot to collaborate with a human. Specifically in these experiments, the intention of the human is defined as the human's desire to move to and collect a specific object. Therefore by finding the human's intention using the ATE-POMDP, the robot can then select and complete another task and the more tasks there are, the more intentions that can be inferred. As the number of unfinished tasks decreases, correct intention inference is required as it allows the robot to know the tasks available based on the tasks it has completed itself and the tasks the human has completed.

Due to the computational constraints of POMDPs, the chosen gridworlds for the simulations will be smaller in size as the location of objects relative to one another determines the size of the ATE-POMDP model. All experiments are implemented using the *Julia* programming language using the framework called *JuliaPOMDP* that provides a number of POMDP specific features [Egorov et al. (2017)]. The chosen POMDP solution algorithm is incremental pruning implemented in the *POMDPSolve* software [Cassandra (1998)]. The experiments adhere to the simulation standards given in chapter 3.

Each simulation is comprised of a number of tasks to be completed with these tasks taking the form of collecting objects throughout a grid. The actions available to both the human and the robot include movement directions of `up`, `down`, `left`, `right` that move the specific actor in that direction and the action `collect` that picks up an object when the actor occupies the same location as the object. Collecting all objects means that all tasks are completed and thus the episode ends. Time steps are a fixed

discrete duration of time that the human and robot either use their *thinking* or *acting* component. The human *thinking* refers to the human selecting their next task while the robot *thinking* refers to reasoning about the intention of the human using the ATE-POMDP.

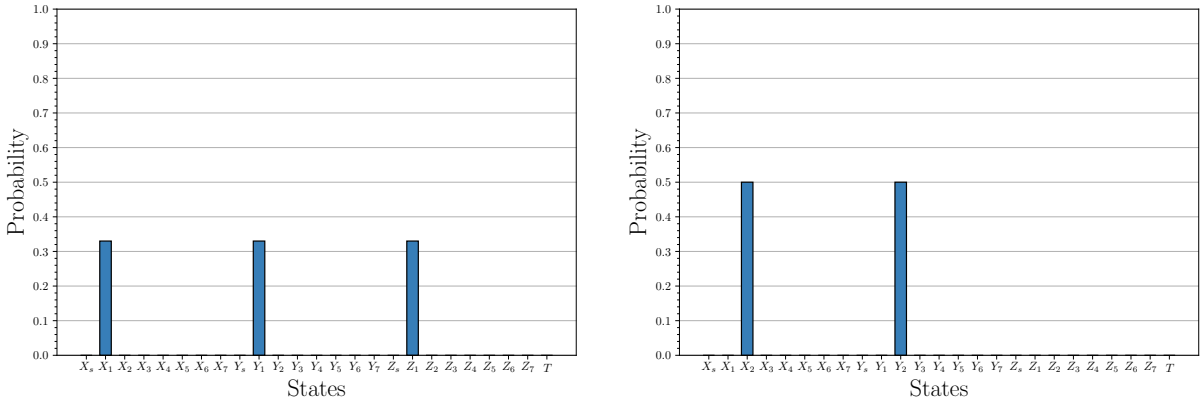
In each simulation, the human is assumed to be able to determine the completed tasks (even those complete by the robot) or in-progress tasks in a single time step. The human does this to infer the robot's intention such that they do not select tasks that the robot is busy with or has already completed. In reality, it is reasonable that the robot announces its intention to the human so that they would know the task the robot is completing or has completed. Both actors are assumed to start and finish tasks in their entirety - they will not select another task halfway through. The human selects new tasks in one time step while the robot selects tasks in the same time step when it has estimated the intention of the human using the ATE-POMDP.

Tasks collisions exist in the form of the robot and the human selecting the same task or a task that has already been completed. Since the human has to act first in order to establish the relationship between the *observer* and the *observed*, these collisions can only be caused by the robot inferring the incorrect intention as it would select its task after the human. Should task collisions occur, the human wastes a time step warning the robot and losing time to complete their own task while the robot spends that same time step selecting a new task or preparing to observe the human again. This means that the more task collisions that occur, the longer the completion of all tasks would take, extending the duration of the episode. In simulation, this warning takes the form of a flag that is checked at every time step by the robot and if this flag is set, it re-initialises its *thinking* component to select a new task based on the inference it has already done.

The ATE-POMDP models themselves are tested under two different belief spaces referred to as $b_{partial}$ and b_{full}. The belief $b_{partial}$ refers to a belief distribution over only a subset of all possible states while the belief b_{full} refers to a belief distribution over all states. The belief reduction in the $b_{partial}$ space comes from the fact that even though the human has n tasks, the received observation represents a single action from each of those n tasks thus only that subset is considered when initialising the belief distribution as opposed to the entire state space. It assumes that a human can only execute a single action at a point in time from each task - not every action from every task at a point in time. The belief b_{full} refers to the fact that the agent will initially be given a belief distribution over every possible state with the idea that the human could potentially be executing every action as the agent does not know the current point in time that the human is executing actions. This is much more realistic as initially, every action has the same likelihood of being executed by the human, making all intentions equally as likely.

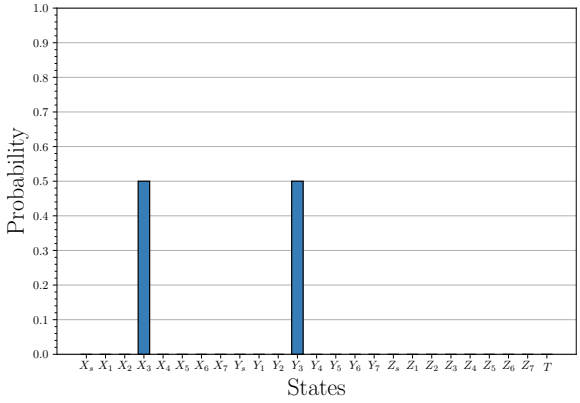
An example of the different belief spaces and their effect on the ATE-POMDP is illustrated using the action trajectories shown in Figure 3.3. An ATE-POMDP designed for these three action trajectories would be comprised of 25 states with states associated for a trajectory denoted by X, Y and Z in Figure 4.1. The symbols X_s, Y_s and Z_s denote the starting states and the terminal state is given by T. The underlying true starting state based on the human's intention is X_1. The initial belief distribution $b_{partial}$ is based on the subset of possible states X_1, Y_1 and Z_1, each with probability 0.33, given in Figure 4.1a. The observations are returned by executing

the action `Wait`. The first observation `up` updates the belief b to the distribution shown in Figure 4.1b while the second observation `right` updates the belief b as shown in Figure 4.1c. The reason that there is no further reduction of uncertainty among these two possible states after two observations is due to sharing a similar action trajectory. The third observation `down` updates the belief b to a probability of 1 for state X_4 as shown in Figure 4.1d. This would be the true underlying state of the human since they started in state X_1 and three time steps passed (given by the three observations) and thus the human's intention is task **X**.

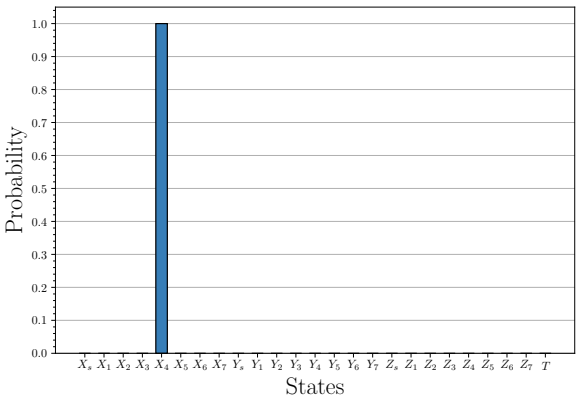


(A) Initial $b_{partial}$

(B) Updated b with `up`



(C) Updated b with `right`

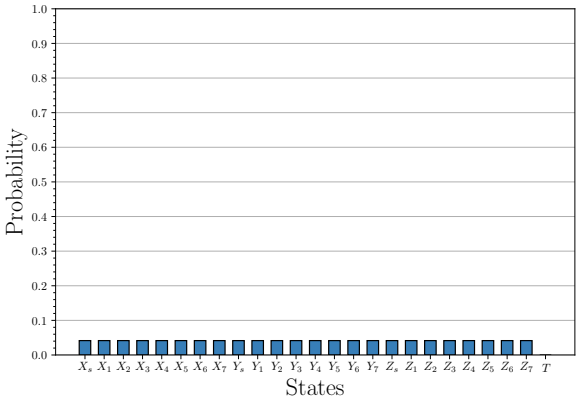


(D) Updated b with `down`

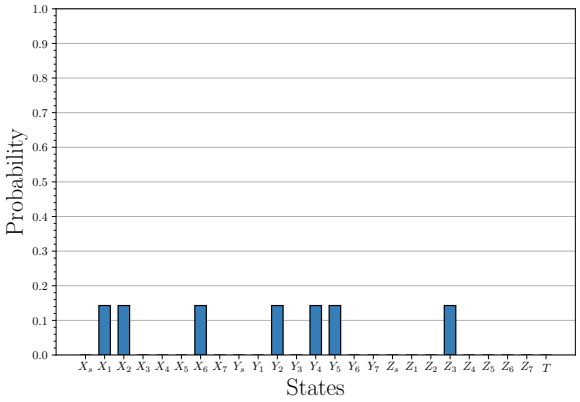
FIGURE 4.1: The initial belief $b_{partial}$ and its updated distribution after each observation

The same ATE-POMDP is now tested with the b_{full} space with the underlying true starting state again being X_1. The b_{full} has a distribution over every state (except the terminal state T) with each state having a probability of 0.0416, shown in Figure 4.2a. The first observation `up` updates the distribution from 24 possible states to 7 possible states as shown in Figure 4.2b. The second observation `right` updates the distribution to 4 possible states (Figure 4.2c) and the third observation `down` to 2 possible states (Figure 4.2d). The fourth observation `right` updates the belief b to a probability of 1 for state X_5 as shown in Figure 4.2e. This is the true underlying state of the human and thus the human's intention is task **X**.

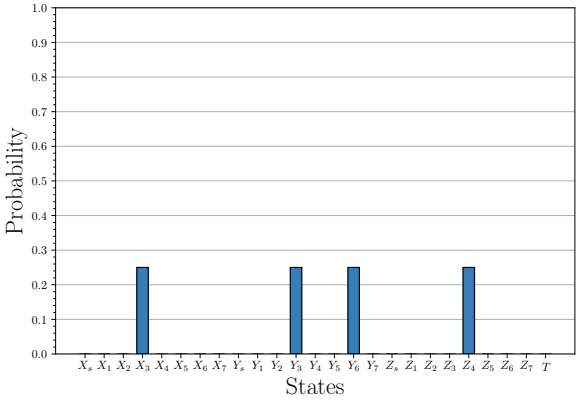
The benefit of the $b_{partial}$ compared to the b_{full} is that it makes the intention inference



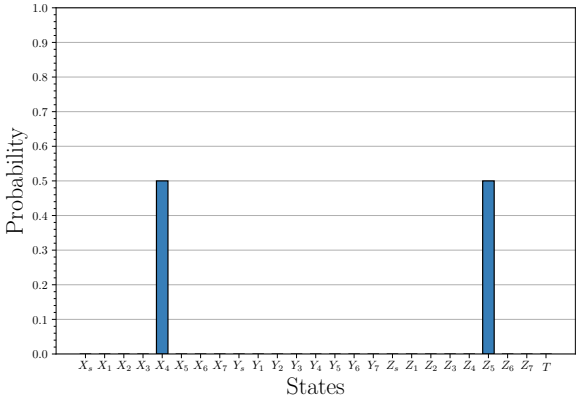
(A) Initial b_{full}



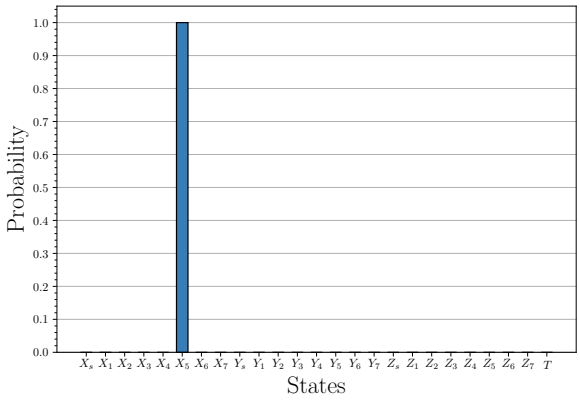
(B) Updated b with `up`



(C) Updated b with `right`



(D) Updated b with `down`



(E) Updated b with `right`

FIGURE 4.2: The initial belief b_{full} and its updated distribution after each observation

easier as there is less uncertainty initially. While it is an idealistic setting, it is still a valid way to test the ATE-POMDPs as the robot tracking the number of time steps passed is a simple way of reducing the initial uncertainty. For example, assuming x time steps have passed, the human would be executing action x-1 from a trajectory from all possible trajectories. The minus one correction factor is due to the human's *thinking* component that always takes one time step. This means that the initial belief distribution should be over all x-1 states from all n

trajectories. In a realistic setting, tracking the time steps is not always possible so the belief b_{full} should be tested as it is more appropriate to use in a real world setting.

4.1.1 Experiment Metrics

The following metrics are recorded during each experiment with each experiment being run for 20 episodes. An episode is complete once all objects are collected (all tasks are completed). The experiments are also run under two different settings - random order and arranged order. Random order means that the objects can be collected in any order in each episode while arranged order means that objects must be collected in an order defined for that particular episode. The results will show both the average ($\bar{x}$) and the standard deviation (σ) for each experiment.

Tasks Completed

This metric records the number of tasks completed by each member of a team to establish the amount of work done. This metric will show whether teammates share the workload evenly or whether one teammate does more work than the other.

Thinking Time

This metric records the number of time steps it takes each teammate to select the task they will be completing during their *thinking* component. This time is variable, taking a single time step for a human to select a task while robots will use as much time as needed according to their *thinking* components. This time is subject to increases based on the number of task collisions that may occur.

Acting time

This metric records the number of time steps it takes each teammate to execute the actions required to complete the task they selected. All robot teammates take the exact same amount of time that the human would take for completing the same task as they are using the same actions. This time is subject to increases based on the number of tasks a teammate will do in an episode.

Total Time

This metric records the total time taken to complete all the tasks in each episode for the team. Depending on the division of tasks (or lack thereof), this value can increase in the case of poor collaboration or decrease in the case of good collaboration.

Collisions

This metric records the number of task collisions in an episode for a team - the number of times the robot selects a task that the human has already chosen or has been completed. If the intention inference is good, there should be little to no collisions. Every collision that occurs adds a time step onto the current episode thereby making it longer.

4.1.2 Defining Collaborative Teams

The ATE-POMDPs are tested not only in their ability to infer intentions but also in their ability to collaborate with the human against other teams that use different techniques to infer intentions and collaborate with the human. In each team, the human uses a single time step to evaluate the completed, in-progress and uncompleted tasks, choosing an uncompleted task to complete and then executing the necessary actions in the next time step until the task is completed. In the case of one remaining task, whoever (human or robot) selects it will complete it while the other teammate will be idle until the task is completed and the episode ends.

Two teams are used to establish the upper (team with a single human) and lower bound (team of two human teammates) of performance in terms of time taken to complete all tasks. All other teams contain a single human as one teammate and a robot as the second teammate. This robot has the same *acting* component in all teams but has a different *thinking* component that will dictate the collaboration effectiveness depending on the method used to infer the human's current intention for a task. The actions performed by the human (during their *acting* component) to collect objects in the gridworlds are found using Q-learning - optimal policies found in a gridworld without a collaborator. The action trajectories based on Q-learning are used to develop the ATE-POMDP models.

If a robot makes a mistake - either during intention inference or during task selection by choosing a task that the human is currently completing or has been completed, the human will issue a warning to the robot as a collision has occurred. This behaviour is not included in the policy obtained by Q-learning but rather by the environment setting a signal that the human sees and then issues a warning. Issuing a warning takes one time step for the human before resuming their behaviour in the following time step while the robot uses the warning time step to re-think (and re-initialise its *thinking* component), resuming its behaviour in the next time step (unless another collision occurs and another warning is issued).

The human in each team functions by having a list that tracks all of the tasks that still need to be completed in the environment and excluding tasks that have been completed or are in-progress with this list being unavailable for the robot to view. The *thinking* component results in a task being selected (based on the list) before the *acting* component completes that task by executing the required actions (to move towards and collect an object in this case). Once the task is completed, the list is updated and the human continually cycles between their *thinking* and *acting* components to select new tasks or execute actions until all tasks are completed. Selecting a task to complete takes a single time step for the human while executing actions takes one time step per action.

Throughout the rest of the document, the teams are identified by the behaviour of the collaborative partner given in bold.

None

This team consists of only a single human that has to complete all tasks by themselves - they have no teammate to collaborate with. This team is used to establish

the upper bound of performance in terms of the (longest) time taken to complete an episode to compare all other teams against. Since the single human will have to complete every task by themselves, it should take the longest time to complete an episode. By acting as the upper bound, this human's time taken per episode needs to be improved on for a team to be considered as collaborating effectively.

Human

This team consists of two humans that can share the workload to complete all tasks. This team is used to establish the lower bound of performance in terms of (shortest) time taken to complete an episode to compare all other teams against. Due to the fact that both teammates are human, they can determine the completed, in-progress and uncompleted tasks since they share the same task tracking list. This team is expected to complete an episode the fastest out of all teams as it needs less time to think compared to the robots in the other teams. Other teams that approach the level of performance of this team means that they are extremely effective in collaborating. The humans are identical in implementation - that is they have the same *thinking* and *acting* components as each other and by sharing a list of all available tasks, this team avoids collisions and collaborates on the tasks in the gridworld.

Random

This team consists of a human and a robot teammate that randomly selects tasks to do in the environment. The fact that the robot randomly selects tasks to complete means that this robot can potentially be very efficient if it randomly selects the needed tasks to be completed without causing a task collision. The efficiency comes from only needing a single time step to select a task as it does not infer the human's intention, potentially saving a lot of time steps. The random selection also has the potential to cause many task collisions, increasing the number of time steps until all tasks are completed. The robot has a list of all the tasks in the environment and randomly selects one using its *thinking* component and, provided no collision occurs, the *acting* component then executes the actions needed to complete the task. This repeats until all tasks have been completed.

Distance

This team consists of a human and a robot teammate that infers the human's intention by calculating the human's distance to an object in the gridworld. When the human is a certain distance away from an object, the robot infers that is the intention of the human and then selects tasks to complete based on that information. This robot may struggle to distinguish between the human's intentions if multiple objects are located similar distances away from the human at once. This robot has a list of all the objects in the gridworld along with their x,y coordinates. The *thinking* component calculates the distance of human from an object and if that distance is less than a threshold level, then that is taken as the object the human is trying to collect (their intention) and so a task is selected based off of this information. The *acting* component then executes the actions needed to complete the task. This repeats until all tasks have been completed.

ATE-POMDP (D)

This team consists of a human and the robot that is using the ATE-POMDP (D) in its *thinking* component. This means that regardless of the order of tasks the human chooses to complete, this model has the capabilities to infer the correct intention. The agent is initialised with a belief (either $b_{partial}$ or b_{full}) and upon taking action `Wait` and receiving observations, the belief is updated such that the true underlying state (intention of the human) can be found. Once this state is known, the agent can execute the corresponding indicating action to establish the intention of the human with the possibility of this being correct or incorrect. Following this, the robot updates its knowledge of the tasks that still remain in the environment as it keeps a list of all available tasks and removing tasks that it completes itself or infers the human to be completing. After this knowledge update about the current available tasks, it selects a task from the list to complete. This is all accomplished by the *thinking* component and following task selection, the *acting* component will take over to execute the required actions to complete the selected task with one action per time step.

Once the *acting* component finishes, the *thinking* component is again initialised with the next ATE-POMDP model due to the new perspective of the human, chosen based on the indicating action in the previous ATE-POMDP model. At any point in the simulation, a warning from the human means that the model has inferred the incorrect human intention or failed to infer an intention entirely so that the robot selected a task that the human is completing themselves (or has been completed) and will be recorded as a collision. This will re-initialise the correct model if the previous intention was incorrect so that the simulation can continue.

ATE-POMDP (S)

This team consists of a human and the robot using the ATE-POMDP (S) that can only infer a set amount of combinations of tasks. The agent is initialised with a belief (either $b_{partial}$ or b_{full}) and upon taking action `Wait` and receiving observations, the belief is updated. The updated belief can be used to determine whether the human is currently completing their preferred task - with the preference order given by the reward function. If the human is completing another task, the agent will choose to assist on that first preferred task and in the case the human is completing that first preferred task, the agent will choose to assist on its second preferred task. This is all accomplished by the *thinking* component and following task selection, the *acting* component will take over to execute the required actions to complete the selected task with one action per time step.

Once the *acting* component finishes, the *thinking* component is again initialised with the next ATE-POMDP model being chosen based on the human's new current perspective. At any point in the simulation, a warning from the human means that the model has inferred the incorrect human intention or failed to infer an intention entirely so that the robot selected a task that the human is completing themselves (or has been completed) and will be recorded as a collision. This will re-initialise the correct model if the previous intention was incorrect so that the simulation can continue.

This team is only applicable to environments that do not have an arranged order of tasks - that is where no order to the collection of objects exists. Due to selecting tasks to complete based on its own preferred order, it cannot uphold the enforced arranged order without inferring the human's intention with complete certainty - an ability that this model does not have.

4.2 Experiments

This section introduces the experiments conducted to evaluate the different teams to establish the effectiveness of the intention inference and collaboration of the ATE-POMDP models compared to other teams. The most important metric that dictates effectiveness of collaboration (as a result of inference) is the total time. The best total time between the teams of **Random**, **Distance**, **ATE-POMDP (D)** and **ATE-POMDP (S)** is highlighted in green in the following tables. The table identifies each team by the collaborative partner's behaviour (in bold) and is sub divided into the human and robot. The type of belief space setting for the ATE-POMDPs is also listed in the tables.

4.2.1 Sample Gridworld

This experiment evaluates *Sample* gridworld, shown in Figure 4.3. This gridworld has a symmetrical placement as all the objects to collect are located in the corners of the grid giving them the same distance from one another and from the starting position in the middle of the grid.

This experiment serves as a coherence check to confirm whether the teams can actually infer intentions and collaborate due to the placement of objects. The overlap between tasks, that is the action trajectory used to collect an object, is minimal which should make intention inference almost trivial when it comes to distinguishing between different intentions of objects to collect. For example, a human moving `up` means that they could be either intending to collect object **A** or **B** and followed by `left`, the intention inferred should be to collect object **A**. The intentions being so easily distinguishable makes *Sample* gridworld an idealised environment.

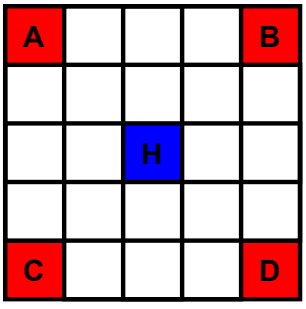


FIGURE 4.3: Symmetrical *Sample* gridworld

Random Order

Table 4.1 shows the results of the various teams where there exists no arranged order to the completion of the tasks.

None has to complete all the tasks by themselves thus taking the longest average total time from all teams with 28.4 time steps. **Human** manages to split the work

									$b_{partial}$				b_{full}			
Team / Metric		None	Human		Random		Distance		ATE-POMDP (D)		ATE-POMDP (S)		ATE-POMDP (D)		ATE-POMDP (S)	
		Human	Human	Human	Human	Robot	Human	Robot	Human	Robot	Human	Robot	Human	Robot	Human	Robot
Tasks Completed	$\bar{x}$	4	2	2	2	2	2	2	2	2	2	2	2.5	1.5	2.15	1.85
	σ	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0.36	0.36
Thinking Time	$\bar{x}$	4	2	2	2	4.7	2	4	2	7	2	6.4	2.5	7.4	2.15	6.4
	σ	0	0	0	0	2.99	0	0	0	0.45	0	0.92	0.5	0.8	0.36	0.92
Acting Time	$\bar{x}$	24.4	11.6	11.6	11.6	11.6	10.6	10.6	11.8	11.8	11.6	11.6	14.5	8.7	12.35	10.85
	σ	3.35	1.96	1.96	1.96	1.96	1.43	1.43	1.99	1.99	1.96	1.96	2.87	3.95	2.15	3.07
Total Time	$\bar{x}$	28.4	13.6		17.55		14.95		19.8		18.0		20.9		18.15	
	σ	3.34	1.96		4.77		1.63		2.23		2.57		2.39		2.54	
Collisions	$\bar{x}$	N/A	N/A		3.95		0.35		0		0		0		0	
	σ				4.64		0.48		0		0		0		0	

TABLE 4.1: Results for *Sample* gridworld with random order.

equally with both teammates completing the same number of tasks and taking 13.6 time steps on average, almost using only half the time of **None**.

Random completes the episode faster than **None** but slower than **Human**. This is due to the increase in thinking time of the robot as it selects tasks that either the human is busy completing or that have been completed, causing the human to issue a warning and requiring the robot to think about its next task. This occurs continuously until it finds a valid task to complete and accounts for the 3.95 collisions on average. Even with this number of collisions it shows that teamwork is better than working alone like **None**.

Distance works well in *Sample* gridworld due to the symmetry of all objects compared to the human. As the human moves closer to objects, this causes the robot to perceive that the closest object is the intention of the human when a set distance away and so the robot chooses a task to complete based on this information. Collisions occur as the robot does not strictly update its task knowledge based on the human's perceived task, resulting in selecting tasks that may have been completed in the past.

ATE-POMDP (D) manages to correctly infer the human's intention without causing any collisions as it waits for full certainty of the human's intention. The design of the ATE-POMDP allows objects that have been collected (or currently being collected) to be removed from the list of possible intentions, making it less likely to cause a collision. This knowledge update of in-progress or completed tasks is exactly what **Distance** lacks to prevent collisions. The extra time taken to be fully certain increases the thinking time beyond any of the other teams under both belief settings. Under the $b_{partial}$ setting, the teammates complete the same amount of work compared to the b_{full} setting where the robot teammate completes less tasks on average only completing 1.5 tasks per episode. The reason for this is that when the human acts, the robot updates its belief until it knows the human's intention with complete certainty and will execute the indicating action in the next time step. The human thinks before the robot in the next time step and selects the task that the robot would have selected resulting in the completion of more tasks.

ATE-POMDP (S) functions correctly for this environment due to tasks having no order. Due to the extra care taken in designing **ATE-POMDP (S)**, it performs better than **ATE-POMDP (D)** by using 0.6 - 1 time steps less for its acting time for the $b_{partial}$ and the b_{full} settings respectively. The effects of the $b_{partial}$ and the b_{full} settings are minimal as **ATE-POMDP (S)** only suffers a 0.15 time step increase on average per episode.

Arranged Order

Table 4.2 shows the results of the various teams where there exists an arranged order that must be upheld during the completion of the tasks.

									$b_{partial}$		b_{full}	
Team		**None**	**Human**		**Random**		**Distance**		**ATE-POMDP (D)**		**ATE-POMDP (D)**	
Metric		Human	Human	Human	Human	Robot	Human	Robot	Human	Robot	Human	Robot
Tasks Completed	$\bar{x}$	4	2	2	2	2	2	2	2	2	2.45	1.55
	σ	0	0	0	0	0	0	0	0	0	0.5	0.5
Thinking Time	$\bar{x}$	4	2	2	2	9.5	2	6.6	2	6.4	2.45	7.2
	σ	0	0	0	0	5.68	0	1.96	0	0.73	0.5	0.6
Acting Time	$\bar{x}$	23.8	11.6	11.6	11.6	11.6	11.6	11.6	11.6	11.6	14.85	9.35
	σ	3.46	1.96	1.96	1.96	1.96	1.96	1.96	1.96	1.96	2.85	4.15
Total Time	$\bar{x}$	27.8	13.6		21.1		18.2		19		21.25	
	σ	3.46	1.96		6.85		3.92		2.35		2.07	
Collisions	$\bar{x}$	N/A	N/A		7.5		0		0		0	
	σ				5.68		0		0		0	

TABLE 4.2: Results for *Sample* gridworld with arranged order

The effect of the arranged order on *Sample* gridworld is minimal with **None** and **ATE-POMDP (D)** (under the $b_{partial}$ setting) taking 0.6 and 0.8 less time steps respectively. **Random**, **Distance** and **ATE-POMDP (D)** (under the b_{full} setting) take 3.55, 3.25 and 0.25 more time steps respectively. **Human** takes the same amount of time steps in the arranged order as it did in the random order.

Distance performs the best from the other teams, being the closest to the performance of **Human**. This is attributed to the symmetrical nature of the *Sample* gridworld as it is quite distinct to know the object the human teammate is collecting since they are placed in the corners. Additionally the robot now does update its task (and intention) knowledge based on the human's perceived task and this results in no collisions unlike for the random order setting. This knowledge update is mandated by the arranged order setting to allow the collection order of objects.

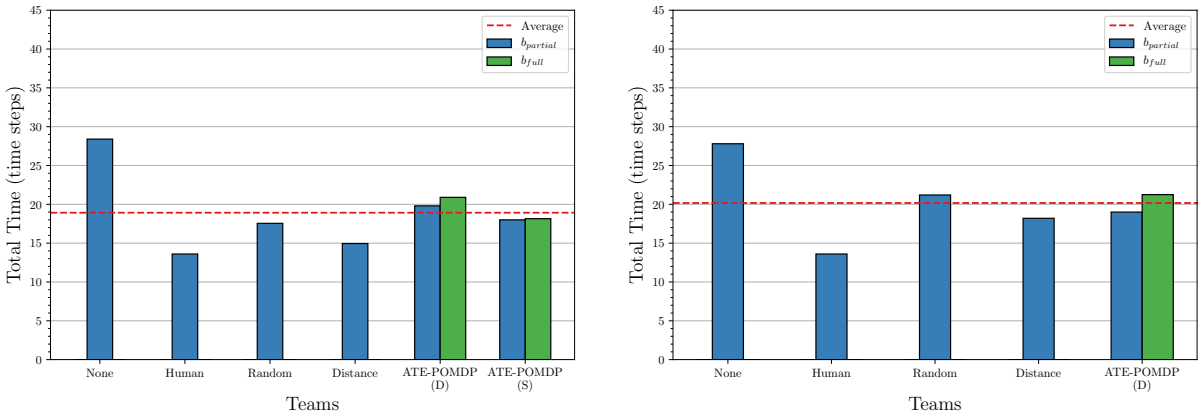
ATE-POMDP (D) performs close to the level of **Distance** under the $b_{partial}$ setting, taking 0.8 more time steps. Under the b_{full} setting, it perform worse than **Random** due to the extra time taken to reduce the uncertainty and thus the longer resulting thinking time. This longer thinking time also causes it to no longer equally split the tasks completed between both teammates in the b_{full} setting.

Analysis

A standard deviation σ of 0 for the tasks completed means that throughout all 20 episodes, the teams do exactly the amount of tasks given by the mean $\bar{x}$ each. For *Sample* gridworld, this means the teams would equally share the amount of tasks each team member completes except in the ATE-POMDP teams under the b_{full} setting. This shows that due to uncertainty, the robots take longer to infer the human's intention, leading to a longer thinking time and thus the human ends up completing more tasks. A standard deviation σ of 0 for thinking time means that the teammate used the same amount of time steps to think over all 20 episodes.

The average total times per episode for the different teams in *Sample* gridworld for the random and arranged order are shown in Figure 4.4. The random order takes an average of 18.91 time steps to complete an episode while the arranged order takes an average of 20.18 to complete an episode. The average total times are given by the

red lines.



(A) Random order

(B) Arranged order

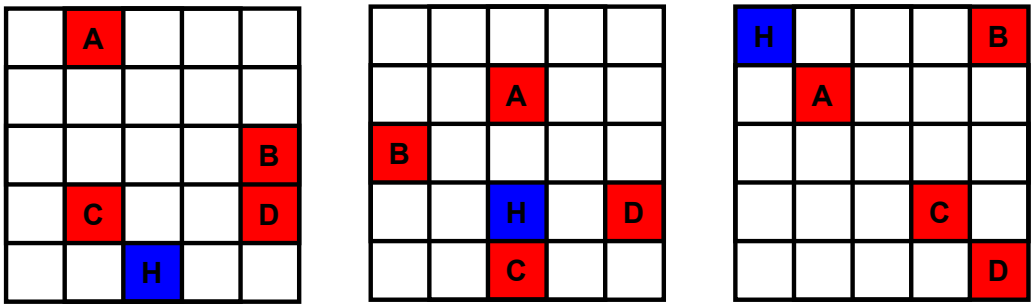
FIGURE 4.4: *Sample* gridworld total average time steps

Based on the results of the *Sample* gridworld, the best team for both the random and arranged order is **Distance** as this team is the closest to the **Human** performance. This shows that when objects are distinguishable due to being placed in equal distances from one another and the human, the distance metric is sufficient to infer the human's intention and enable collaboration. This raises a question whether the success of this team is due to the symmetrical nature of *Sample* gridworld. However, based on the standard deviations σ for the total time, **Distance** is not significantly better than either of the ATE-POMDP teams.

The average total times confirm that the addition of a teammate in the environment benefits the completion of the tasks through collaboration. Due to the objects being symmetrically located fixed distances from one another and the human, all robots' *thinking* components have enough time to infer the human's intention with minimal difficulty as they are quite distinct in the *Sample* gridworld. Additionally, due to the symmetrical nature of *Sample* gridworld itself, the acting time for all teammates is the same as each team completes the same amount of work except under the b_{full} setting. While this environment does have objects spaced out equally, other environments may not, and so updating the knowledge based on distance can lead to incorrect inference (and collisions) in cases whereby the human is simply moving past an object to get to another or objects are located closely to one another.

4.2.2 Alternative Gridworlds

The results of the *Sample* gridworld experiment show that objects being evenly spaced from the human and other objects allows for sufficient time for the robots to infer the human's intention using the different *thinking* components. To determine the effect of different placements of objects, three more gridworlds of 5×5 size are generated to be tested. The objects in the gridworlds were placed arbitrarily to determine the effects of asymmetrical placement on the different teams. The gridworlds are referred to as $5\times5A$, $5\times5B$ and $5\times5C$ and are shown in Figure 4.5.



(A) 5×5A (B) 5×5B (C) 5×5C

FIGURE 4.5: Three alternative 5×5 gridworlds

Random Order

Tables 4.3 - 4.5 shows the results of the three 5×5 gridworlds with the random order for collection of objects.

									$b_{partial}$				b_{full}			
Team / Metric		None	Human		Random		Distance		ATE-POMDP (D)		ATE-POMDP (S)		ATE-POMDP (D)		ATE-POMDP (S)	
		Human	Human	Human	Human	Robot	Human	Robot	Human	Robot	Human	Robot	Human	Robot	Human	Robot
Tasks Completed	$\bar{x}$	4	2	2	2	2	2.25	1.75	2.6	1.40	2.15	1.85	2.65	1.35	2.25	1.75
	σ	0	0	0	0	0	0.43	0.43	0.49	0.5	0.36	0.36	0.43	0.49	0.43	0.43
Thinking Time	$\bar{x}$	4	2	2	2	3.7	2.25	5.75	2.6	7.05	2.15	6.9	2.75	7.6	2.25	7.25
	σ	0	0	0	0	2.39	0.43	2.57	0.49	2.22	0.36	1.48	0.43	2.11	0.43	2.02
Acting Time	$\bar{x}$	18.65	9.25	8.8	9.85	9.2	9.7	8.1	12	6.85	10.7	8.2	12.65	6	11.3	7.5
	σ	2.5	2.28	2.27	1.85	2.38	2.19	2.4	1.37	2.33	1.52	2.46	1.96	2.35	2.43	2.81
Total Time	$\bar{x}$	22.65	12.4		16.1		15.1		17.3		16.3		17.7		16.55	
	σ	2.5	2.18		4.33		3.25		1.79		1.42		1.79		1.8	
Collisions	$\bar{x}$	N/A	N/A		2.9		0.7		0		0		0.3		0.3	
	σ				3.92		0.95		0		0		0.9		0.9	

TABLE 4.3: Results for 5×5A gridworld with random order

									$b_{partial}$				b_{full}			
Team / Metric		None	Human		Random		Distance		ATE-POMDP (D)		ATE-POMDP (S)		ATE-POMDP (D)		ATE-POMDP (S)	
		Human	Human	Human	Human	Robot	Human	Robot	Human	Robot	Human	Robot	Human	Robot	Human	Robot
Tasks Completed	$\bar{x}$	4	2	2	2	2	2	2	2.15	1.85	2.15	1.85	2.3	1.7	2.15	1.85
	σ	0	0	0	0	0	0	0	0.36	0.36	0.36	0.36	0.46	0.46	0.36	0.36
Thinking Time	$\bar{x}$	4	2	2	2	5.85	2	3.3	2.15	6	2.15	5.45	2.3	5.75	2.15	5.65
	σ	0	0	0	0	5.51	0	1	0.36	0.84	0.36	0.74	0.46	0.99	0.36	0.85
Acting Time	$\bar{x}$	16.95	7.6	7.45	7.55	7.45	7.75	8.2	8.55	6.65	8.3	6.1	8.55	6.25	8.3	6.1
	σ	1.16	1.11	0.86	1.16	1.2	1.3	1.21	1.4	1.42	1.19	1.48	2.18	2.09	1.19	1.48
Total Time	$\bar{x}$	20.95	10.15		15.45		11.85		14.45		12.45		14.65		12.65	
	σ	1.16	0.91		7.29		1.28		0.92		0.5		1.15		0.65	
Collisions	$\bar{x}$	N/A	N/A		5.15		0.7		0		0		0		0	
	σ				7.33		0.71		0		0		0		0	

TABLE 4.4: Results for 5×5B gridworld with random order

									$b_{partial}$				b_{full}			
Team / Metric		None	Human		Random		Distance		ATE-POMDP (D)		ATE-POMDP (S)		ATE-POMDP (D)		ATE-POMDP (S)	
		Human	Human	Human	Human	Robot	Human	Robot	Human	Robot	Human	Robot	Human	Robot	Human	Robot
Tasks Completed	$\bar{x}$	4	2.15	1.85	2.1	1.9	2.45	1.55	2.55	1.45	2.6	1.4	2.5	1.5	2.75	1.25
	σ	0	0.36	0.36	0.3	0.3	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.43	0.43
Thinking Time	$\bar{x}$	4	2.15	1.85	2.1	3.34	2.45	3.6	2.55	7.55	2.6	4.95	2.5	9.6	2.75	5.6
	σ	0	0.36	0.36	0.3	3.4	0.5	1.2	0.5	2.42	0.5	2.38	0.5	1.96	0.43	3.04
Acting Time	$\bar{x}$	20.8	11.05	9.95	11.4	10.6	13.75	10.15	14.1	8.55	14.2	7.95	15.6	7.6	15.25	6.25
	σ	2.79	1.86	1.88	2.35	2.18	2	2	2.41	1.86	2.52	1.8	2.4	2.6	2.23	3.43
Total Time	$\bar{x}$	24.8	13.9		16.65		17.35		19.6		18		21.15		19	
	σ	2.79	1.84		4.71		2.03		1.83		1.55		1.68		2.05	
Collisions	$\bar{x}$	N/A	N/A		2.05		0.4		0		0		0		0	
	σ				3.41		0.58		0		0		0		0	

TABLE 4.5: Results for 5×5C gridworld with random order

As established prior, **None** takes the most time steps while **Human** takes the least time steps in all three 5×5 gridworlds.

Random manages to split the amount of tasks evenly in the $5\times5A$ and $5\times5B$ gridworlds while also having the most number of collisions compared to all teams. It performs the best in the $5\times5C$ gridworld as this configuration is difficult for a distance based method as objects **C** and **D** are located close together. Even though $5\times5A$ gridworld has a similar placement with **B** and **D** located close together, the stochastic nature of selecting the objects may have played a role in having it perform worse than **Distance**.

Distance completes the $5\times5A$ and $5\times5B$ gridworlds the fastest compared to the other teams (except **Human**) as these objects have placements that are distinguishable from one another. Particularly in the $5\times5B$ gridworld, moving in any cardinal direction would immediately give away the intention of the human. This is indicated by the 11.85 total time steps, only 1.7 time steps more than **Human**. Due to the human starting close to object **C**, the robot may perceive the human's intention to be that object even though the human may move to another. This results in a small number of collisions.

ATE-POMDP (D) generally tends to complete episodes with an average time step total more than any other team (except **None**) even with 0 task collisions in the $b_{partial}$ setting. This is due to its longer thinking time compared to the other teams. The effectiveness of this team is shown to decrease based on the increased uncertainty under the b_{full} setting.

ATE-POMDP (S) performs better than **ATE-POMDP (D)** in all grids (under both belief settings). The team generally performs slightly worse than **Distance** and **Random** with no collisions except on the $5\times5A$ gridworld under the b_{full} setting, particularly due to the objects **B** and **D** being located close together.

Arranged Order

Tables 4.6 - 4.8 shows the results of the various teams where there exists an arranged order that must be upheld during the completion of the tasks.

									$b_{partial}$		b_{full}	
Team		**None**	**Human**		**Random**		**Distance**		**ATE-POMDP (D)**		**ATE-POMDP (D)**	
Metric		Human	Human	Human	Human	Robot	Human	Robot	Human	Robot	Human	Robot
Tasks	$\bar{x}$	4	2	2	2	2	2.55	1.45	2.55	1.45	2.65	1.35
Completed	σ	0	0	0	0	0	0.5	0.5	0.5	0.5	0.48	0.48
Thinking	$\bar{x}$	4	2	2	2	7.75	2.55	6.95	2.55	6.85	2.65	6.9
Time	σ	0	0	0	0	5.1	0.5	2.73	0.5	1.71	0.48	1.64
Acting	$\bar{x}$	18.75	9.1	9.2	9.1	9.2	11.55	6.55	11.55	6.9	12.1	6.35
Time	σ	2.47	2.17	2.2	2.17	2.2	1.83	2.06	1.86	2.49	1.61	2.2
Total	$\bar{x}$	22.75	12.55		18.3		16.45		17		17.1	
Time	σ	2.47	1.99		5.21		2.82		1.79		1.54	
Collisions	$\bar{x}$	N/A	N/A		5.75		1.3		0		0	
	σ				5.09		0.9		0		0	

TABLE 4.6: Results for $5\times5A$ gridworld with arranged order

None and **Human** behave as expected, taking the most and least amount of time steps to complete an episode respectively.

Random performs the worst out of all the teams due to the arranged order to collect objects. It also has the highest amount of collisions indicating it is ill-suited when an

Metric \ Team		None	Human		Random		Distance		$b_{partial}$ ATE-POMDP (D)		b_{full} ATE-POMDP (D)	
		Human	Human	Human	Human	Robot	Human	Robot	Human	Robot	Human	Robot
Tasks Completed	$\bar{x}$	4	2	2	2	2	2	2	2.1	1.9	2.25	1.75
	σ	0	0	0	0	0	0	0	0.3	0.3	0.43	0.43
Thinking Time	$\bar{x}$	4	2	2	2	8.75	2	6.95	2.1	5.85	2.25	6.2
	σ	0	0	0	0	5.4	0	2.56	0.3	0.91	0.43	1.12
Acting Time	$\bar{x}$	16.85	7.8	7.7	7.8	7.7	7.85	7.75	8.3	7.15	8.95	6.35
	σ	1.01	1.21	1.27	1.21	1.27	1.19	1.22	1.42	1.42	1.66	1.9
Total Time	$\bar{x}$	20.85	10.65		17.4		14.7		14.55		14.9	
	σ	1.01	0.96		5.48		1.9		1.07		1.18	
Collisions	$\bar{x}$	N/A	N/A		6.75		2.1		0		0	
	σ				5.35		1.04		0		0	

TABLE 4.7: Results for *5×5B* gridworld with arranged order

Metric \ Team		None	Human		Random		Distance		$b_{partial}$ ATE-POMDP (D)		b_{full} ATE-POMDP (D)	
		Human	Human	Human	Human	Robot	Human	Robot	Human	Robot	Human	Robot
Tasks Completed	$\bar{x}$	4	2.1	1.9	2.1	1.9	2.85	1.15	2.7	1.3	2.75	1.25
	σ	0	0.3	0.3	0.3	0.3	0.36	0.36	0.46	0.46	0.43	0.43
Thinking Time	$\bar{x}$	4	2.1	1.9	2.1	9.45	2.85	8.65	2.7	7.15	2.75	8.2
	σ	0	0.3	0.3	0.3	5.18	0.36	5.5	0.46	1.98	0.43	1.86
Acting Time	$\bar{x}$	20.8	11.3	11	11.3	11	15.05	7.15	14.2	7.9	14.55	7.65
	σ	3.06	2	2.21	2	2.21	3.19	2.55	2.66	1.97	2.54	2
Total Time	$\bar{x}$	24.8	14.5		22.05		20.25		19		19.6	
	σ	3.06	2.09		5.04		4.12		2.21		2.06	
Collisions	$\bar{x}$	N/A	N/A		7.55		1.9		0		0	
	σ				5.57		1.64		0		0	

TABLE 4.8: Results for *5×5C* gridworld with arranged order

order needs to be upheld as it does not attempt to infer the human's intention at all.

Distance performs the best only in the *5×5A* gridworld although it does have a small number of collisions. These collisions are a result of the close proximity of objects to collect as the robot struggles to differentiate between them or when the human is simply walking past an object with no intention of collecting it.

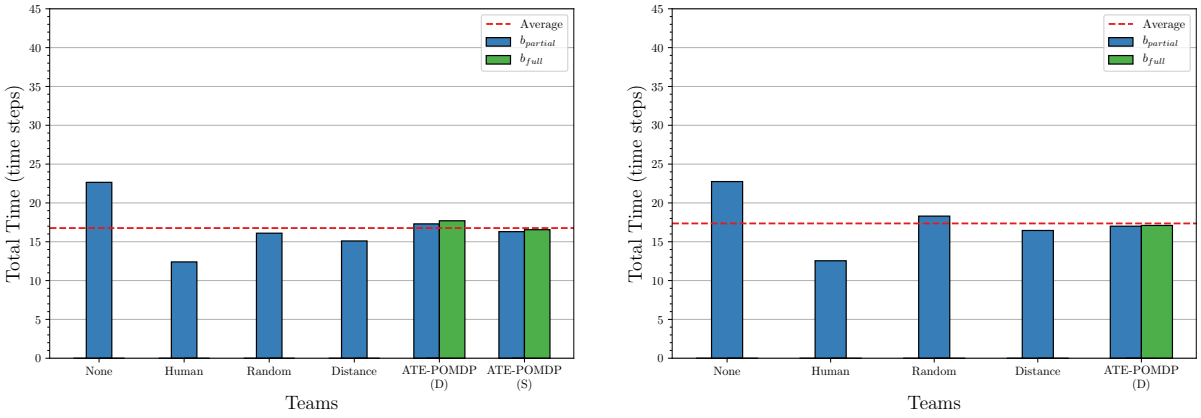
ATE-POMDP (D) performs the best in the *5×5B* and *5×5C* gridworld under the $b_{partial}$ setting. The b_{full} setting increases the total time by a small amount (0.2 time steps in *5×5B* and 0.6 time steps in *5×5C*) making it take slightly longer than **Distance** in the *5×5B* gridworld.

Analysis

The three asymmetrical gridworlds show that all teams find it difficult to infer the intention of the human and collaborate, failing to either split tasks evenly or having an increase in the number of collisions. The ATE-POMDP teams need more thinking time to infer intentions, causing them to not split the tasks between the human and the robot evenly in all three gridworlds under both the random and arranged order as given by the standard deviation σ greater than 0. Both **ATE-POMDP (D)** and **ATE-POMDP (S)** have 0 collisions for all gridworlds in the random and arranged order under both belief settings apart from a collision in *5×5A* under the b_{full} setting for random order - the smallest compared to the other teams. **None**, **Human** and **Random** still manage to evenly split the amount of tasks among their team members in the *5×5A* and *5×5B* although **Random** does have the highest number of collisions across all gridworlds in the both the random and arranged order. **Distance** acts as a middle ground in that it splits the tasks between the human and

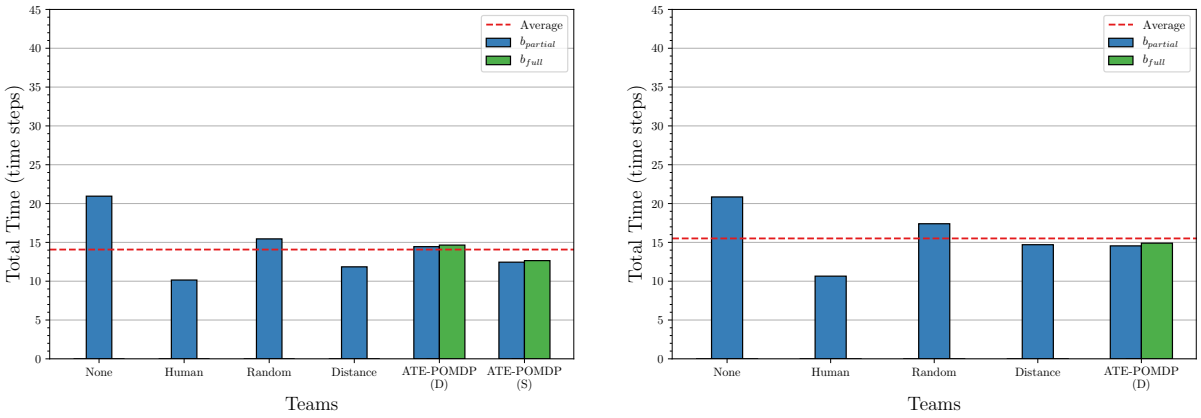
robot better than both ATE-POMDP teams at a higher collision cost albeit lower than **Random**. This is only for the random order as **ATE-POMDP (D)** does perform generally better than **Distance** in the arranged order due to no collisions.

The average total times per episode for the different teams in the three 5×5 gridworlds for the random and arranged order are shown in Figures 4.6 - 4.8. The random order takes an average of 16.75, 14.08 and 18.81 time steps to complete an episode for $5\times5A$, $5\times5B$ and $5\times5C$ respectively. The arranged order takes an average of 17.36, 15.51 and 20.03 time steps to complete an episode for $5\times5A$, $5\times5B$ and $5\times5C$ respectively. The average total times are given by the red lines.



(A) Random order (B) Arranged order

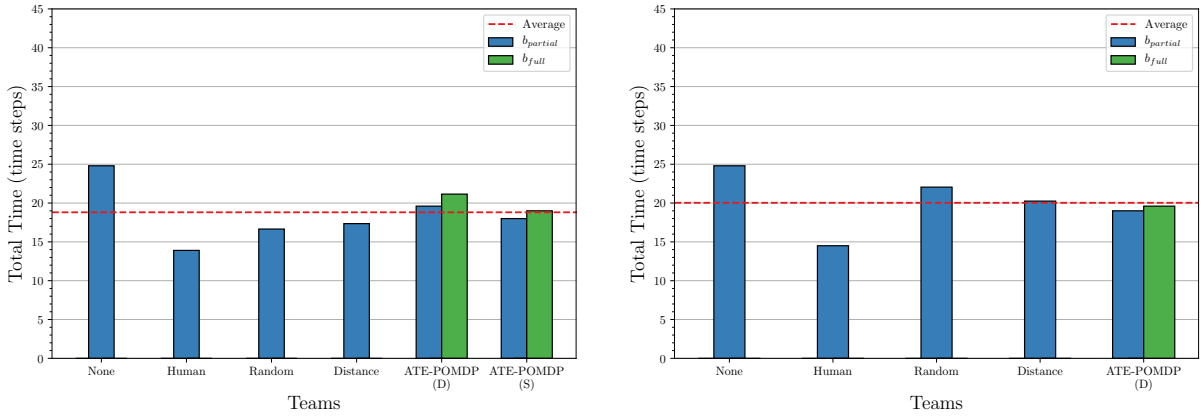
FIGURE 4.6: The average total times for the $5\times5A$ gridworld



(A) Random order (B) Arranged order

FIGURE 4.7: The average total times for the $5\times5B$ gridworld

For **ATE-POMDP (D)**, it is noted that as the level of uncertainty is higher (when under the b_{full} setting), more time is required to infer the intention, increasing the thinking time and thus reducing the robot's ability to effectively collaborate by completing as many tasks as the human. The effects of the different belief distributions have a bigger effect on the **ATE-POMDP (D)** than they do on **ATE-POMDP (S)**. This is due to the fact that the more time spent on designing the **ATE-POMDP (S)** allows it to have similar trajectories grouped with the same assisting action that selects a task without requiring to be fully certain about the human's intention. **ATE-POMDP (D)** uses less total time than **Distance** in the arranged order setting of



(A) Random order (B) Arranged order

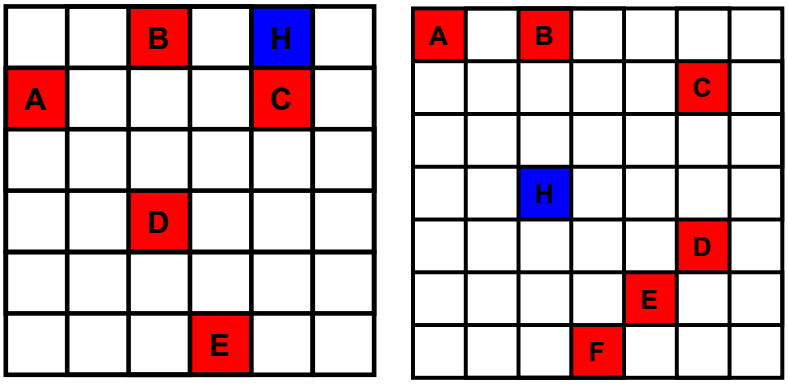
FIGURE 4.8: The average total times for the $5\times5C$ gridworld

the three gridworlds.

As the objects to collect are no longer spaced equally around the grid, some task intentions now have less time than others to be inferred meaning that the robots can potentially run out of time to infer the current task before the human completes it and moves onto another. This would cause the robot to need to spend even longer inferring the new human intention, increasing the thinking time and leading to poor collaboration overall as it would not be able to decide on a task to complete itself and therefore share the workload. These tasks are however more realistic as tasks generally are quite different from one another, varying in length and number of actions required to complete them. This means that inference should be as fast as possible to be able to detect these variable length task intentions in the shortest time possible to enable collaboration.

4.2.3 Larger Gridworlds

The size of gridworlds was restricted to 5×5 and four objects for the previous two experiments. This experiment uses gridworlds of sizes 6×6 and 7×7 with five and six objects respectively. The gridworlds are shown in Figure 4.9.



(A) 6×6 (B) 7×7

FIGURE 4.9: Gridworlds with increased size and object number

Random Order

Tables 4.9 and 4.10 show the results of the 6×6 and 7×7 gridworlds respectively with the random order for collection of objects.

									$b_{partial}$				b_{full}			
Metric \ Team		None	Human		Random		Distance		ATE-POMDP (D)		ATE-POMDP (S)		ATE-POMDP (D)		ATE-POMDP (S)	
		Human	Human	Human	Human	Robot	Human	Robot	Human	Robot	Human	Robot	Human	Robot	Human	Robot
Tasks	$\bar{x}$	5	2.5	2.5	2.55	2.45	3	2	3.1	1.9	3.1	1.9	3.35	1.65	3.45	1.55
Completed	σ	0	0.5	0.5	0.5	0.5	0.67	0.4	0.3	0.3	0.3	0	0.48	0.48	0.5	0.5
Thinking	$\bar{x}$	5	2.5	2.5	2.55	4.4	3.05	3.9	3.1	10.9	3.1	5.45	3.35	11.25	3.45	7.45
Time	σ	0	0.5	0.5	0.5	2.78	0.67	0.99	0.3	2.75	0.3	0.86	0.48	2.32	0.5	0.8
Acting	$\bar{x}$	24.35	12.45	12.55	13.15	12.95	14.55	12.65	16.1	8.7	15.3	12.95	17.2	7.7	16.0	9.75
Time	σ	2.26	1.83	2.01	1.68	1.82	2.22	1.31	2.34	2.49	1.52	1.43	1.83	2.23	2.47	3.59
Total	$\bar{x}$	29.35	16.45		19.9		18.9		22.15		19.5		22.8		21.0	
Time	σ	2.26	1.77		3.39		1.97		2.22		0.97		1.75		1.1	
Collisions	$\bar{x}$	N/A	N/A		2.95		0.5		0.55		0		1.4		0	
	σ				3.15		0.59		1.53		0		2.71		0	

TABLE 4.9: Results for 6×6 gridworld with random order

									$b_{partial}$				b_{full}			
Metric \ Team		None	Human		Random		Distance		ATE-POMDP (D)		ATE-POMDP (S)		ATE-POMDP (D)		ATE-POMDP (S)	
		Human	Human	Human	Human	Robot	Human	Robot	Human	Robot	Human	Robot	Human	Robot	Human	Robot
Tasks	$\bar{x}$	6	3	3	2.95	3.05	3.6	2.4	3.45	2.55	3.35	2.55	3.95	2.2	3.55	2.2
Completed	σ	0	0.32	0.32	0.22	0.22	0.59	0.62	0.59	0.49	0.48	0.67	0.38	0.4	0.5	0.8
Thinking	$\bar{x}$	6	3	3	2.95	7.1	3.95	9.2	3.45	9.9	3.35	7.45	3.95	12.6	3.55	7.65
Time	σ	0	0.32	0.32	0.22	3.96	0.59	2.86	0.59	3.08	0.48	1.36	0.38	4.31	0.5	1.62
Acting	$\bar{x}$	36.55	18.45	18.25	18.4	18.8	25.55	18.35	22.1	15.45	20.65	16.55	23.3	13.15	21.6	15.15
Time	σ	5.25	2.36	3.06	2.78	2.46	4.7	3.61	3.62	3.53	1.93	3.89	3.72	2.35	2.08	4.19
Total	$\bar{x}$	42.55	22.75		28.75		31.55		29.4		26.55		29.9		27.1	
Time	σ	5.25	2.77		5.9		4.81		3.18		2.25		3.43		2.12	
Collisions	$\bar{x}$	N/A	N/A		6.85		0.6		0.3		0		0.9		0.1	
	σ				6.45		0.58		0.9		0		1.37		0.44	

TABLE 4.10: Results for 7×7 gridworld with random order

None and **Human** remain the upper and lower bound in terms of average time taken to complete an episode.

Random starts to feel the effects of the increased number of objects as it now has a bigger selection of objects. This is seen as compared to the 5×5 gridworlds where it generally beat the ATE-POMDP teams for the random order, it is now being outperformed by the **ATE-POMDP (S)** under the $b_{partial}$ setting for both the 6×6 and 7×7 gridworld. This means that as the number of objects increases, the number of collisions increases and the effectiveness of **Random** decreases.

Distance continues to generally perform better than the other teams although **ATE-POMDP (S)** does outperform it in the 7×7 gridworld. This can be attributed to objects **D**, **E** and **F** being placed close to one another, making it difficult for the robot to determine the true intention of the human when they approach one of these objects in this gridworld.

ATE-POMDP (D) generally performs the worst out of all teams due to its thinking time, made worse under the b_{full} setting. However, it is noted that the effects of the $b_{partial}$ setting to the b_{full} setting are minimal, increasing the total time by 0.65 and 0.5 for the 6×6 and 7×7 gridworld respectively.

ATE-POMDP (S) performs better than **Random** under the $b_{partial}$ setting. This shows that as the number of objects increases, the total time taken by this team begins to beat **Random** with less collisions. The collisions occur in the b_{full} setting due to the added uncertainty.

Arranged Order

Tables 4.11 - 4.12 shows the results of the various teams where there exists an arranged order that must be upheld during the completion of the tasks.

									$b_{partial}$		b_{full}	
Team		**None**	**Human**		**Random**		**Distance**		**ATE-POMDP (D)**		**ATE-POMDP (D)**	
Metric		Human	Human	Human	Human	Robot	Human	Robot	Human	Robot	Human	Robot
Tasks Completed	$\bar{x}$	5	2.45	2.55	2.45	2.55	3	2	3	2	3.25	1.75
	σ	0	0.5	0.5	0.5	0.5	0	0	0	0	0.43	0.43
Thinking Time	$\bar{x}$	5	2.45	2.55	2.45	14.2	3	12.75	3	9.6	3.25	11.65
	σ	0	0.5	0.5	0.5	8.36	0	3.66	0	2.69	0.43	2.57
Acting Time	$\bar{x}$	26.15	12.95	12.5	12.95	12.5	15.35	9.3	15.4	9.25	16.95	7.45
	σ	2.31	1.8	2.11	1.8	2.11	2.48	2.49	2.48	2.55	1.6	2.75
Total Time	$\bar{x}$	31.15	16.7		28.35		24.05		21.3		22.9	
	σ	2.31	1.65		8.47		3.15		1.85		1.81	
Collisions	$\bar{x}$	N/A	N/A		11.65		4.35		0		0.3	
	σ				8.3		1.46		0		0.9	

TABLE 4.11: Results for 6×6 gridworld with arranged order

									$b_{partial}$		b_{full}	
Team		**None**	**Human**		**Random**		**Distance**		**ATE-POMDP (D)**		**ATE-POMDP (D)**	
Metric		Human	Human	Human	Human	Robot	Human	Robot	Human	Robot	Human	Robot
Tasks Completed	$\bar{x}$	6	3	3	3	3	4.2	1.8	3.45	2.55	3.85	2.15
	σ	0	0.32	0.32	0.32	0.32	0.4	0.4	0.5	0.5	0.45	0.49
Thinking Time	$\bar{x}$	6	3	3	3	15.35	4.2	13.65	3.45	9.05	3.7	11.4
	σ	0	0.32	0.32	0.32	6.82	0.4	5.39	0.5	1.75	0.45	3.45
Acting Time	$\bar{x}$	36.65	18.85	18.85	18.85	18.85	26.15	11.2	20.8	15.85	21.9	12.55
	σ	4.87	2.63	3.11	2.63	3.12	2.82	3.41	3.54	3.32	3.22	3.38
Total Time	$\bar{x}$	42.65	23.4		35.75		32.3		28.1		28.4	
	σ	4.87	2.71		7.75		3.46		3.34		3.95	
Collisions	$\bar{x}$	N/A	N/A		12.35		1.95		0		0.25	
	σ				6.73		1.88		0		0.71	

TABLE 4.12: Results for 7×7 gridworld with arranged order

None and **Human** remain the upper and lower bound in terms of average time taken to complete an episode.

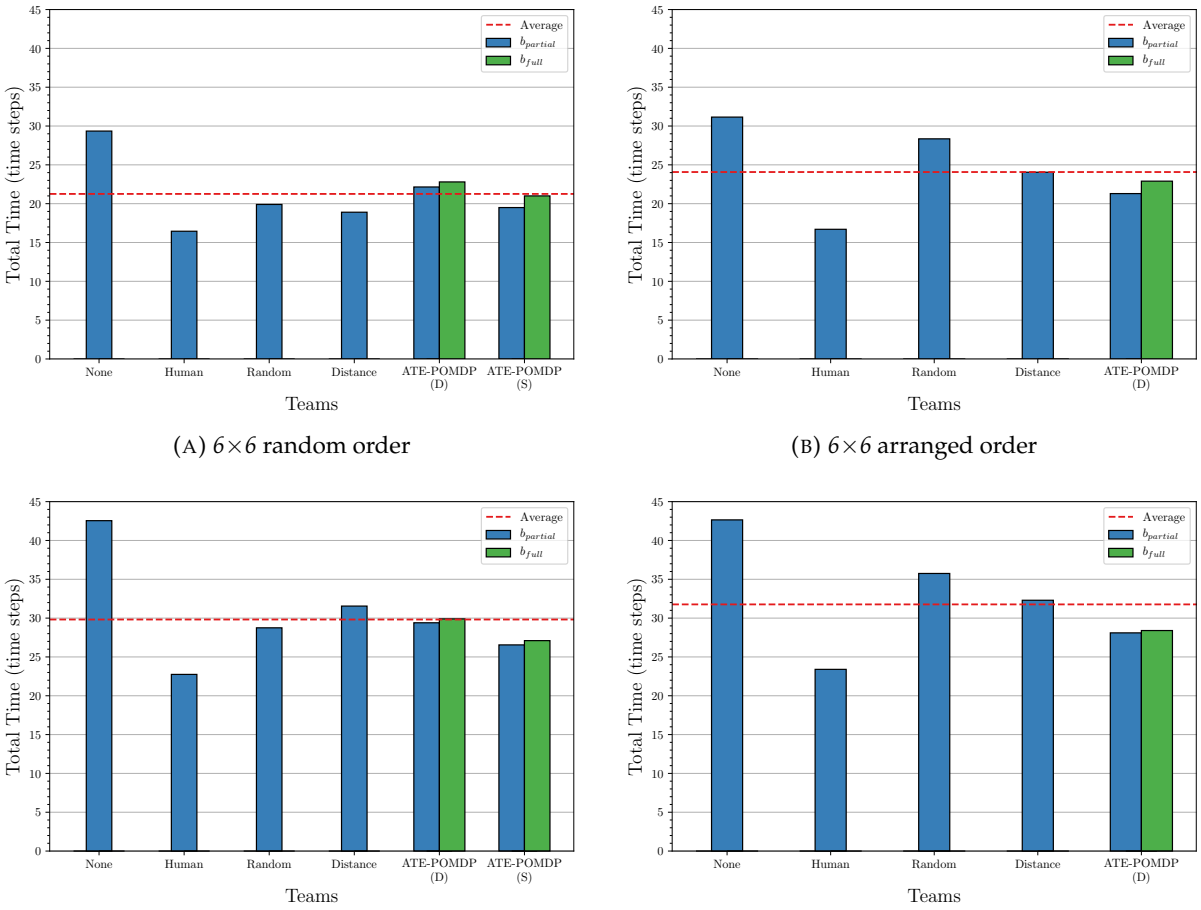
Random performs worse than every other team in each gridworld. This is attributed to the fact that there is an arranged order to the collection of objects that needs to be enforced. The increased number of objects leads to more collisions as they are randomly selected until eventually finding the correct task that adheres to the order, leading to a large thinking time.

Distance performs better than **Random** but worse than **ATE-POMDP (D)** under both belief settings. The arranged order causes more collisions as the human is moving past objects, and unlike the random order where even if the inference was wrong, the correct object to collect could be selected, this case does not happen for the arranged order.

ATE-POMDP (D) performs the best in both the 6×6 and 7×7 gridworlds compared to all teams under both the $b_{partial}$ and b_{full} settings. The team performs well due to the extra thinking time taken to infer with complete certainty the intention of the human allowing the corresponding task that follows that order to be found. The b_{full} setting introduces collisions, although these are still less than the other teams. The collisions come from the fact that the human acts first and selects the task before the robot can due to the extra thinking time - something that didn't happen in the $b_{partial}$ setting.

Analysis

The average total times per episode for the different teams for the 6×6 and 7×7 gridworlds for the random and arranged order are shown in Figure 4.10. The random order takes an average of 21.26 and 29.82 time steps to complete an episode while the arranged order takes an average of 24.08 and 31.77 to complete an episode for the 6×6 and 7×7 gridworlds respectively. The average total times are given by the red lines.



(A) 6×6 random order

(B) 6×6 arranged order

(C) 7×7 random order

(D) 7×7 arranged order

FIGURE 4.10: The average total times for the 6×6 and 7×7 gridworlds

The increased number of objects in the 6×6 and 7×7 gridworlds affect **Random** the most as it now has a larger selection of objects to choose from. This leads to the increased number of collisions. The effects of scaling the grids themselves affect the ATE-POMDP models themselves as now there are more actions the human has to do, leading to more states in the model and thus making the mode larger and harder to solve. **Random** should be avoided if there is an arranged order that exists between tasks, as while it does help complete the episode on average faster than **None**, it is not a significant time decrease. As the number of tasks increase, **Random** can begin to perform worse than **None** making it ill-suited for collaboration.

Collisions in the ATE-POMDP models come from the fact that the simulation runs sequentially with the human teammate always acting first, having completed their

task and selecting their next task that robot would have chosen after inferring the human's previous intention in the same time step. Collisions increase in the b_{full} setting as the robot spends more time thinking and so infers intentions slower.

4.3 Discussion

The results show that while the ATE-POMDP models can provide performance close to or exceeding the teams of **None**, **Random** and **Distance** in the simple gridworld environments with random order between tasks, none of the teams approach the total time taken by **Human**. However, taking into account the collision metric that has the most influence on the total time taken to complete an episode, a clear winning team can be found in terms of using the shortest amount of time to assist in completing all tasks in the gridworlds. The time cost of a collision was set to 1 time step for the purpose of simulation. This collision time cost is difficult to quantify as tasks can have variable time costs depending on the actions that are executed as they can hinder instead of assist collaboration. By setting the time cost collision to 1, the robot was assumed to unrealistically receive immediate feedback from the human in the form of a warning when a collision occurred as the human would always need to be observing the robot to be able to issue warnings so quickly if it was acting incorrectly.

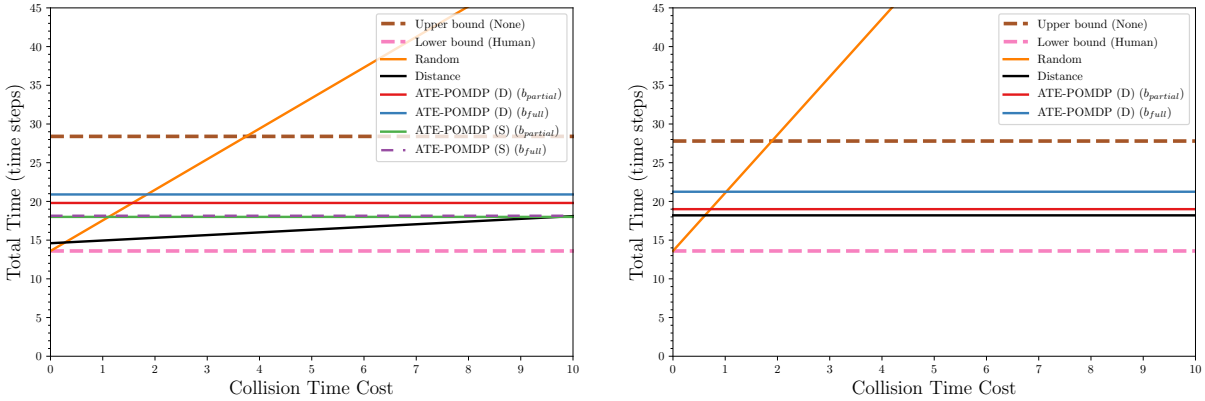
Consider the case in the real world whereby the human has completed the coffee making task, set down their mug on the kitchen counter and moved away to complete a different task. The robot incorrectly selects the coffee making task as its next task and moves to the mug so that it can be picked and moved to the sink to wash so that it can be used in the coffee making task (that is already complete). The human teammate, seeing this, issues a warning and so the robot would simply put it down and attempt to select a new task to complete. This is an example of a minor time cost. The robot could have potentially moved towards the sink and only then received a warning from the human so it would need to move back to the kitchen counter and set the coffee down. This could be considered a minor to moderate time cost. In the case that the robot moved to the sink and threw out the coffee before washing the mug to use it in the coffee making task, a warning would be too late to prevent the robot from having created a mistake that now takes multiple time steps to rectify - the coffee making task must now be re-completed.

Based on the above example, the value of the time step cost changes the effectiveness of teams in the different gridworlds. The total time is calculated as:

$$total\ time\ = (completion\ time) + (c * collisions) \tag{4.1}$$

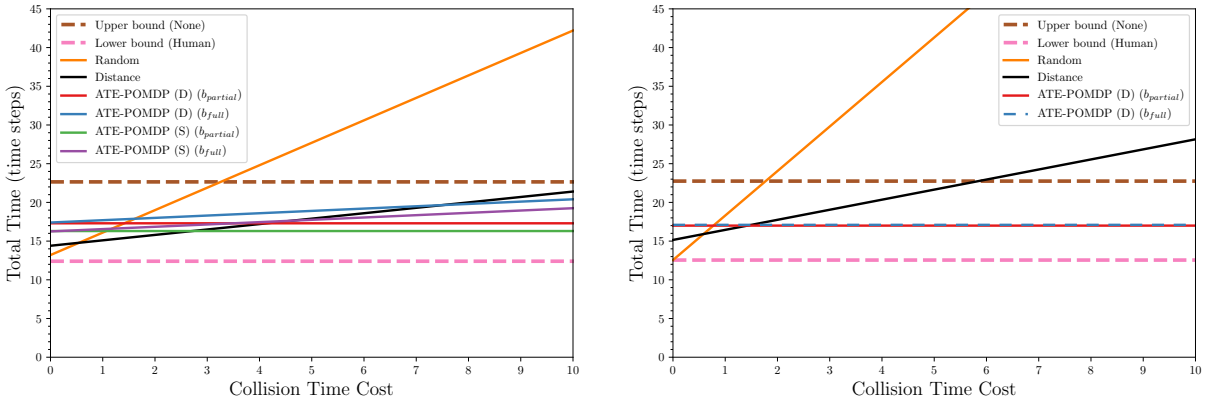
Where c is a constant chosen to represent the time cost of a collision. Equation (4.1) is a function in c and varying the value of c shows that the average total time per episode increases based on the number of collisions as more collisions mean greater time costs. Figures 4.11 - 4.16 show the increasing time per episode when the value of c is varied from 0 - 10 for each gridworld experiment.

In Figures 4.11 - 4.16, the *lower* and *upper* bound are given by **Human** and **None** respectively in terms of shortest and longest times to complete an episode. Any team exceeding the lower bound means that it is performing better than the team of



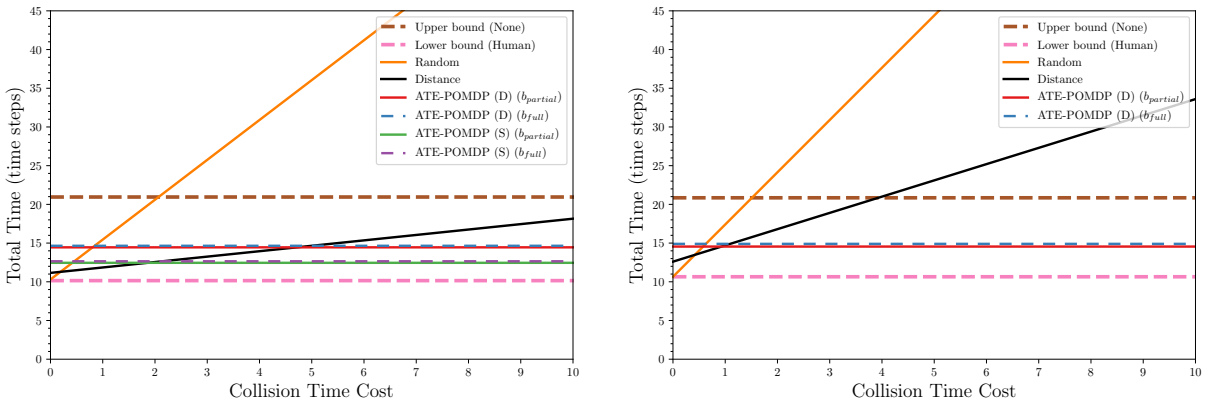
(A) Random order (B) Arranged order

FIGURE 4.11: *Sample* gridworld total times with varying collision cost



(A) Random order (B) Arranged order

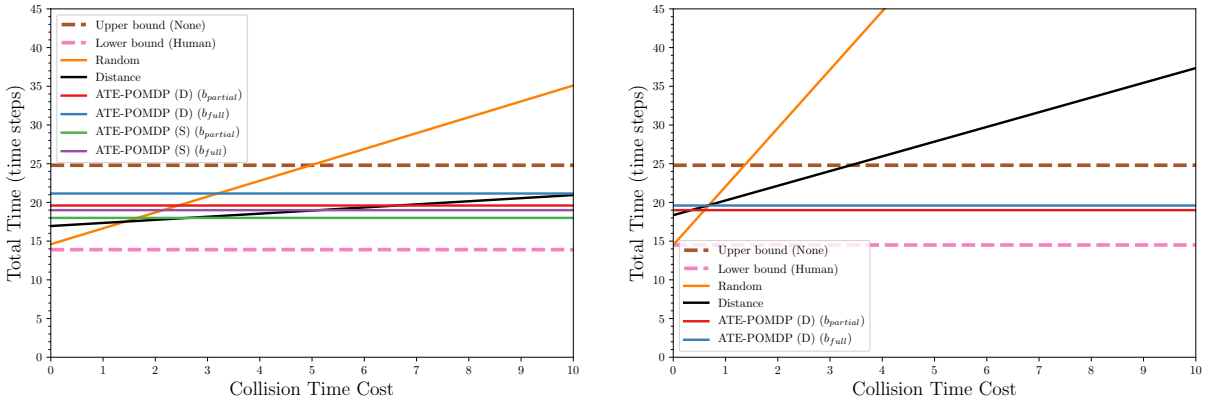
FIGURE 4.12: *5×5A* gridworld total times with varying collision cost



(A) Random order (B) Arranged order

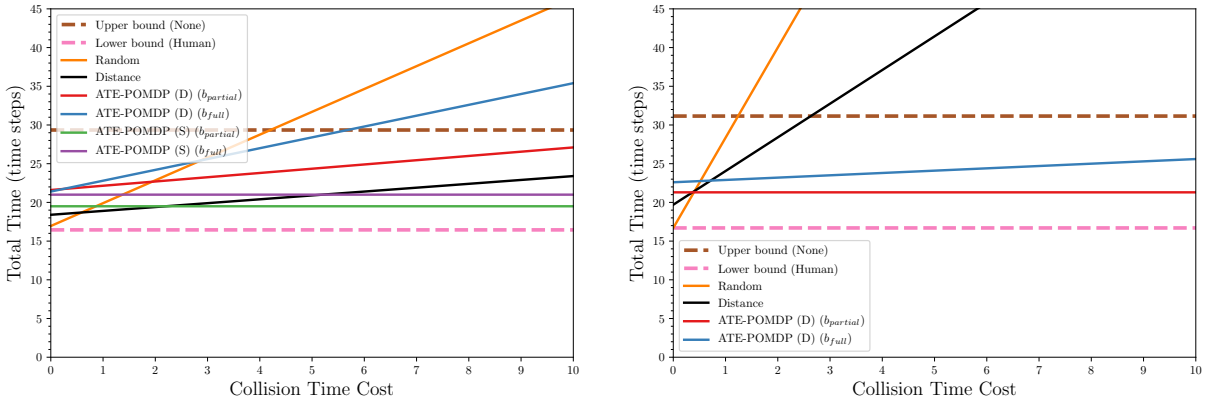
FIGURE 4.13: *5×5B* gridworld total times with varying collision cost

two humans while teams above the upper bound are worse than a single human in performance.



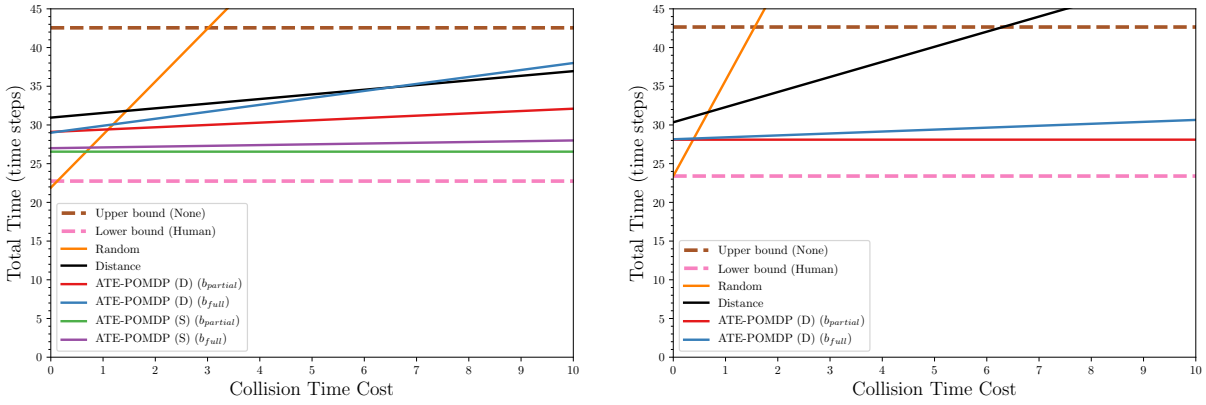
(A) Random order (B) Arranged order

FIGURE 4.14: *5×5C* gridworld total times with varying collision cost



(A) Random order (B) Arranged order

FIGURE 4.15: *6×6* gridworld total times with varying collision cost



(A) Random order (B) Arranged order

FIGURE 4.16: *7×7* gridworld total times with varying collision cost

In an unrealistic setting where time costs for collisions do not exist (c has a value of 0), the robot randomly selecting the next task to complete is good enough to achieve the performance of two humans working in collaboration even with a large number

of collisions. However, increasing the value of c increases the average total time by several magnitudes, indicating that **Random** should not be used as an approach to enable collaboration of a robot with a human. Inferring the intentions of the human remains the best method to enable collaboration and reduce collisions.

In the *Sample* gridworld, **Distance** outperforms all teams, only matched by the **ATE-POMDP (S)** under the $b_{partial}$ setting when the collision time cost is about 8.5. The reason that it performs so well in this environment is because of its symmetrical nature - all objects to collect are fixed distances from one another and the human. As the human moves towards objects, it is very easy for a distance metric to find the true intention of the human for a task - it is the object whereby the distance to the human is shortening over time steps. Once objects are placed in variable locations (such as in the Alternative and Larger gridworlds) and the gridworld symmetry is broken, **Distance** has a number of collisions, especially in the arranged order setting whereby there is an order for the tasks to be completed in. Once the time cost c is above the range of 7.5 time steps for gridworlds with random order and above the range of 1.5 times steps for gridworlds with arranged order, **Distance** is outperformed by both **ATE-POMDP (D)** and **ATE-POMDP (S)**.

The attempts of the **ATE-POMDP (D)** to infer the intention with complete certainty (probability distribution of 1 for a single state) is proven to be more effective for the gridworlds with arranged order. Once the intention of the human is correctly inferred, selecting the corresponding task to follow the order is a matter of changing the reward function - allowing actions that follow the order to give more reward whereas actions that deviate from the order penalise the agent. Even in the random order gridworlds (no order between tasks), **ATE-POMDP (D)** performs similarly to other teams and beating them as the collision time cost increases. **ATE-POMDP (S)** is particularly suited for gridworlds where the completion of tasks can be in any order (random order), outperforming **ATE-POMDP (D)** in these gridworlds. This is at the cost of spending more time during its offline phase in the design to ensure that the correct combinations of tasks have been considered.

The results of the gridworlds in Tables 4.9, 4.10, 4.11 and 4.12 show that as the grids begin to increase in size and objects, **Random** begins to perform worse in terms of an increasing number of collisions while **Distance** continues to perform well enough that it does assist in completing tasks through collaboration at the cost of some collisions. The **ATE-POMDP (D)** scales better with these changing features with less than 1 collision on average in the gridworlds with arranged order under the b_{full} setting and no collisions under the $b_{partial}$ setting. The **ATE-POMDP (S)** scales better in the gridworlds with random order under both belief settings compared to the **ATE-POMDP (D)**. This means that both ATE-POMDP models are much more robust as the gridworlds scale.

The more time steps taken to infer the human's intention lowers the division of work between the human and the robot for **ATE-POMDP (D)** and **ATE-POMDP (S)** with the human completing more work. This is shown by the number of tasks completed by the robot teammate compared to the human teammate under the $b_{partial}$ setting compared to the b_{full} setting. This is due to the initial level of uncertainty - a belief distribution over all states (b_{full}) takes more observations to reduce the uncertainty compared to a belief distribution over a subset of states ($b_{partial}$). The more time spent reducing uncertainty, the more time taken by the robot for its thinking time,

the less work it does and so the human has to do more work. This means that there should be as much effort as possible in reducing the initial belief distribution as this allows the ATE-POMDP models to have a better chance at inferring the true intention quicker and thus enable collaboration faster allowing it to do more work.

While the ATE-POMDP (S) model can be used when the tasks can be completed in any random order, the automation in choosing the next ATE-POMDP (S) model based on the approximation of the large ATE-POMDP as described in section 3.5.1 cannot be achieved by the models themselves. This is due to the fact that the models use assisting actions instead of indicating actions - the task it completes is based on the assisting action that does not give any indication of what the human's new perspective is. In the case of the ATE-POMDP (D), the indicating action that specifies the human's current task can be used to choose the next correct ATE-POMDP (D) perspective model. Before the ATE-POMDP (S) can be used in a real setting, this automation (that is choosing the next correct model based on the human's perspective) needs to be solved as it cannot be achieved based on the agent's own actions (unlike in the case of the ATE-POMDP (D)).

The ATE-POMDP (D) can fail to detect the human's change in perspective due to the location of objects in the grid if they are located near one another. For example in the 7×7 gridworld in Figure 4.9, if the indicated intention of the human was for task **A**, and the robot chose a task, the robot would choose the next ATE-POMDP (D) based on the **A** perspective as it expects the human to now be following some trajectory from this location. However, if the human selected task **B** as its next task, it could complete it before the robot is using its *thinking* component to infer the intention of the human again (it is still completing the chosen task using the *acting* component). As a result, when the ATE-POMDP (D) is chosen, it is using **A**'s perspective when it should be using **B**'s perspective. While this does not affect the ability of an ATE-POMDP (D) to infer the intention, it does affect the automation of the entire robot as it cannot initialise the correct model. Instead of using the approximation of the large ATE-POMDP model using multiple smaller ATE-POMDPs, using the large model would solve this issue as all perspectives are valid at the same time at the cost of more computational requirements.

Another criticism is that the ATE-POMDP model requires the action trajectories of the human or actor whose intention is being inferred, so that they can be used to design and implement the model. This amounts to having an ATE-POMDP curated offline for each person it will collaborate with and this could be tedious and impractical depending on the number of action trajectories this person could have. This could be achieved in realistic settings as a calibration step of sorts - where the human shows the robot the different actions of a trajectories to be aware of them. Alternatively the robot could passively observe the human in the environment for a period of time, approximating the human's action trajectories and attempt to construct the ATE-POMDPs internally and without human intervention. This is left as a future recommendation.

Interestingly it is shown that the ATE-POMDP models can work based on actions alone (without the need for more observations such as location or tools), simplifying the amount of information that must be encapsulated by the models. However it remains important that action trajectories need to be sufficiently distinct from one another at some points to distinguish them which is not always the case. Splitting

the models based on perspectives allows the same starting location and its tasks to be stored together - similar to having separate ATE-POMDP models for kitchen tasks, lounge tasks or outdoor tasks. In the case that only actions are not enough to distinguish between action trajectories, the observation set could be changed to include not only actions but locations as well at the cost of scaling the ATE-POMDP model.

4.4 Conclusion

This chapter presented the experiments conducted in this research to evaluate the effectiveness of ATE-POMDPs in inferring human's intentions to enable collaboration. The simulations use simulated humans whose actions are found using Q-learning. Five metrics are collected and averaged over 20 episodes. These include the number of tasks completed, the thinking time, the acting time, the total time and the number of collisions. There are six teams tested in total: **None** (a single human), **Human** (two human teammates), **Random** (a human and robot that randomly selects tasks to complete), **Distance** (a human and a robot that uses object distance to infer the human's intention), **ATE-POMDP (D)** (a human and a robot that uses the dynamic version of the ATE-POMDP to infer the human's intention) and **ATE-POMDP (S)** (a human and a robot that uses the static version of ATE-POMDP to infer the human's intention for not completing a task). These teams are tested in the *Sample* gridworld, Alternative gridworlds (three 5×5 grids) and Larger gridworlds (6×6 and 7×7 grids). Each of these gridworlds are tested with a random order (objects can be collected in any order) and an arranged order (objects must be collected according to a defined order). The teams that use the ATE-POMDP models consistently have a low number of collisions and scale better than the other teams when the grid size and number of objects are increased. If the time cost of collisions are increased beyond the simulation value of 1 time step, then using an ATE-POMDP for intention inference to enable collaboration is the best choice compared to all the other teams.

Chapter 5

Literature Review

A core focus in the development of robots is to benefit humans in some form - completing a complex task that would take a human too long to complete themselves or improving the quantity or quality of the output. The survey done by Edwards (1984) shows how robots were utilised in industry for mainly factory related tasks such as spot and arc welding, injection moulding and surface treatment in the 1980s, with a steady growth of more robots being added to the workforce with each successive year. The works of Gates (2007) and Dario et al. (2001) describe and review methods that will move robots from factory production settings into different domains such as personal assistants.

The survey done by Ray et al. (2008) evaluates the feelings of people with regards to robots, showing that participants like the idea of robots that are responsible for menial tasks such as cleaning while not wanting robots that are responsible for social activities such as playing with children, keeping someone company or entertaining people. This general idea is also repeated by the survey of Dautenhahn et al. (2005) where participants are open to robots that can be assistants or servants but not friends. This means that robots that can assist a person in their everyday life are very desirable.

These surveys all share the same general idea that with continued advancements in technology, robots will slowly move away from factory settings where they are used for a small number of specific tasks into more complex domains. They will be able to function near or above human performance in those domains, solving problems and completing tasks that will alleviate the difficulty of humans' own lives by taking over the most difficult tasks or tasks that cannot even be attempted by humans.

In the setting of providing assistance to humans in their everyday lives, a model of the humans is required so that they can be understood by the robot and it can provide the optimal responses based on this model. The survey done by Albrecht and Stone (2018) examines different methods utilised by agents to model other agents and identifies seven main agent modelling methods with the work in this dissertation falling into the category known as *Plan Recognition*. Plan recognition is concerned with identifying or inferring the goals or plans of another agent based on their observed actions [Carberry (2001)]. By knowing the intended goal of the agent being modelled, an appropriate response to either assist or hinder depending on the goal of the agent doing the modelling can be selected.

Plan recognition methods utilise a *plan library* that details all the plans or goals of the modelled agent, with these being represented by actions (or states) that can

be observed [Albrecht and Stone (2018)]. These can be tedious to fully implement and may even be incomplete, causing the selection of actions by the agent to be difficult if the modelled agent utilises an unknown plan. Additionally, if the agent is complex in nature, its plans can also scale to be a complex combination of actions and states that may be difficult to encode in a library or increase the library's size to an unusable scale. This plan library is similar to the action trajectories that the ATE-POMDP model encapsulates in its design.

An early example of *Plan Recognition* was the work done by Nguyen et al. (2012) that aimed to create a non-playable character that can assist players in a collaborative game. The game consisted of three fleeing ghosts that must be cornered and destroyed by the player with help from their dog companion. The overall task is decomposed into smaller sub-tasks each consisting of one of the ghosts, the player and the dog, modelled as Markov Decision Processes (MDPs) and solved to determine the optimal joint action of the dog and human per sub-task. The intended ghost (and thus the current sub-task MDP) of the player is determined using a finite state machine and Bayesian belief update whereby the finite state machine allows switching between ghosts without the need for the player to first completely capture it.

The main difference is the type of collaboration being considered - their work considers collaboration on a sub-task that requires joint actions for its completion while the work of this dissertation considers collaboration on tasks that can be completed independently, requiring no assistance on the task itself. Their system comprised of a finite state machine with a Bayesian belief update that achieved similar functionality to a Partially Observable Markov Decision Process (POMDP). Their system allows switching between ghosts being captured and is possible due to having the optimal behaviour of the dog and human for the entire state space, with a notable concern being that larger state spaces could be intractable. While the work in this dissertation did not consider humans that could switch between tasks before completing them - it can be extended to achieve this as deviating from an action trajectory is a trajectory itself - adding this new deviated trajectory into the ATE-POMDP model allows it to deal with humans that switch between tasks. Additionally, with the approximation of a large ATE-POMDP by smaller ATE-POMDP models, scaling is better for different gridworlds as there is no need for solving the entire state space to obtain optimal behaviour as the robot (using the ATE-POMDP) aims to function independently to the human.

The work done by Karami et al. (2009) also aims to tackle plan recognition using a POMDP. The chosen environment consists of a gridworld with an object clearing task whereby the human and robot need to store three objects away in a box with a commitment to ensuring that the robot does not hinder the human's intention for a task with conflicting actions. The state space is decomposed into different sub-tasks depending on the intention of the human towards one of the three objects. The focus of the work lies in being able to compute a policy using different methods that can adapt fast enough to the environment changes, notably the human's intention with respect to the objects they seek to collect and put away. The number of objects is limited to three and the computed policy is tested with a human with random behaviour (the worst case scenario) and closest objects first behaviour although only results of the former are given. Their motivation for using a POMDP is that every time the human is done with a sub-task, the robot can obtain a clearer belief

about the human's intentions by updating its belief.

Their work only ever considers the three objects to put away along with increasing grid sizes and its effects on different POMDP solvers while the work of this dissertation considers not only increased grid sizes but also an increased number of objects and manages to deal with both. In their concluding remarks, they mention for future work that should they solve missions with bigger sub-tasks (specifically more objects), the agent would maintain a belief over the possible chain of tasks of the human. This chaining of tasks is the different combinations of tasks that the ATE-POMDP (D) can infer.

Other noteworthy POMDP related work for Human-Robot collaboration (HRC) is that of Gopalan and Tellex (2015) where the aim is to model HRC as a joint POMDP where there is a common goal and turn taking between the robot and human. While this is shown to be a possible method to achieve HRC, it is not a natural form of collaboration as either the human or robot effectively *"block"* one another due to their turn taking in the model. The work of this dissertation aims to have both the human and robot as their own separate entities that can function independently of one another - neither blocks the other, although there is a dependence in that the ATE-POMDP must be initialised at the correct time so that it can function based on the observations that come from the human.

The work of Karami et al. (2010) aims to use a POMDP along with a new belief space that will determine the human's current intention based on the observed actions. The results show that in a real world situation, the robot can determine the true intention of the human (from two possible intentions) and respond. Their work does not motivate why the standard belief utilised by the POMDP model cannot be used instead of including this new belief space. The work of this dissertation encapsulates all the required belief updates using the standard belief of the POMDP model thus requiring no extra belief spaces or computations.

The work of Görür et al. (2018) aims to evaluate the effects of anticipation on human-robot collaboration by accounting for unexpected human behaviours: actions taken by the human potentially being irrelevant to the goal and the human potentially rejecting the robot's assistance. The selected environment is a conveyor belt whereby boxes need to be picked up and stored in containers by either the human or robot if the human is unable to. Two robot models, reactive (modelled as an MDP) and proactive (modelled as a POMDP), are evaluated with variable humans (modelled as MDPs). The proactive robot can anticipate the human's behaviour and plan its own actions accordingly while the reactive robot only acts once certain perceivable conditions are met. The results indicate that the proactive robot is more efficient in collaborating on the task as it decreases the task duration, increases the success rate and is less intrusive to the human's work, each of these being desired in human-robot collaboration. The work in this dissertation considers anticipation in situations whereby there exists an order for the completion of tasks. Inferring the intention of the human towards a task allows the robot to anticipate the next task that upholds the order and to select it for completion. Additionally the collaboration considered in this dissertation allowed tasks to be completed concurrently.

The beneficial effects of anticipatory action in human robot teamwork settings are further reinforced in the work of Hoffman and Breazeal (2007). The simulated robot worked with humans on a collaborative cart assembly task whereby humans delivered materials and the robot would use the correct tools with the materials to build the cart. Both a reactive and anticipatory robot were tested with the latter showing that it decreased the construction time as it pre-emptively selected tools based on the materials the human was delivering. The work of Huang and Mutlu (2016) also tested anticipation in HRC and provided similar results - decreased task time based on humans selecting fruits for a smoothie with the robot inferring the desired fruits using eye gaze. Anticipation can only be achieved through first inferring the human's intention.

The completion of everyday tasks requires a number of different skills and actions. Finding these using reinforcement learning algorithms may be intractable due to the curse of dimensionality - the larger the problem domain (that is the more states and actions), the more time these algorithms would take to find a solution, if ever [Hengst (2012)]. Hierarchical reinforcement learning (HRL) breaks difficult tasks down into a smaller set of tasks that can be solved and their solutions combined to obtain the solution to the overall difficult task. This means that HRL agents learn not only the primitive actions to solve a problem (like normal RL) but also subroutines - a collection of actions to solve a particular problem [Botvinick (2012)].

HRL works by the idea of abstraction - instead of the agent considering all possible states and actions at the same time, focus on the subset that is important to the current task to be solved [Barto and Mahadevan (2003)]. For example, in a gridworld with designated areas for two separate tasks, first solve the smaller problem of manoeuvring around the gridworld to those designated areas, then solve those two separate tasks by learning the required actions to complete them. After this is done, the manoeuvring can be combined with the actions needed for either task to obtain a behaviour to complete those specific tasks.

Another method of learning the skills required to solve complex tasks is through the use of inverse reinforcement learning (IRL). IRL aims to retrieve the reward function of a MDP having observed the optimal behaviour such that the reward function can explain the behaviour [Ng, Russell, et al. (2000)]. This is a more natural method of learning to solve a task as the robot can simply observe the human and their actions, apply IRL and obtain a reward function that the robot can then use to try and learn the necessary actions for the task itself. The work done by Ranchod et al. (2015) uses IRL for skill discovery by segmenting action trajectories into multiple reward functions to model the multiple skills contained in an action trajectory. This method allows learning the multiple behaviours needed to complete a task, as most tasks in the real world require a number of skills.

The literature reviewed showed that there is a focus on moving robots from specific domains into more general settings such that they can perform on the level of humans. While techniques like hierarchical reinforcement learning and inverse reinforcement learning can be used to learn the necessary actions to solve complex tasks or to learn from human demonstrations respectively, the robots require models of humans to be able to infer their intention so that they can collaborate with them. Partially Observable Markov Decision Processes are one possible approach that can be used to model and infer the intentions of humans to enable collaboration.

Chapter 6

Conclusion

This chapter concludes the dissertation. Section 6.1 summarises the entire dissertation while section 6.2 provides future prospects for this research.

6.1 Conclusion

The phrase *"two heads are better than one"* is a common statement that is used to highlight that working collaboratively can decrease the overall time spent on a task. To achieve this collaboration, inferring a teammate's intention is required, especially if no explicit communication such as speech exists. This can easily be achieved by humans due to their Theory of Mind, the ability to reason about others' behaviours prior to responding, but is significantly harder for robots to achieve.

The main challenge in inferring the intention of a human towards a task is that many tasks can exist in any environment with these tasks sharing similar actions - observing only the actions can initially indicate multiple intentions. Thus, can the actions a human is performing be used to infer their intention? If the intention of the human is found, how can collaboration be enabled with that human?

These two questions are answered by the Action Trajectory Encapsulated Partially Observable Markov Decision Process (ATE-POMDP), the reinforcement learning framework that models and infer the human's intentions. By encapsulating all the action trajectories for tasks the human could be completing, the human's intention can be found and indicated by the agent after observing enough of the human's actions. With the knowledge of the human's intention for a task, collaboration is achieved by simply selecting another task for the robot to complete. This aids the human by lowering the overall work they would have to do in an environment.

Three simulation experiments are run to test collaboration through intention inference using gridworlds of varying sizes, objects and object placements. These are run under two settings: random and arranged order for the collection of objects. Six teams are tested in total: **None** (a single human), **Human** (two human teammates), **Random** (a human and robot that randomly selects tasks to complete), **Distance** (a human and a robot that uses distance to infer the human's intention), **ATE-POMDP (D)** (a human and a robot that uses the dynamic version of the ATE-POMDP to infer the human's intention for completing a task) and **ATE-POMDP (S)** (a human and a robot that uses the static version of ATE-POMDP to infer the human's intention for not completing a task).

The results recorded in the tables in section 4.2 show the ATE-POMDP teams completing the gridworlds with total times slightly longer than the other teams

in the random order setting and generally being the fastest or close to the fastest for the gridworlds with arranged order setting. However, the Figures 4.11 - 4.16 show the effects of increasing the collision time cost from a value of 1 - the value used in the experiments. This value was chosen for the sake of simulation but is unrealistic as it is difficult to quantify the exact time cost of a mistake or collision in the real world. Collisions could cause damage or distract the human such that the entire collaboration effort might be halted to deal with the effects of the collision. The figures show that as the collision time cost increases, the ATE-POMDP teams complete the episodes with the fastest average total time compared to the other teams. This means that if a holistic view was taken in terms of performance based on the total time and the collisions, then ATE-POMDPs outperform the other teams and should be chosen for collaboration through intention inference over the other approaches.

6.2 Future Recommendations

There is a need for solvers that can deal with much larger POMDP models and hence larger ATE-POMDP models without approximating them using multiple smaller ATE-POMDPs. This could be achieved through the use of neural networks to approximate the value function as opposed to having exact solvers calculate the value function. Having larger ATE-POMDP models allows for larger gridworld sizes with increased number of objects to be tested. Furthermore, the *acting* component should be designed and implemented using hierarchical reinforcement learning or inverse reinforcement learning to find the necessary actions to complete tasks.

Another potential avenue to take this work is to apply ATE-POMDPs to environments that have two or more humans to collaborate with. The humans can share a number of tasks and completing those specific tasks would aid both of those humans in completing all their respective tasks. The difficulty comes from being able to infer the intention of two (or more) humans at the same time or deciding on the human whose intention should be inferred first.

The last but most interesting avenue to take this work is in creating ATE-POMDP models for real-world settings. The real-world actions performed by humans to complete tasks do not fall into discrete, equal time steps as they do in a simulation. For example, the action of walking could range from one to five minutes but based on the walking speed, the end location could potentially be the same. This means that even though the trajectories are different, the intention (to reach somewhere) remains the same. This requires trajectories to be learned from real world data indicating a need for cameras and computer vision technology. Real-world trajectories will be much longer than simulated ones which leads back into needing more powerful solvers that can deal with larger POMDP models.

Appendix A

Sample Gridworld ATE-POMDP

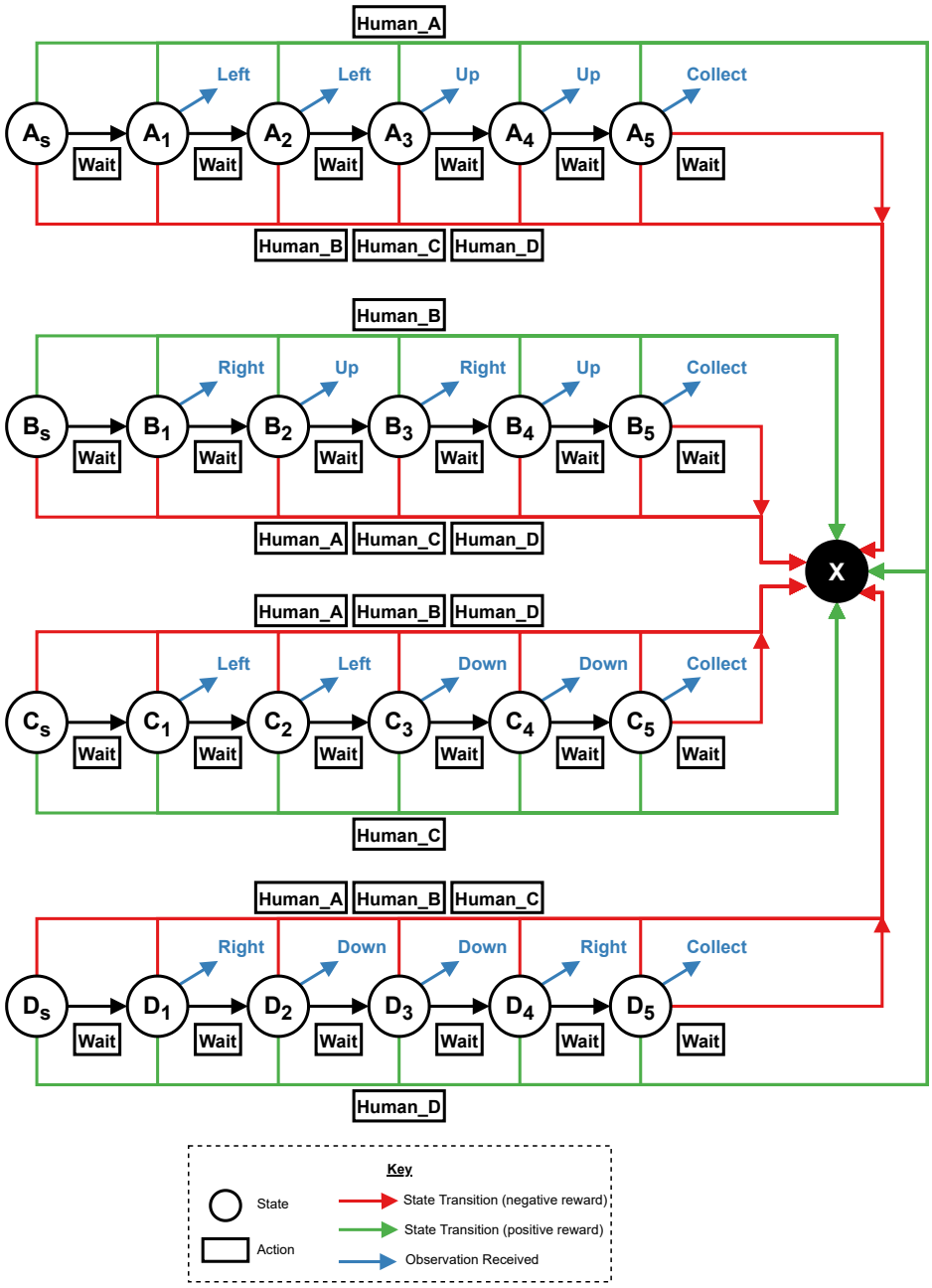


FIGURE A.1: Full ATE-POMDP for the *Sample* gridworld

Appendix B

Multiple Starting Locations

The reasoning for splitting the ATE-POMDPs based on the human's perspectives to reduce uncertainty and model scale is illustrated in this section. The separation of the ATE-POMDP into multiple models allows solutions to these models to be found much more easily as the models are smaller compared to solving one large ATE-POMDP. Action trajectories from the same starting location can be grouped together to reduce overall uncertainty.

The *Two-Starts* gridworld shown in Figure B.1 illustrates an environment whereby the human has two potential starting locations, $\mathbf{H_1}$ and $\mathbf{H_2}$ (both given by the colour blue). Figures B.2 and Figure B.3 show the resulting action trajectories based on the starting location of the human (obtained by Q-learning).

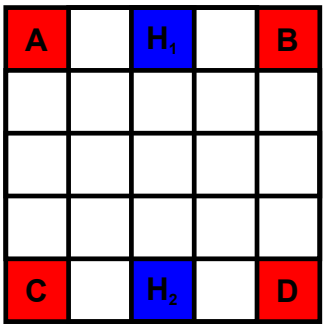


FIGURE B.1: *Two-Starts* gridworld

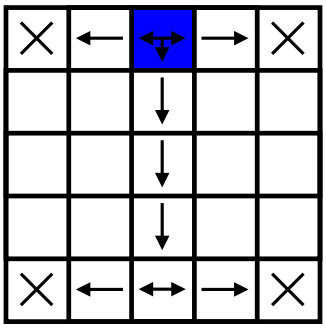


FIGURE B.2: $\mathbf{H_1}$ action trajectories

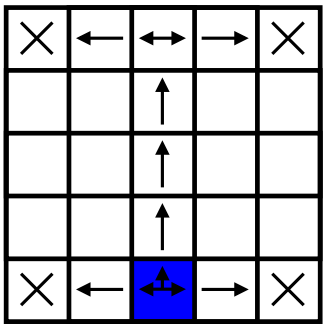


FIGURE B.3: $\mathbf{H_2}$ action trajectories

From the 8 total action trajectories shown by Figures B.2 and B.3, there exist only 4 that are distinct enough to be assigned to one particular intention based on the starting location in the ATE-POMDP model. These 4 trajectories are $\mathbf{H_1}$**-C**, $\mathbf{H_1}$**-D**, $\mathbf{H_2}$**-A** and $\mathbf{H_2}$**-B**. Regardless of the true starting location ($\mathbf{H_1}$ or $\mathbf{H_2}$), the human's intention towards these tasks can be detected correctly as they are unique compared to one another and the four remaining trajectories.

The other 4 action trajectories form pairs whereby the actions overlap exactly making distinguishing one trajectory from the other impossible. The first pair [$\mathbf{H_1}$**-A**,$\mathbf{H_2}$**-C**] has the action trajectory of `left`, `left`, `collect` while the second pair [$\mathbf{H_1}$**-B**,$\mathbf{H_2}$**-D**] has the action trajectory `right`, `right`, `collect`. Due to the nature of POMDPs, mapping a single action to a belief distribution, the same indicating action will be used for both intentions resulting in one correct and one incorrect intention inferred.

This means that for the *Two-Starts* gridworld, from the 8 possible human intentions, at most only 6 will ever be indicated correctly by the ATE-POMDP. The easiest solution to this problem is to create a model for each perspective, with a perspective referring to a starting location and how the goals are collected by the human from this location. That means that there would be two separate ATE-POMDP models, one for Figure B.2 and one for Figure B.3. Knowing when to use one model over another can by determined through trial-and-error using the reward function. One model is initialised to collaborate with the human and based on the received rewards it can be determined whether this is the right model as negative rewards mean it is the incorrect model while positive rewards mean it is the correct model. As the models will contain all the trajectories for a specific starting location, the scale of the model and the complexity of finding an optimal policy can be reduced as all action trajectories are not stored in one model.

To have only a single large ATE-POMDP model that would encapsulate all 8 trajectories, there needs to be some other identifying information in the observations to distinguish action trajectories that are identical. This could take the form of having the observation set of the ATE-POMDP not only use actions but locations as well. While this is a possible approach to only having a single model, the computational complexity begins increasing as the number of observations increases as well. An observation function based on the action set A of the human results in $|A|$ observations while a large ATE-POMDP model that includes the action set A and states space S results in $|A| * |S|$ observations. As the state space S is generally much larger than the action set A, this would drastically scale the observation set Z and make solving the POMDP significantly harder if not intractable. The multiple ATE-POMDP models allow the computational complexity to be manageable so that solutions can be found and the model will not scale to intractability.

Bibliography

Albrecht, S. V., & Stone, P. (2018). Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, *258*, 66–95.

Baron-Cohen, S., Leslie, A. M., & Frith, U. (1985). Does the autistic child have a "theory of mind"? *Cognition*, *21*(1), 37–46.

Barto, A. G., & Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete event dynamic systems*, *13*(1), 41–77.

Bauer, A., Wollherr, D., & Buss, M. (2008). Human-robot collaboration: A survey. *Int. J. Humanoid Robotics*, *5*, 47–66.

Bellman, R. (1957). A markovian decision process. *Journal of mathematics and mechanics*, *6*(5), 679–684.

Bluethmann, W., Ambrose, R., Diftler, M., Askew, S., Huber, E., Goza, M., Rehnmark, F., Lovchik, C., & Magruder, D. (2003). Robonaut: A robot designed to work with humans in space. *Autonomous robots*, *14*(2-3), 179–197.

Botvinick, M. M. (2012). Hierarchical reinforcement learning and decision making. *Current opinion in neurobiology*, *22*(6), 956–962.

Byom, L. J., & Mutlu, B. (2013). Theory of mind: Mechanisms, methods, and new directions. *Frontiers in human neuroscience*, *7*, 413.

Carberry, S. (2001). Techniques for plan recognition. *User modeling and user-adapted interaction*, *11*(1-2), 31–48.

Cassandra, A. R., Littman, M. L., & Zhang, N. L. (2013). Incremental pruning: A simple, fast, exact method for partially observable markov decision processes. *arXiv preprint arXiv:1302.1525*.

Cassandra, A. R. (1998). *Pomdp-solve*. Retrieved March 28, 2021, from http://www.pomdp.org/code/

Dario, P., Guglielmelli, E., & Laschi, C. (2001). Humanoids and personal robots: Design and experiments. *Journal of robotic systems*, *18*(12), 673–690.

Dautenhahn, K., Woods, S., Kaouri, C., Walters, M. L., Koay, K. L., & Werry, I. What is a robot companion-friend, assistant or butler? In: *2005 ieee/rsj international conference on intelligent robots and systems*. IEEE. 2005, 1192–1197.

De Farias, D. P., & Van Roy, B. (2003). The linear programming approach to approximate dynamic programming. *Operations research*, *51*(6), 850–865.

Edwards, M. (1984). Robots in industry: An overview. *Applied ergonomics*, *15*(1), 45–53.

Egorov, M., Sunberg, Z. N., Balaban, E., Wheeler, T. A., Gupta, J. K., & Kochenderfer, M. J. (2017). POMDPs.jl: A framework for sequential decision making under

uncertainty. *Journal of Machine Learning Research*, *18*(26), 1–5. http://jmlr.org/papers/v18/16-300.html

Gates, B. (2007). A robot in every home. *Scientific American*, *296*(1), 58–65.

Gilbert, G. R., & Beebe, M. K. (2010). *United states department of defense research in robotic unmanned systems for combat casualty care* (tech. rep.). Army Medical Research, Material Command Fort Detrick MD Telemedicine, and Advanced Tech Research Center.

Gleeson, B., MacLean, K., Haddadi, A., Croft, E., & Alcazar, J. Gestures for industry intuitive human-robot communication from human observation. In: *2013 8th acm/ieee international conference on human-robot interaction (hri)*. IEEE. 2013, 349–356.

Gopalan, N., & Tellex, S. Modeling and solving human-robot collaborative tasks using pomdps. In: *Rss workshop on model learning for human-robot communication*. *32*. (4). 2015, 590–628.

Görür, O. C., Rosman, B., Sivrikaya, F., & Albayrak, S. Social cobots: Anticipatory decision-making for collaborative robots incorporating unexpected human behaviors. In: *Proceedings of the 2018 acm/ieee international conference on human-robot interaction*. 2018, 398–406.

Green, S. A., Billinghurst, M., Chen, X., & Chase, J. G. (2008). Human-robot collaboration: A literature review and augmented reality approach in design. *International journal of advanced robotic systems*, *5*(1), 1.

Hengst, B. Hierarchical approaches. In: *Reinforcement learning*. Springer, 2012, pp. 293–323.

Hoffman, G., & Breazeal, C. Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team. In: *Proceedings of the acm/ieee international conference on human-robot interaction*. 2007, 1–8.

Huang, C.-M., & Mutlu, B. Anticipatory robot control for efficient human-robot collaboration. In: *2016 11th acm/ieee international conference on human-robot interaction (hri)*. IEEE. 2016, 83–90.

Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial intelligence*, *101*(1-2), 99–134.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, *4*, 237–285.

Karami, A.-B., Jeanpierre, L., & Mouaddib, A.-I. Human-robot collaboration for a shared mission. In: *2010 5th acm/ieee international conference on human-robot interaction (hri)*. IEEE. 2010, 155–156.

Karami, A.-B., Jeanpierre, L., & Mouaddib, A.-I. Partially observable markov decision process for managing robot collaboration with human. In: *2009 21st ieee international conference on tools with artificial intelligence*. IEEE. 2009, 518–521.

Kochenderfer, M. J. (2015). *Decision making under uncertainty: Theory and application*. MIT press.

Littman, M. L. (2009). A tutorial on partially observable markov decision processes. *Journal of Mathematical Psychology*, *53*(3), 119–125.

Ng, A. Y., Russell, S. J. et al. Algorithms for inverse reinforcement learning. In: *Icml*. *1*. 2000, 2.

Nguyen, T.-H. D., Hsu, D., Lee, W.-S., Leong, T.-Y., Kaelbling, L. P., Lozano-Perez, T., & Grant, A. H. (2012). Capir: Collaborative action planning with intention recognition. *arXiv preprint arXiv:1206.5928*.

Pineau, J., Montemerlo, M., Pollack, M., Roy, N., & Thrun, S. (2003). Towards robotic assistants in nursing homes: Challenges and results. *Robotics and autonomous systems*, *42*(3-4), 271–281.

Puterman, M. L. (1990). Markov decision processes. *Handbooks in operations research and management science*, *2*, 331–434.

Ranchod, P., Rosman, B., & Konidaris, G. Nonparametric bayesian reward segmentation for skill discovery using inverse reinforcement learning. In: *2015 ieee/rsj international conference on intelligent robots and systems (iros)*. IEEE. 2015, 471–477.

Ray, C., Mondada, F., & Siegwart, R. What do people expect from robots? In: *2008 ieee/rsj international conference on intelligent robots and systems*. IEEE. 2008, 3816–3821.

Spaan, M. T. Partially observable markov decision processes. In: *Reinforcement learning*. Springer, 2012, pp. 387–414.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine learning*, *3*(1), 9–44.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*(3-4), 279–292.

Zhang, N. L., & Liu, W. (1996). *Planning in stochastic domains: Problem characteristics and approximation* (tech. rep.). Technical Report HKUST-CS96-31, Hong Kong University of Science and Technology.

Zhang, Y.-F., Zhang, Q.-F., & Yu, R.-H. Markov property of markov chains and its test. In: *2010 international conference on machine learning and cybernetics*. *4*. IEEE. 2010, 1864–1867.