

# A Method for Embodied Co-Learning in Interdependent Human-Robot Teams

## MSc Thesis Report

H.W. Veldman-Loopik





# A Method for Embodied Co-Learning in Interdependent Human-Robot Teams

**MSc Thesis Report**

by

Hugo W. Veldman-Loopik

to obtain the degree of  
**Master of Science**

At the:  
Department of Cognitive Robotics  
Delft University of Technology  
to be defended publicly on Thursday July 6, 2023 at 10:00 AM.

Student number:

4478231

Supervisor and Assessment committee:

Dr. L. (Luka) Peternel

(Chair, Supervisor)

Prof. dr. ir. D.A. (David) Abbink

(Member, Supervisor)

Dr. ing. M.C. (Marco) Rozendaal

(Member)

Prof. dr. ir. M. (Martijn) Wisse

(Member)

Ir. E.M. (Emma) van Zoelen

(Supervisor)





# Contents

<b>Preface</b>	<b>v</b>
<b>1 Paper</b>	<b>1</b>
<b>A Visualization of all the episodes</b>	<b>18</b>
<b>B Action preference in phase 2 for all teams</b>	<b>20</b>
<b>C Questions of the questionnaire</b>	<b>24</b>
<b>D Complete results of the questionnaire</b>	<b>25</b>
<b>E Overview of task selection process</b>	<b>26</b>



# Preface

I am very proud to finally present my final thesis report. On one hand, I am a little bit sad, but mostly I am very relieved and proud that the end of this project is in sight at last. Before we dive into the matter, however, I want to thank some people, without whom I could not have done it.

Firstly, I would like to thank Luka Peternel and Emma van Zoelen for their invaluable, positive and professional guidance and support during both my literature review and this Master thesis. They both always had time for me when I needed support. I especially want to thank Luka for his faith in me throughout my entire Master studies.

Next, I wish to thank David Abbink, Marco Rozendaal and Martijn Wisse, for sitting on my committee. Specifically, I want to thank David for his support, helpful feedback and his suggestions for the upcoming presentation.

During the project I spent a lot of time in the lab with the robot. During this time Leandro de Souza Roza, Micah Prendergast and Nicky Mol were always ready to help when I ran into trouble.

Thanks also goes to my dad Alex Loopik and my study-buddy Susanna Halman. Both were a great help in reading my concept texts and providing valuable advice. I would not have been able to write and complete this thesis without the constant support of my wife Doris Veldman.

Finally, I would like to thank all the participants in my study for their time and willingness to participate in my experiment.

Hugo Veldman-Loopik  
*Delft*, June 2023



1

Paper

# A Method for Embodied Co-Learning in Interdependent Human-Robot Teams

Hugo Veldman-Loopik

supervised by Emma van Zoelen, David A. Abbink and Luka Peternel

**Abstract**—This paper addresses the research question: “How can a human-robot team achieve co-learning, and interdependence in physically embodied tasks?”. A method has been developed that enables a human-robot team to co-learn the handover of an object from the robot to the human. Five design requirements were composed to address the challenges of human-robot co-learning in physically embodied environments. The method is based on a Q-learning algorithm that was adapted and extended to meet these requirements. An experiment was conducted with six participants. For every human-robot team, each design requirement was qualitatively evaluated. Interdependent co-learning was identified in three of the six teams. The limitation of the design, and how this method can be improved further, was discussed. The method, presented in this paper, demonstrates how human-robot co-learning and interdependence can be enabled in physically embodied tasks.

## I. INTRODUCTION

Human-robot interaction has rapidly evolved in the last decade [1], [2]. Robots are being used in various industries such as manufacturing [3], [4], healthcare [5], and transportation [6], [7]. The latest developments in this area involve the addition of machine learning to collaborative robotic systems [8]. One of the key challenges in this field is to enable self-learning robots not only to improve their performance, but to use learning to improve collaboration with humans [9]. Co-learning is a collaborative learning process between humans and robots, where they both learn simultaneously how to collaborate effectively [10], [11]. Human-robot co-learning can be used to improve performance and personalize the robot behavior to the human [12]. Recent research shows promising results in using co-learning to improve human-robot fluency and interdependence in human-robot teams [10], [11], [13], [9].

However, while this type of co-learning has been studied in virtual [11] and in simulated environments [10], there is still limited research in co-learning in physically embodied environments. This gap is due to the complexity of the physical environment, which involves real-world interactions and unpredictability that cannot be fully replicated in simulations [14].

The development of co-learning in physically embodied environments can have significant implications in the field of human-robot interaction. It could lead to the creation of robots that are better able to adapt to their human team members and the environment, so that they can assist humans in complex tasks in a more personalized way. Additionally, co-learning could enable robots to learn with their human partners in real-time, leading to a more natural and efficient interaction.

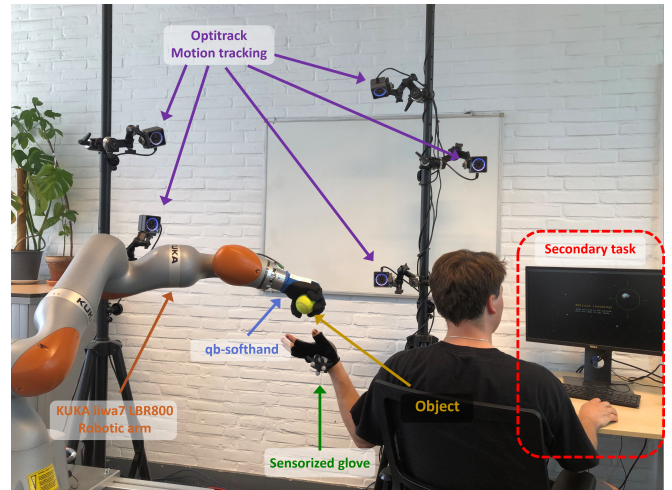


Fig. 1: Experiment setup for the human-robot co-learning of an object handover task. The robot consists of the KUKA iiwa7 LBR800 robotic arm with the qb-softhand attached. The Optitrack motion tracking system is used to track the pose of the human hand via a sensorized glove. The human is also performing a secondary task, as explained in subsection III-B.2. The experiment and the setup are explained in more detail in section IV.

### A. Research question and Scope

This paper addresses the outlined research gap by exploring the challenges and opportunities of co-learning in physically embodied environments. Specifically, we will investigate the following research question: “How can a human-robot team achieve Co-Learning, and interdependence in physically embodied tasks?”

In this research, we narrow our focus to the specific context of human-robot teams that involve one human and one robot. Secondly, we focus only on reinforcement learning (RL) techniques to enable robot learning and do not consider direct learning from demonstration since it limits self-exploration. This was further narrowed down to Q-learning specifically. Lastly, we focus our investigation on a handover task, which involves the transfer of an object from the robot to the human. This task is a good use case since it requires both spatial and temporal coordination between two agents and has many ways to perform it, thus offering a large learning space to explore.

### B. Challenges of exploring co-learning

In co-learning, team members learn together how to collaborate effectively, by finding strategies that work for them as a team [10], [11], [9]. This means that both agents learn



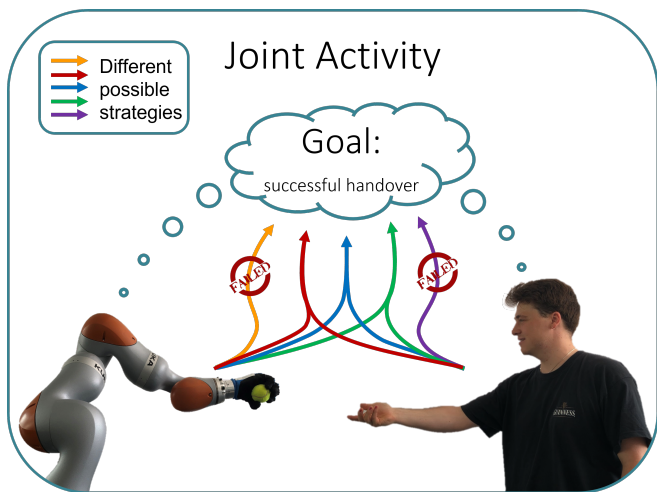


Fig. 2: A schematic demonstration of how different strategies can lead to the same outcome, and how joint activity is only achieved when the individual strategies of the agents are congruent. Congruent strategies are visualized as arrows of the same color.

simultaneously which strategies allow them to collaborate as one interdependent, symbiotic unit [13], [15]. Take for instance a hand-over task (Figure 1) where a robotic arm has to hand over an object to its human team member. The robot can choose the strategy of dropping the object above the hand of the human, trusting that the human will catch it. Alternatively, the robot can hold the object close to the human, allowing the human to just seize it. The human, on the other hand, also has to choose a strategy: they could hold their hand up, hoping the robot will drop the object in their hand, or the human could assume the robot will hold the object until they seize the object. If the robot chooses the first strategy, while the human chooses the latter, the object will fall on the ground. Hence, for effective collaboration, the human and robot have to pick congruent strategies. In Figure 2 it is visualized how the team members can use different strategies to reach the same goal. Some strategies have a congruent counterpart, shown in the same color. Only if both agents execute a congruent strategy, the task can succeed.

To achieve co-learning, the team must be able to explore different strategies and learn what strategy works for them as a team. Thus, joint activity can be achieved in various ways, making co-learning an open-ended process with multiple possible ways that result in similar outcomes. Therefore, the outcome alone, will not provide enough information to identify whether a team was co-learning. Co-learning can even appear without an immediate performance increase, as the essence of co-learning is improving the collaboration in the team. The improvement of performance is only a consequence of co-learning. Therefore, there is not a single metric that can identify co-learning in human-robot teams. Instead, the team dynamics have to be assessed by quantitative analyses of the development of strategies and interaction patterns [11] in the team. In other words, co-learning can only be identified with qualitative, case-by-case, analysis of collaboration in human-robot teams.

### C. Structure of the paper

To answer the research question, we developed a method for co-learning an embodied task, and examined this method qualitatively on six human-robot teams. This way we explored the effects of specific design aspects of the method and how these aspects enable co-learning and interdependence. Hence, the main research question is answered in two steps.

The first step was to develop a new method that consists of two physically embodied parts: a collaborative human-robot task and a RL technique for the robot. To do so, five design requirements were composed. This was done based on prior literature research [16] which is recapitulated in section II. How these design requirements were composed and how the developed method was designed is elaborated in section III.

The second step is to qualitatively examine our newly developed method, and explore how the design requirements contribute to co-learning and interdependence within the human-robot team. We did this by conducting an experiment where six participants performed the developed task in collaboration with our RL agent. The experimental design is explained in section IV. The results are displayed and interpreted in section V. These results are qualitatively interpreted for each human-robot team to examine which design aspects did and did not contribute to the interdependence in that team. Then, in section VI, we reflect on the method, the experiment, and our findings to gain a deeper understanding of how embodied co-learning can be achieved in human-robot teams. In this section, we also give potential improvements for the method and recommendations for future research. Finally, in section VII, we conclude this research by recapitulating the answer to the research question.

## II. BACKGROUND

In this section, the relevant parts of a prior literature review [16] are summarized, to provide background information on how interdependence can be achieved using RL and on how standard Q-learning works, as Q-learning forms the foundation of the RL algorithm that we developed.

### A. Co-learning and interdependence

The most important aspect for team members to collaborate [15], [17], [13] and more specifically to co-learn is interdependence [11], [10], [9], [18]. So in this section we explain how an interdependent relationship between two agents can emerge. Interdependence between two team members can be seen as a symbiotic relationship, where the team members allow themselves to depend on each other to increase task performance and efficiency. The concept of interdependence is often used in studies on team collaboration [15], collaborative performance [18], [15], team task design [13], [11] and team learning [9], [10]. It is established by concepts like responsibility and regular dependence.

Interdependence is an essential aspect of co-learning, as co-learning is learning as a team rather than collaborating agents learning individually. So, to be able to achieve this,

the team must learn how to operate as one interdependent unit.

According to Tal [15], there are two main parts that determine the level of interdependence: means and outcomes. The means, in this context, consists of the capabilities and dependencies described in subsection II-A.1. The outcomes on the other hand are effected by how complementary the group goals and rewards are. How the later can be achieved is explained in subsection II-A.2.

1) *Dependencies*: Interdependence starts with dependence. Team members are considered dependent, when their individual capacity is not enough to complete a task, but their combined capacity is [13]. In other words, they both need each other in order to succeed. This type of dependence is referred to as *hard-dependence*[13]. In an interdependent team, however, there is also *soft dependence*[13]. Soft-dependencies, or opportunistic dependencies, are dependencies between team members that are not strictly needed to achieve the group goal, but they arise from opportunities to perform better as a team. Hence, the team chooses a strategy, where the individual team members are dependent on each other's actions to complete a part of the task in a better way.

Soft dependencies are key to enable an interdependent relationship [13]. Moreover, the emergence of soft dependencies, recursively adds to the level of interdependence in the team. Johnson [13] calls this the "cascading effect". So when interdependence is established, and soft dependencies can arise, retaining and strengthening the interdependent relationship. This means that soft dependencies are most likely to emerge when an interdependent relationship is already established. Thus, this cascading effect must be kick-started with some mutual hard-dependence [13], and enough room for soft dependencies. In the example of the handover task (Figure 1), there is mutual hard-dependence, as a handover can not be accomplished alone. The soft dependencies in this example can arise due to the multiple possible strategies that allow completion, visualized in Figure 2. If the human for instance chooses the red strategy, without first making sure that the robot does this as well, they allow themselves to be dependent on the robot to choose the red strategy as well, and therewith a soft dependency has emerged.

2) *Reward and Shared goal*: For interdependence to arise, both team members must have interest in the same outcome [10], [13], [15]. In other words, the team should have the same goal. To achieve this in a human-robot team, to think about how the robot could be rewarded.

Akalin and Loutfi [2] organize RL techniques in Social Robotics into two relevant categories: *interactive RL*, *task performance driven techniques*. The learning algorithms from the first category (*Interactive RL*), are based on feedback from interaction with the human. Two well known examples are: TAMER [19] and COACH [20]. Such *interactive RL*-techniques, are based on an actor-critic relationship where the human (critic) provides the reward or feedback to the robot (actor). However, in order to create a good team, team members should be equals without such a hierarchy [17], [15]. Therefore, if such a technique is to be used for

co-learning, the robot should also provide feedback to the human, to avoid unbalanced hierarchy.

Alternatively, a learning algorithm from the category *task performance driven techniques* can be used. In this type of RL, the robot is rewarded based on the performance of the task. This category has a promising potential for co-learning, as it would enable the human and robot to have a shared goal. In other words, when the robot and the human are both reward on the same collaborative performance, they intrinsically are motivated to collaborate and work as a team. Furthermore, rewarding the robot based on overall performance would be most feasible for co-learning, since the optimal policy of the robot is dependent on the behavior of both team members [10]. For this reason, we focused on *task performance driven techniques*, where the robot is rewarded based on the collaborative performance of the team. In subsection III-C.2 it is further elaborated how we designed a reward function that ensures interest in the same outcome for both agent.

## B. Q-learning

In combination with the right reward function, Q-learning is such a *task performance driven technique*. In the method proposed in this paper, we choose to use a Q-learning based RL algorithm as explained in more detail in subsection III-C. In this subsection we provide background information on how Q-learning works.

Q-learning is a fundamental reinforcement learning technique, rooted in dynamic programming [21]. It relies on the concept of assigning a quality to each action in each state. These qualities are represented by Q-values, and stored in a Q-table, which serves as a basis for determining the optimal action to take in each state. This table that stores a Q-value for each state-action pair, is referred to as the Q-table, Q-function or value function, and it describes the current a policy of the agent. Initially, the Q-values are unknown and need to be learned through the RL algorithm. To accomplish this, all states are assigned a reward ( $R(s)$ ), which reflects the task at hand. By considering the reward of a reached state ( $R(s')$ ) and potential future states, the quality of a state-action pair ( $Q(s, a)$ ) can be fully assessed.

To learn the Q-values, an iterative process is employed using the Bellman equation (1). The future reward is estimated by considering the Q-values of future state-action pairs, determined by the policy based on the current Q-function. Essentially, when an action is taken, the Q-value for that state-action pair ( $Q(s, a)$ ) is updated by summing the reward obtained upon entering the next state ( $R(s')$ ) and the highest known Q-value for the next possible state-action pair:

$$Q_*(s, a) = E[R(s') + \gamma \max_{a'} Q_*(s', a')] \quad (1)$$

This maximum Q-value in the next state ( $\max_{a'} Q_*(s', a')$ ) represents the expected cumulative future reward, recursively accounting for all future rewards. A discount factor ( $\gamma$ ) is applied to discount the expected future reward, ensuring that each successive reward contributes proportionally less to the

Q-value as it extends into the future. This guarantees the convergence of Q-values to a finite limit [21]. Through iterative updates, the Q-values converge to the optimal Q-function,  $Q_*(s, a)$  (1), indicating the attainment of an optimal policy.

The to be learned task, is often broken up into episodes. An episode refers to a complete sequence of interactions between an agent and its environment. It starts with the agent being in an initial state and progresses through a series of actions, transitions to subsequent states, and receiving corresponding rewards. The episode concludes when the agent reaches a terminal state or a predefined stopping condition. Each episode provides an opportunity for the agent to learn from its experiences and refine its decision-making process to achieve optimal performance.

### III. METHODS

This section describes the developed method for co-learning a physically embodied task in a human-robot team. This is done by introducing a set of design requirements in subsection III-A. These requirements were used to provide guidance during the development of the method. Next, subsection III-B describes, the specific task that is to be co-learned in this method, and it explains why it is suitable for an attempt at co-learning a physical embodied task in a human-robot team. Lastly, subsection III-C explains how the method and the learning algorithm is realized, as well as how this makes the method meet the defined design requirements.

#### A. Design Requirements

To design our method, we defined five design requirements based on extensive literature research [16]. These requirements are examined individually below. All five are important with respect to achieving seamless co-learning, as they outline the challenges of human-robot co-learning in physically embodied tasks. Essentially, these design requirements define the problem space to which the method provides a solution, and cover all aspects of co-learning. In other words, the design requirements are defined in such a way that co-learning is present if they are met.

1) *Dependencies*: We aim to assure that an interdependent relationship between the human and the robot is formed in order to enable co-learning through the cascading effect [13] described in subsection II-A.1. This is done by ensuring some hard dependencies between the human and the robot and creating opportunities for soft dependencies to emerge during the co-learning process, in order for interdependence to grow. So the first requirement is:

**R1** The method ensures hard dependencies and allows for soft dependencies between the human and the robot, in both directions.

2) *Learning pace*: Co-learning is most likely to succeed when both agents learn at a similar pace, to avoid a disbalance in contribution over time. If one team member learns faster than the other one, it might outperform it. This could cause it to slowly lose its motivation for soft-dependencies,

as it is better off doing it alone than being dependent on its inferior team member. It could also cause a hierarchy in the team that could be harmful for the interdependence [15]. Our second design requirement therefore states:

**R2** The Robot has the ability to learn at the same pace as the human team member.

3) *Shared Goal*: A design requirement that ensures that both team members have the same goal [10], [13], [15] is crucial to make sure that the agents converge to congruent strategies as outlined by Figure 2. This can be done by rewarding both team members based on their collaborative performance.

Furthermore, to allow for the development of various strategies and team dependencies throughout the learning episodes, we avoided overly constraining the team's learning process by only rewarding the team at the end of each episode and not giving any intermediate rewards. In other words, giving the team complete freedom in their choice of policies, and only rewarding the team based on their end result, encourages them to find strategies and soft dependencies that work for them, contributing to their interdependence as a team. We capture this in the design requirement:

**R3** Both the human and the robot are rewarded similarly, based on their collaborative performance.

4) *Adaptability*: In co-learning it is important that the robot algorithm stays adaptable to change. This is because the human team member also learns and might therefore change its behavior later on, with the possible effect that certain state-action pairs, that previously were discarded, now should be preferred due to the change of policy by the human. To enforce this, we defined a requirement that ensures that the robot always keeps exploring, to maintain its adaptability to changing human behavior.

**R4** The RL algorithm can continuously adapt its behavior during all stages of the learning process.

5) *Observability*: Lastly, observability should not be overlooked, as it is one of the fundamentals for predictability and directability [13], [9]. This requirement states that the robot algorithm should be able to observe the state and actions of the human team member. When looking from a broader perspective, however, not only the RL agent should have observability, but both team members should be able to observe each other for co-learning to be possible.

Moreover, both agents should meet this criterion to a similar extent to avoid hierarchical inequalities within the team. For example, when the robot is barely able to observe the human, but the human can fully observe the robot, an imbalance might impair the equality in the team. This leads to the fifth design requirement that facilitates communication and avoids any hierarchical imbalances.

**R5** The human and the robot must be able to observe each other's state and actions, and neither should have any observability advantages.

## B. The Task

To develop this co-learning method a suitable task had to be designed first. We decided that a human-robot handover task [22], [5] was most appropriate. Passing an object involves multiple elements where soft dependencies can arise. For instance, the position and orientation at which the object is handed over need to be predicted or learned. As described in section I, another example of a soft dependency that can arise, is that the team can learn either of the following strategies: a) the robot drops the object while the human holds its hand up, or b) the robot conveys the object close to the human until the humans seizes it.

To coordinate this specific moment, where the responsibility of not dropping the object, switches from one to the other agent, the agents must by definition collaborate, to successfully complete the task. This ensures that, additional to the soft dependencies that can arise here, a mutual hard dependency is embedded in the task itself. This is what makes a handover task stand out compared to other collaborative tasks that are often seen in HRC, such as for instance polishing [23], [24], sawing [25], and or assembly tasks [7], [26]. Thus, handing over an object between a human and a robot very suitable to meet **R1**.

Additionally, the task of handing over an object is relatively short and can either succeed or fail. It is ideal for rewarding the team based on their collaborative performance (**R3**), and, as it is a short task, the team can rehearse the task often in a short amount of time. Therefore, the robot gets often rewarded, allowing it to update its policy regularly. This contributes to requirement **R2**, as it improves the learning pace of the robot.

Moreover, this type of task allows for various implementations depending on the circumstances and context. This gives it the opportunity to be shaped to meet the other requirements, such as **R5**. In this subsection, explains how a handover task is developed for achieving physical co-learning in human-robot team.

To accommodate **R1** even better, the task was designed in a way that responsibilities are divided over both agents, creating dependencies between the human and the robot. Some of these responsibilities are given to a specific agent by design, to ensure hard dependencies. Other responsibilities will still have to be distributed by the agents themselves during the learning process, creating the opportunity for soft dependencies to arise. This was done by carefully designing the capabilities of both agents during the task. These capabilities include their possible actions and their ability to observe the environment. The design of these capabilities is based on all the design requirements. In this subsection, the details of the handover task that the human-robot team will learn in the developed method is explained. First, subsection III-B.1 describes the capabilities of the robot, which are determined by its state-action space. Next, subsection III-B.2 explained how we established a fixed set of capabilities for the human, by creating a secondary task

that limits the human ability to act as well as their ability to observe the environment.

1) *State-Action space (Robot)*: To meet **R2**, the state-action space of the RL agent should be designed as such that it enables a sufficient learning pace for the RL algorithm. Foremost, we want to keep the state-action space as small as possible. Since a Q-value has to be determined during the learning process, for each possible state-action pair, in order to learn which action to take given what state, as explained in subsection II-B. This means that the amount of to be learned Q-values is equivalent to the amount of possible actions times the amount of possible states. Thus, by keeping the state-action space small, we can reduce the amount of Q-values that have to be learned and therewith, decreasing the amount of trials needed to learn the value function, and increasing the overall learning pace. Moreover, having less Q-values overall increases adaptability, as fewer Q-values have to change in order to change the policy. This contributes to **R4**.

We designed the capabilities of the robot as such, that it has the minimum amount of states and actions needed to complete the task. These actions are a set of seven predetermined movements, and the states are a set of four binary conditions, visualized in Figure 3. The states describe the information about the human team member that is needed for the robot to select its actions. In other words, the states provide the robot with observability (**R5**). Furthermore, the handover task is broken down into three distinct phases. In each phase, the robot has different capabilities (states and actions) as shown in Figure 3. The three phases and their corresponding states and actions are described next.

The first phase describes the start of the handover. Here, the robot needs to learn when to start handing over the object. During this phase, the robot can only observe whether the hand of the human team member is in the workspace of the robot. Only when the human hand is in the workspace, the human should be ready to receive the object. The robot has two actions to choose from during this phase, it waits until the state changes, or it can move the object towards the human with the action *Go to human*. During this phase, the first hard dependency is created: the robot must start the handover process. It should, however, wait with doing this, until the human is ready to receive the object. If the robot starts the handover sequence too early, the human won't be able to take the object, resulting in a failed episode.

When the robot took the blue action (*Go to human*), it switches to the second phase of the task. This is a short phase, during which, the robot is moving towards the human. While moving, it decides on the orientation it will use to handover the object. The robot will base this decision on the orientation of the human hand only. The robot can choose between two predetermined orientations: it can provide the object with the palm of the robot hand facing up (*Serve*), so that the human can take it out, or it can do it with the palm facing down (*Drop*), to drop the object into the human hand. This phase allows for a soft dependency to arise, as

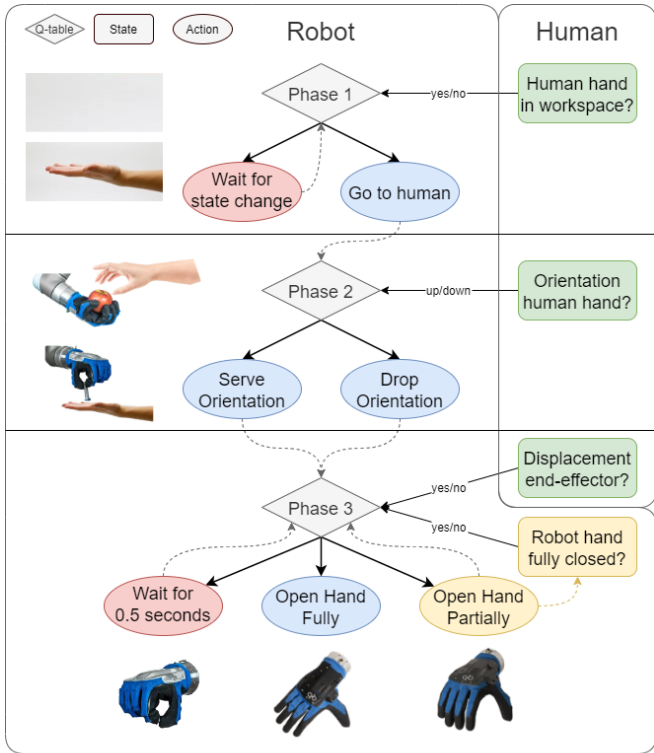


Fig. 3: A flow diagram that shows the capabilities of the robot throughout its three phases. These capabilities include its binary states (shown as rectangles) and its actions (shown as ellipsoids). Actions are red when they do not affect the environment, and blue when they result in the robot advancing its phase. The yellow action influences the yellow state, as shown with the yellow dotted arrow.

this orientation is not crucial for a successful handover, but it might make the collaboration more smooth. Furthermore, either the human can adapt to whichever orientation the robot prefers due to prior coincidence, or the robot can learn how to adapt to the orientation of the human hand, for a smooth handover. This creates an opportunity for predictability or directability [9], [13] for either team member, accommodating the interdependence in the team, as explained in subsection II-A.

After either action in phase 2, the robot will move on to phase 3. In this final phase of the task, the focus is on the moment of handover itself. The robot needs to learn when and how far to open its hand, while the human needs to grasp or catch the object to prevent it from falling. The human can influence the robot's behavior by pulling on the object and displacing the end-effector. Furthermore, an additional state describes whether the robot has its hand still fully closed or if it already opened its hand partially. This combination of capabilities presents opportunities for multiple strategies and soft dependencies to emerge. For example, the robot can learn to wait until the end-effector is displaced before opening its hand to ensure that the human is already holding the object when the robot lets go. Alternatively, the robot can learn to open its hand enough to allow the human to take the object out without dropping it. A third interaction

pattern that can emerge, is that the robot learns to open its hand first partially, before it fully lets go of the object. That way, it can communicate that it is about to open its hand fully, directing the human to catch the object.

Additionally, the robot always starts each episode in the initial state and a final state. The initial state describes that no state changes have been detected, in other words, nothing has happened in yet. It ensures a steep learning pace at the beginning of the learning process, as it makes gives the robot the ability to learn that all actions except for *Wait for state change* should not be taken when no changes are observed in the environment. This initial state is observable though all the phases, to help the robot make the connection between failure at the end caused by errors made at the beginning. The final state describes whether the handover was successful or not.

2) *Secondary task (Human)*: A secondary task was introduced for the human to bridge the gap between the human and robot capabilities, and to create a reason to get an object handed over in the first place. The secondary task is an engaging game-like task, that can only be completed if the human has received the object in time. This secondary task fulfills three functions:

Firstly, the secondary task has to give the human incentive to complete the task. In other words, it should reward the human for the collaborative performance, to ensure a shared goal (**R3**).

Secondly, the secondary task creates a motive for the human to get the object handed over from the robot. In other words, the human must not be able to get the object themselves, so they need the robot to give it to them. This completes the hard dependency (**R1**), that kick-starts the cascading effect of soft dependencies described in subsection II-A.1.

Lastly, the secondary task should compensate for the superior observability of the human, to prevent an observability advantage (**R5**).

The human continuously needs one hand for the secondary task to has to track an asteroid on a screen. To do this correctly, they also can not look away from the screen. The human is incited to deflect this asteroid to complete the game, for this to be possible, however, they need a physical object that serves as a projectile. They cannot get up and get the object themselves, as they need to keep tracking the asteroid on the screen as well to not fail the task. During the first part of the game, the human needs to keep its second hand on a button, until a loading bar is filled (see Figure 4a). As soon as this bar is full, the human can let go of the button, and the human hears a timer starts to tick down. This timer is visualized as a red bar, that slowly decreases until it is empty (see Figure 4b). In this part of the game, the human has 20 seconds to receive the object from the robot, and it is still tracking the target on the screen to compensate the human's observability. When the team succeeds, the human gets rewarded with the same score as the robot, which is the amount of seconds left, +10 for success (see Figure 4c). Otherwise, the human will, like the robot,



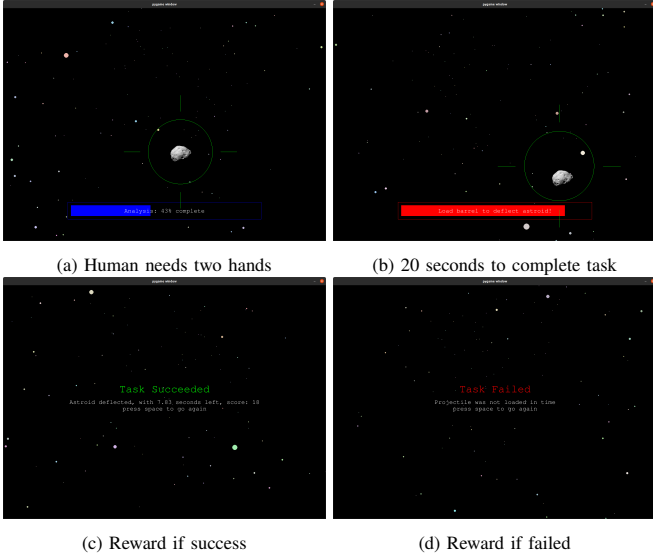


Fig. 4: A storyboard of the visualization of the secondary task

be rewarded negatively (see Figure 4d). Auditory feedback is provided, in addition to the reward screens, to engage the human even more. The reward function used to reward the robot is explained in subsection III-C.2.

### C. Robot RL Algorithm

After an extensive comparison of multiple RL algorithms [2], [27], [28], and their suitability for embodied human-robot co-learning applications in prior literature research, we chose to extend and adapt a Q-learning algorithm [21], because Q-learning is a robust RL technique, it is often used in the domain of social robotics [29], [30], [31], and specifically co-learning [10]. Furthermore, Q-learning can easily be adapted and fitted to the task at hand. The Q-learning based algorithm is adapted using decomposition techniques based on MAXQ value decomposition [32] and extended with eligibility traces [33] to specifically meet the design requirements from subsection III-A. In this subsection, it is explained how the algorithm that enables the robot to learn works, and what design choices were made to make sure the design requirements are met.

1) *Decomposition*: There are more ways to decrease the amount of Q-values, without decreasing the amount of states or actions. The hierarchical RL algorithm MAXQ value decomposition [34], for instance, uses value decomposition to decompose the learning problem into multiple smaller problems with a hierarchical structure, resulting in faster learning [35]. Furthermore, splitting the problem into smaller problems can also increase adaptability [32], as the policy of one phase of the learning problem can change without affecting the policies of other phases.

The three phases in our task (see Figure 3) are sequential instead of hierarchical, meaning they can not be decomposed using Diettrich’s [34] hierarchical value decomposition. The idea, of decomposing the problem, is based on the concept that not every state variable is important during every phase

of the task. As explained in subsection III-B.1, this concept is very applicable in our sequential problem. So, instead of using MAXQ value decomposition, we decomposed the learning problem into three sequential Q-learning problems, each with its own Q-table, creating the same effect of decreasing the amount of Q-values without affecting the amount of actions and state variables. In other words, we base the Q-values not only on state ( $s$ ) and action ( $a$ ) alone, but also on phase ( $\phi$ ). Which mathematically looks like this:

$$Q_*(\phi, s, a) = E[R(\phi', s') + \gamma \max_{a'} Q_*(\phi', s', a')] \quad (2)$$

In our case, for instance, we decomposed the start of the task into two steps. First, the robot has to decide when to start the passing process, then it chooses in what orientation it should provide the object to the human. The robot could make this first decision based only on whether the human is ready to receive the object. While in this second phase, the robot could base the orientation at which it provides the object, only on the orientation of the human’s hand. In table Table I and Table II the shape of the Q-tables, and the amount of Q-values in both scenarios are displayed. In these scenarios, the robot can observe two binary states as explained in subsection III-B.1: it can measure whether the human hand is in the workspace, and if the human hand palm is facing predominantly up or down. Based on these states, the robot can effectively choose from three actions: it can wait and do nothing until the state changes, or it can provide an object near the human either in the *Serve* orientation or in the *Drop* orientation. In the first scenario, where the two phases are not decomposed, the state-action space is covered by 12 Q-values as shown in Table I. In the second scenario, the robot first decides whether it should wait, or go towards the human, effectively starting the passing process. During this movement, it can immediately decide in what orientation it will provide the object, based only on the orientation of the human hand. So, as shown in Table II, even though an extra action was added, this decomposition reduced the amount of Q-values to be learned from 12 to 8.

TABLE I: Size of Q-table without decomposition of phase 1 and 2

Phase 1 + 2	In WS Palm up	In WS Palm down	Not in WS Palm up	Not in WS Palm down
Wait	$Q_1$	$Q_2$	$Q_3$	$Q_4$
Orient. A	$Q_5$	$Q_6$	$Q_7$	$Q_8$
Orient. B	$Q_9$	$Q_{10}$	$Q_{11}$	$Q_{12}$

TABLE II: Size of Q-tables with decomposition of phase 1 and 2

Phase 1	In WS	Not in WS
Wait	$Q_1$	$Q_2$
Go	$Q_3$	$Q_4$

Phase 2	Palm up	Palm down
Ori. A	$Q_5$	$Q_6$
Ori. B	$Q_7$	$Q_8$

In other words, by decomposing the task, we provide the robot with some information about which state variables are



important during each phase of the task. Without this decomposition, the amount of Q-values would be 112, as there are four binary states and seven actions (see Figure 3) and all combinations needed to be accounted for. The decomposition reduces this number to 20. This significantly decreases the scale of the learning problem, increasing the overall learning pace and adaptability of the RL agent, consolidating the method to meet **R2** and **R4**.

2) *Reward function*: Design requirement **R3** states that both agents get rewarded based on the performance of the task and that both agents get rewarded similarly to ensure they have the same goal. In the task of our method, this goal is successfully completing the handover task without dropping it. So, both agents receive either positive or negative feedback at the end of the episode. This feedback is based on whether the task was completed successfully as well as the time that was left to do so. As explained in subsection III-B.2, this time limit prevented the possibility for the team to do nothing to avoid failure. So, to mimic this in the reward function of our RL algorithm, it receives a positive reward (+10) when the task is completed successfully, and negative (-10) when the task fails. Additionally, when the task succeeded, the amount of seconds left to complete the task, was added to the positive reward. As the team was given 20 seconds to do so, the positive reward would always be between +10 and +30. To accommodate **R3** even more, the human would see this same reward as a score given for the completion of the task. In Q-learning, however, the Q-values get updated after each action, and not just when the episode is completed. So, this function is extended with a small punishment for each action. This prevents a policy where the robot gets stuck in a loop, taking the same action over and over again.

3) *Eligibility traces*: Rewarding our Q-learning algorithm only at the end of each episode, however, creates two problems that are both solved with eligibility traces [33], which are described below.

The first problem is most actions will get a delayed reward [36]. This means that because only the last state-action pair before completion will get rewarded, only this Q-value is associated with that reward. It would then take a new episode, to reach the same final state, for the second-to-last Q-value to get updated accordingly. This is because each Q-value gets updated based on the reward received after the corresponding action and the maximum Q-value of the reached state, the cumulative future reward, as explained in subsection II-B and shown in (1). This is a problem for both **R2** and **R4**, as it takes multiple episodes to learn the connection between the received reward and earlier state-action pairs.

The second problem is that this cumulative reward is calculated as the maximum Q-value of the reached state. This causes a problem, because we decomposed the learning problem, and a reached state is not necessarily influenced by the previous state-action pair, when the last action causes the agent to go to the next phase. For instance, when the robot takes the action *Go to human* in phase 1, it goes to phase

2 (see Figure 3). Now, the reached state only describes the orientation of the human hand, and tells the robot nothing about whether the human is in the workspace. So, when this action was wrongfully taken, when the human is not in the workspace, this state-action pair can still be rewarded positively, because any of the actions in the reached state could have a great Q-value due to success in an earlier episode.

Using eligibility traces [33], the algorithm keeps track of all state-action pairs reached during the episode. At the end of each episode it additionally updates all the corresponding Q-values based on the reward received. This does not only speed up the learning process, but it also makes sure that mistakes made in early phases of the task also get rewarded negatively in case of an unsuccessful episode [37].

An eligibility trace is a trace of all the previously visited Q-values. These traces are stored in a table for each state-action pair in each phase  $S(\phi, s, a)$ . All values for the eligibility initiate as zero at the beginning of each episode. Every time an action is taken, the corresponding value for the eligibility of that phase-state-action combination is set to 1. Then, all other values are reduced by the discount factor  $\gamma$  (explain in subsection II-B) and the eligibility factor  $\lambda$ , that determines how much previously visited Q-values should be updated with respect to the last Q-value:

$$S(\phi, s, a) = \gamma\lambda S(\phi, s, a) \quad \forall S(\phi, s, a) \quad (3)$$

In Q-learning without eligibility traces only the last Q-value is updated after a taken action. With eligibility-traces however, all Q-values are updated after every action, based on the eligibility  $S(\phi, s, a)$ . To do so, we first calculate what would have been the updated Q-value for the last phase-state-action combination  $\hat{Q}(\phi, s, a)$  shown in (4a), using to the decomposed Bellmann equation (2). Then we use  $\hat{Q}$  to calculate the update-value  $\Delta_Q$  (4b):

$$\hat{Q}(\phi, s, a) = R(\phi', s') + \gamma \max_{a'} Q(\phi, s', a') \quad (4a)$$

$$\Delta_Q = Q(\phi, s, a) - \hat{Q}(\phi, s, a) \quad (4b)$$

This update-value ( $\Delta_Q$ ) is then used to update all Q-values based on their eligibility. As shown in (5):

$$Q(\phi, s, a) = Q(\phi, s, a) + \alpha \Delta_Q S(\phi, s, a) \quad \forall S(\phi, s, a) \quad (5)$$

The learning rate  $\alpha$  is a value between 1 and 0 and it is used in the update equation, to determine to what extent new experiences override what has been learned all ready.

4) *Epsilon decay*: The algorithm uses epsilon decay to address the balance between exploration and exploitation, or greediness. When the robot explores, it picks a random action, while when exploiting, it takes the action that corresponds to the highest Q-value as explained in subsection II-B.

A parameter  $\epsilon$  portrays the chance that the agent explores. By starting with a high  $\epsilon$ , the algorithm explores fast at first, after which a lower  $\epsilon$  lets the algorithm explore more around the higher Q-values. Normally, when the optimal

policy is found,  $\epsilon$  could decay all the way to zero, so the algorithm would only exploit its learned policy. When the algorithm should however stay adaptable during all stages of the learning process (**R4**), the system can never stop exploring. Therefore,  $\epsilon$  should never decay all the way to zero.

It was iteratively found during multiple pilots that 20% changes of exploration was low enough to stay adaptable, while it also ensured predictable behavior. Note that because the robot always has only two or three actions to choose from, it would still have a significant probability of taking the action with the highest Q-values when picking an action at random.

Alternatively to  $\epsilon$ -decay, there are other strategies to determine the greediness of the algorithm, such as the Boltzmann strategy [38], or the frequency maximum Q-value (FMQ) heuristic [39]. These strategies can outperform traditional  $\epsilon$ -decay on multiple aspects, as is shown by Kapetanakis et al. [39]. However, all these alternatives are designed to converge to a greedy policy after a while, which is not beneficial for the adaptability in later stages of the learning process. Rendering them unsuitable for our algorithm.

$$\epsilon = \begin{cases} \max(\gamma_\epsilon \epsilon, 0.2) & \text{if } R < 0 \vee \epsilon > 0.5 \\ \min(\frac{1}{\gamma_\epsilon} \epsilon, 0.5) & \text{if } R > 0 \end{cases} \quad (6)$$

In (6) it is shown how are epsilon changes over the episodes.  $R$  in these equations represents the reward at the end of an episode that is either negative or positive and  $\gamma_\epsilon$  represent the epsilon decay rate. Epsilon starts at a value of 1, to guaranty exploration when no policy is learned yet. Epsilon then slowly decays during the first seven episodes until it reaches a 50% change of exploration ( $\gamma_\epsilon = 0.9$ ). Then, during the rest of the episodes, epsilon is changes in such a way, that when the team has a high success rate, the robot has a higher chance to exploit its current policy. While when the team experiences more failure, the chance of exploring grows.

After the initial decay  $\epsilon$  changes in such a way, that when the team has a high success rate, the robot has a higher chance to exploit its current policy. While when the team experiences more failure, the chance of exploring grows. (6) ensures that  $0.2 \leq \epsilon \leq 0.5$ . The exploration rate was capped at 50% to ensure predictable behavior, even when multiple episodes failed in a row.

#### IV. EXPERIMENTAL DESIGN

We conducted an experiment to test whether the newly developed meets the design requirements. Furthermore, our aim was to gain new insights about their effects on the team interdependence and co-learning. We tested our method by trying to establish co-learning in six human-robot teams. To learn how these different aspects of co-learning are related to the design requirements, multiple scenarios should be compared, as the development of different aspects of co-learning can be different for every human-robot team. In other words, we tested our method of co-learning in multiple

human-robot teams, to learn how interdependence and other aspects of co-learning are achieved as an effect of the composed design requirements.

##### A. Setup

The chosen hardware setup (shown in Figure 1) consists of the *KUKA iiwa7 LBR800* robotic arm and the *qb-softhand*, which fostered a safe environment for the human participants to engage in co-learning with the robot. This setup enabled collaborative capabilities, such as the ability to measure physical interactions, and force limitation for safety.

The *KUKA iiwa7 LBR800* robotic arm was chosen as it is designed to work alongside humans. It incorporates torque sensors in each joint, allowing the robot to perceive external forces, ensuring a safe and responsive collaboration during physical interactions between the human and the robot.

The *qb-softhand* is attached to the robotic arm, providing a versatile grip. This hand allowed the robot to grasp and manipulate objects effectively during the experiment.

To provide observability to the robot and measure human actions, the *OptiTrack* motion tracking system was employed. We attached reflective markers to a glove that is worn by the human, which were tracked online by multiple cameras. Specialized software uses triangulation to calculate the positions of these markers, and thus the pose of the hand of the human participants, in 3D space in real-time. This pose is then used to determine the state and actions of the human team member, as can be seen in Figure 3.

Furthermore, the human uses a mouse and a keyboard to perform the secondary task as explained in subsection III-B.2 and is provided visual and auditory feedback from a PC.

The separate hardware systems, are each monitored and controlled by separate scripts that run asynchronously on multiple machines. A main script, running the learning algorithm, controls these separate systems, using the publish-subscribe model of ROS, integrating the whole system together as one RL agent.

##### B. Experiment

Before the experiment, we first explained to each participant the goal of the task and the secondary task. Next, we showed them how the robot moves. Each participant got the chance to interact physically with the robot to get familiar with the setup. This was done, to show that the chosen hardware was safe to interact with, and to make the human feel safe and at ease collaborating in the same workspace as a strong robotic arm. During this familiarization, however, the action space of the robot was purposely not shown, as this was something that the human should learn during the co-learning. Next, when the participant was asked to wear the sensorized glove, it was shown that the cameras are used to observe this hand, but similar to the action space, nothing about how this observable pose was annotated as the state-space of the robot was explained beforehand. In other words, the human would have to learn during the experiment, what movements of their hand could be used to communicate with the robot.

Once the participant was familiar with the setup and understood the assignment, the experiment started. The human-robot team were allowed to learn the task in four sets of 10 minutes. During the experiment we collected data such as the task performance of each episode and the development of the Q-values of the robot. After each set, the participants filled in a questionnaire on human-robot fluency [40] to capture their perception of performance and interdependence, and how it changed over time. The whole experiment was also recorded on video to be able to analyze certain events or interaction patterns that occurred during the experiment. At the end of the four sets the human was interviewed to qualitatively capture their goal, strategies and understanding of the behavior of the robot. The interview was also held to better understand what was learned by the human and to apprehend how certain aspects of the developed method contributed to the interdependence of the team.

### C. Metrics

To understand what happened during the experiment, and to be able to visualize this data, we created six metrics: *Performance rate*, *Human perception*, *Strategies*, *Relative liability*, *Action preference* and *The Interview*. Each of them is explained in the following subsections. Note there are infinite many ways to co-learn, and that these metrics are meant to be evaluated qualitatively. The results will have to be interpreted and discussed case-by-case, in order to identify whether co-learning, and interdependence was present in the team.

1) *Performance rate*: The first metric is introduced to describe the collaborative performance of the team over time. This metric describes the success rate of the team during each 10-minute session. It is expressed as the percentage of successful episodes with respect to the total amount of episodes during that session:

$$\text{Performance rate} = \frac{\text{Amount of successful episodes}}{\text{Total amount of episodes}} 100\% \quad (7)$$

This metric is used in Figure 6.

2) *Perception of human-robot fluency*: To describe how the human experiences collaboration and performance, a questionnaire on human-robot fluency [40] was filled in by the participants after each 10-minute session. This questionnaire contained 15 questions related to the following six categories:

- 1) Collaboration Fluency
- 2) Relative Contribution
- 3) Trust in the Robot
- 4) Positive Teammate Traits
- 5) Perception of Improvement
- 6) Perception of Shared Goal

Every question posed a statement, for instance: "The human-robot team improved over time", on which the participants used a seven-point Likert scale to indicate whether they felt the statement was true. The scale went from *strongly disagree* (1) to *strongly agree* (7). The overall metric for

the perception of human-robot fluency was obtained by an average of these six sub-metrics. The complete questionnaire is shown in appendix C. This metric is used in Figure 6.

3) *Strategies*: As described in section III the method is designed such that multiple strategies could lead to success and to encourage the emergence of soft dependencies. To demonstrate this ability (**R1**) as well as the adaptability of the robot (**R4**), we distinguish three strategies that can be used by the team to determine the exact moment where the object is transferred:

- S1 The robot lets go of the object, trusting the human will catch it.
- S2 The human pulls on the object, letting the robot know it can let go.
- S3 The robot opens its hand partially, letting the human take the object.

These strategies are identified, based on the state-action combinations that the robot was provided in phase 3 (see Figure 3)

In the first strategy (S1), the robot has no conformation, that the human is ready to grasp the object. The robot just opens its hand and depends on the human to catch the object. In the second strategy (S2), the robot first confirms that the human is holding the object before it releases the object. It does this by measuring the displacement of the end effector, so the human has to exert some force on the object to let the robot know it can release the object. In the last strategy (S3), the robot does never open its hand completely, as it has learned that when it opens its hand partially the participant will take the object out when they are ready.

In each successful handover, exactly one of these three strategies must be chosen, as there are no other ways to transfer the item. We distinguish this for each successful handover using algorithm 1. In Figure 5 each successful handover is allocated to one of the three strategies. The strategy can be determined based on the last state-action pair of each successful episode:

---

#### Algorithm 1 Distinguish strategy in phase 3

---

```

for episode,  $i \in \text{Successful Episodes}$  do
  if last action $i$   $\neq$  Open Hand Fully then
    | strategy $i$   $\leftarrow$  S3
  else
    | if last state $i$  = End-effector displaced then
      | | strategy $i$   $\leftarrow$  S2
    | else
      | | strategy $i$   $\leftarrow$  S1
    | end
  end
end

```

---

4) *Relative liability*: This metric describes the proportion in which the team members caused the episodes to fail in each 10-minute session. This metric visualizes the relative learning pace of both agents, since when the learning pace

is similar, this proportion should stay the same over time. If an agent learns faster than their team member, there is a shift in relative liability because the proportion of mistakes made by the superior agent goes down.

The relative liability is defined, by determining for each failed episode which agent made the mistake that caused the episode to fail. This is represented as a percentage of the total amount of failed episodes in that 10-minute session. This metric is used in Figure 7.

The robot is responsible for a failed episode when the object was not passed within the allocated time, while the human does try to signal the robot. Or when the robot dropped the object without the human touching it. For other reasons of failure, the human is said to be liable. These reasons include mistakes made in the secondary task.

5) *Action preference from Q-values*: This metric describes the specific policy of the robot in phase 2 of the task (Figure 3). In this phase the robot can measure two possible states: *Palm up*, *Palm down*, that describe the orientation of the human hand. Based on the state, the robot can choose between two actions that determine the orientation in which it provides the object: *Drop* and *Serve*. This means, there are four possible state-action pairs, as visualized in Table III.

TABLE III: Q-table in phase 2

Phase 2	Palm up	Palm down
Drop	$Q(s_{up}, a_{drop})$	$Q(s_{down}, a_{drop})$
Serve	$Q(s_{up}, a_{serve})$	$Q(s_{down}, a_{serve})$

This Q-table describes the policy of the robot in phase 2. When  $Q(s_{up}, a_{drop})$  is higher than  $Q(s_{up}, a_{serve})$ , for example, the robot will choose action *Drop* when the state is *Palm up*. To visualize this preference ( $P_s$ ), for each state over time, the difference between the Q-values for the two actions is calculated for each state  $s$ , as shown in (8):

$$P_s = Q(s, a_{drop}) - Q(s, a_{serve}) \quad (8)$$

When  $P_s$  is positive, the robot prefers the action *Drop*, and when it is negative, it will choose *Serve* in state  $s$ . This metric is used to indicate adaptability of the robot in team A in Figure 8. In appendix B the same figure is displayed for the other teams.

6) *Interview*: The last metric is a qualitative interview that was held with each of the participants after the last learning session. In this interview, we ask the three following questions:

- Q1 Please indicate what your objective was during the learning process.
- Q2 Describe the different strategies that you used, and how did this change over time.
- Q3 Did you rely on a specific strategy of the robot?

The first question was asked specifically to investigate whether the goal of the human correspond to the goal of the robot, so it could be indicated whether the team had a shared goal **R3**.

The aim of the second question was to find if the human explored different strategies during the learning process, and more specifically whether it converged towards preferring one strategy over other strategies.

Using the last question, we aimed to find whether the human experienced soft dependencies, and therefore allowed for interdependence to grow as explained in subsection II-A.1.

After each question, follow-up questions are asked to start a conversation on the topic. This helped to create a more complete answer to each of the questions.

## V. RESULTS AND INTERPRETATION

In this section, we present the results of the experiment by analyzing whether each design requirement is met for each human-robot team. This is summarized in Table IV. Each subsection represents a requirement

Note, that we take a qualitative instead of a statistical approach. The motivation for this choice is that co-learning is an open-ended process where the same aspects can manifest in many different ways, due to the multiple possible strategies that can be taken by the team. Therefore, case-by-case qualitative analysis is required to identify and study these particularities, which would otherwise not be visible in statistical analysis.

TABLE IV: Overview of each requirement and whether it was met during the experiment in each team. Additionally, the bottom row shows whether the results indicate that co-learning took place during the experiment. The content of the bottom row is discussed in section VI.

Teams	A	B	C	D	E	F
<b>R1</b> - Dependencies	✓	✓	✓	✓	-	✓
<b>R2</b> - Learning pace	✓	✓	-	X	✓	✓
<b>R3</b> - Shared Goal	X	✓	✓	-	✓	✓
<b>R4</b> - Adaptability	✓	✓	-	✓	-	✓
<b>R5</b> - Observability	✓	✓	X	-	✓	✓
Co-learning	✓	✓	-	-	-	✓

### A. Dependencies (**R1**)

Different individuals prefer different strategies. Figure 5 shows that the method enables different teams to learn different strategies. Team A, for instance, converges completely to strategy S1, while teams B and D learned that this strategy did not work for them.

Post-experiment interviews revealed some important underlying insights about the development of the strategies during co-learning. Participants A and C stated that they did not want to take the object from the robot without its permission in an attempt to maintain the trust of the robot. This complies with the quantitative data that shows that strategy S3 was not preferred in these teams. By actively not choosing this strategy, the human depends on the robot to open its hand completely for the task to be completed. This clearly shows an establishment of a soft dependency between the human and the robot that is beneficial for the relationship of the team.

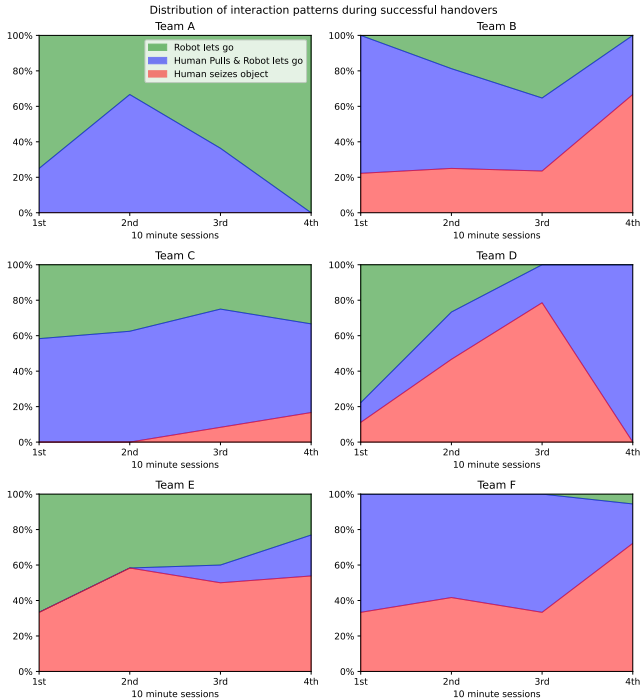


Fig. 5: The distribution of the three different strategies in phase 3 that could lead to a successful handover. This figure shows how the preference for these strategies changes over time. The three interaction patterns are as follows: **S1**: The robot lets go of the object, trusting the human will catch it. **S2**: The human pulls on the object, letting the robot know it can let go. **S3**: The robot opens its hand partially, letting the human seize the object.

Similarly, all three strategies, are paired with soft dependencies. This means that soft dependencies arise during the learning process, when a team converges to preferring one specific strategy. It can be seen in Figure 5 that in all teams multiple strategies were explored. In teams A, B, D and F, it is clear that there was convergence to one specific strategy during the experiment. Thus, soft dependencies emerged in these teams.

Team E is the only team which kept executing all strategies until the end of the experiment. So both team members never fully committed to being completely dependent on the other one. Making it the only team where it is inconclusive whether design requirement item **R1** is met.

### B. Learning pace (**R2**)

This design requirement prescribes that the robot should be able to learn at a similar pace as the human. In Figure 7 it can be seen that in all teams except teams C and D, the ratio of who was responsible for episodes to fail is constant over time. This means that in these teams, the learning pace of the human and the robot are similar. If one agent would, for instance, learn faster than the other agent, we should see a shift in this proportion over time, until the other agent is responsible for almost all the mistakes.

This happened in team D during the first 3 sessions, where the robot could not keep up with the learning pace of the human. Hence, Table IV shows that this requirement was not met in team D. Moreover, the learning pace of the human

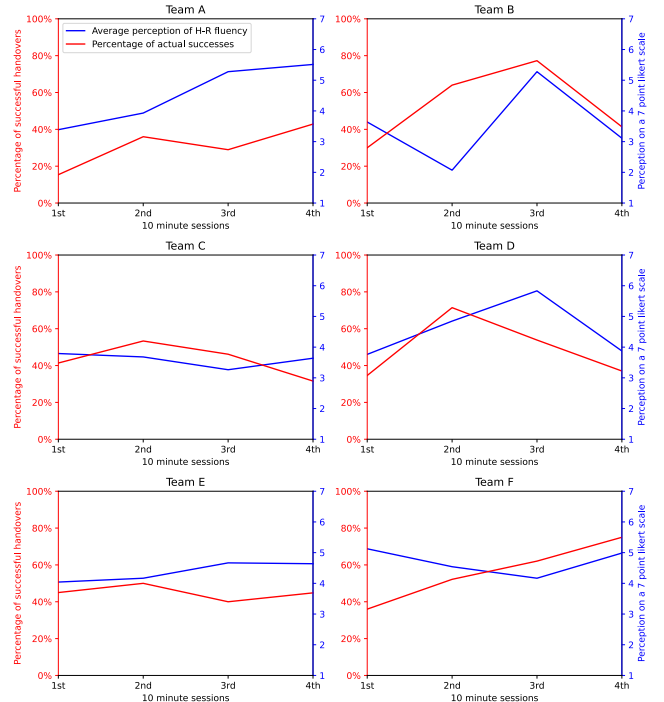


Fig. 6: Overview of co-learning and performance. The red line shows the collaborative performance of each team, which is objectively measured as the success rate during each 10-minute session. The blue line shows the team fluency perceived by the human, which is measured with a questionnaire [40] after each session.

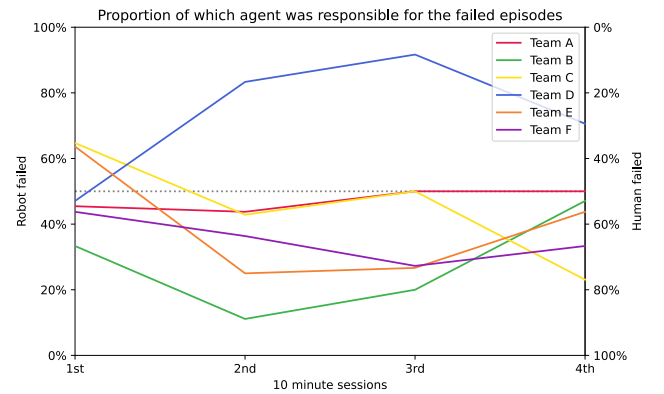


Fig. 7: The percentage of how many times each agent was responsible for failing an episode during each 10-minute session is shown for each team. The rate of failure of the robot can be read on the left axis, while the human failure rate is displayed on the right axis

was still faster than the robot during the fourth session, even though Figure 7 implies that the robot made a recovery. The reason this proportion drops back towards 50%, however, is that the human started to deliberately fail the task to actively train the robot to prefer strategy **S2** (see Figure 5), as they mentioned in the interview. This can also be seen by the sudden decrease in performance rate during this session in Figure 6.

In team C, a significant shift in the proportion of liability can be seen in Figure 7. The robot improved its policy, faster than the human did. When we combine this information with

the fact that the team barely improved their performance during the four sessions (see Figure 6), we can deduce that the human did not improve its policy at all. Therefore, no conclusion could be drawn about this requirement for this team.

### C. Shared Goal (R3)

This design requirement is met if both agents share the same goal. As the robot gets rewarded based on collaborative performance only, its mere goal, is to improve this performance. One of the interview questions was specifically targeted to identifying what the goal of the human was. Participants B, C, E and F indicated that their goal was to complete each episode without dropping the object. Participants B and F even indicated that they had a secondary goal of improving the time in which they succeeded, to optimize their score. Therefore, it is shown in Table IV that requirement **R3** is met in these teams.

Participants A and D, on the other hand, indicated that their main goal was not necessarily to succeed at the task but mainly to train the robot to follow their preferred strategy.

Participant A said that this was their main objective during the whole experiment. For instance, they never let the task succeed if the robot did not let go of the object. This is why strategy S3 is never seen in team A in Figure 5. Additionally, in Figure 6 it can be seen that the human perception is constantly higher than the team’s actual performance. This can be explained by the fact that the human met their objective of influencing the robot’s behavior, at the expense of the robot’s goal of succeeding at the task. Resulting in both agents perceiving different rewards.

Participant D, on the other hand, indicated during the interview that they changed their objective between the third and fourth session. First it matched the objective of the robot, while during the last session their goal was only to train the robot to their preferred strategy. Their goal changed, as participant D explained in the interview, because when they realized that the robot used trail-and-error learning, they knew they could influence the robot’s behavior by consequently rewarding desired behavior and punishing undesired behavior. At this moment, this participant suddenly changed their behavior, resulting in the performance drop in the 4th session (Figure 6) and the human deliberately failing the task to train the robot (Figure 7). While this participant started with the same goal as the robot, i.e., meeting requirement **R3**, they changed their goal over time, resulting in the requirement first being met and then not met, leaving **R3** inconclusive in team D.

### D. Adaptability (R4)

Requirement **R4** is about the ability of the robot to adapt its policy in all later stages of the learning process. In Figure 5, it can clearly be seen that teams B, D and F made a change in preferred strategy between the last two sessions of the experiment. This shows that the RL algorithm was able to adapt its policy as well to accommodate this switch in strategy.

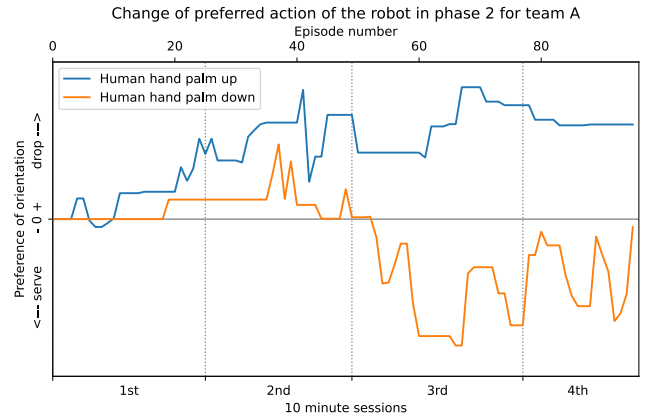


Fig. 8: The difference between the two Q-values for the actions *Drop* and *Serve* is shown for team A. The value of this difference is plotted over the episodes for the two states the robot can observe in phase 2. The two actions determine the orientation in which the robot will convey the object, and the states display the predominant orientation of the human hand, as explained in subsection III-B.1. The difference between those two Q-values shows which action is preferred during which state, as explained in subsection IV-C.5 and (8). When this metric is negative, the robot learned to choose the action *Serve* in that state, while above the gray null line, it prefers the *Drop* orientation.

Adaptability of the robot in team A, is shown in Figure 8. This figure is about phase 2 from Figure 3. The preference  $P_s$  is positive for both states during the first two sessions of the experiment. As explained in subsection IV-C.5 and (8), this indicates that the robot has the strategy of dropping the object in the hand of the human regardless of the orientation of the human hand. However, just after the start of the third session, this preference changed for the state where the hand palm of the human is facing down (the orange state). This shows that the requirement **R4** was present in team A, as the robot changed its policy during a later stadium of the learning process.

In teams C and E, no specific adaption of the policy of the robot occurred during the experiment. However, this does not necessarily mean that the robot had no adaptability. That adaptability did not show in this experiment, does not mean that the robot is incapable of adapting to changing behavior of the human. Therefore, these cells are left inconclusive in Table IV.

### E. Observability (R5)

The method allows both agents to observe each other by design: The robot has multiple states that describe the behavior of the human, as explained in subsection III-B.1 and Figure 3. A human naturally has the ability to observe its environment, including the physical robot. Therefore, the method enables observability for both agents by design. An additional part of **R5** is that both agents can observe each other to a similar extent in order to prevent an imbalance in the learning pace.

Figure 7 shows that there was no imbalance in learning pace in teams A, B, E and F, as explained in subsection V-B. The unequal learning pace in team C, however, was



caused by the fact that the human was not able to learn the policy of the robot. This was a result of the human being too occupied by the secondary task, resulting in non-similar observability. There is, on the other hand, no evidence that the unequal learning pace in team D was caused by non-similar observability.

While the secondary task prevented visual observability, as explained in subsection III-B.2, Figure 5 shows that in teams B, D and F, the human preferred to rely on tactile sensing to know where to grasp the object, as they do not follow S1. Further investigation of the video recordings of the experiment showed that participants A and E also relied on tactile sensing to locate the object, they just did it subtle enough to not displace the robot.

In short, in teams A, B, E, and F we can state that both agents had observability, and that no unwanted imbalance was caused. This means the requirement is met (see Table IV). In team C the secondary task overcompensated the observability of the human, causing this requirement not to be met, while in team D the results are inconclusive.

## VI. DISCUSSION

### A. Identifying Co-learning

As explained in section I and shown in Figure 2 joint activity can be achieved in multiple different ways and co-learning is an open-ended process with multiple possible ways that can result in similar outcomes. Therefore, in section V, we did a quantitative case-by-case analysis on the results and development of strategies in each of the six human-robot teams to identify each of five design requirements.

As explained in subsection III-A, the design requirements are not only required to be met for co-learning to occur, but they also represent the five aspects of co-learning. So, when it can be shown that a team meets all the requirements, it can be stated that the team was co-learning during the experiment. Table IV shows that all the requirements are met in team B and team F. Therefore, the results show that interdependence and co-learning was present in these teams.

In team A, even though the human and the robot did not have the same goal (**R3**), they still co-learned to improve their collaboration and formed an interdependent relationship in the process. Even though the requirement is strictly not met, the aim of the requirement is still realized: As explained in subsection III-A, having the same goal is important for co-learning and the development of joint activity, because in order to achieve joint activity, both agents must resort to compatible strategies (as explained in Figure 2). The reason **R3** was not met in team A, is that the goal of the human was to train the robot, while the goal of the robot was to succeed at the task. In practice, however, these goals overlap enough that there are still multiple strategies congruent that reach both goals. Moreover, Figure 6 shows an increase in performance over time, as well as a growth in the participant's perception of the fluency in the team. These effects are both the result of the emergent of soft dependencies, and development of joint activity that can be interpreted from Figure 5 and Figure 8.

In team C, the human struggled to understand how to do the task, and was not able to learn this within the given time. Even though, the team was able to develop some interaction patterns, and soft dependencies (**R1**). This still resulted in multiple requirements that were not met. Making it inconclusive whether this team was able to co-learn.

In team D, the human learned much faster than the robot, which led to an imbalance in contribution over time (Figure 7). Because of this, the human changed its motivation over time. So even though co-learning might have been present during the first sessions of the experiment, it did not sustain during the last session. Thus, in team D multiple design requirements were left inconclusive or were not met. So we can not show that that interdependent co-learning was present.

In Team E, we were not able to show that soft-dependencies emerged during the experiment. Additionally, Figure 6 does not show an increase in performance or perception. Therefore, co-learning was not identified here. However, changes in preferred interaction patterns over time, can still be observed in Figure 5, even if they are not substantial enough to prove that **R1** or **R4** were met, it does not mean they are not met. Furthermore, Figure 7 shows a balanced learning pace between the two agents. From this we can conclude that the team was still learning after the four sessions, and co-learning might have happened, while the effects are not yet measurable.

In short, in three out of six teams, interdependent co-learning could be identified by the results.

### B. Limitations, and Future work

We can show that our developed method enables co-learning. Figure 6 does however not show a significant increase in performance for most teams. This can be explained by the fact that the method is designed with the focus on co-learning. The essence of co-learning is improving the collaboration by creating an interdependence relationship in the team. Improvement of performance is a result of this. This means that co-learning can be present without an immediate performance increase. We expect that when the same experiment is done for a longer duration of time, such an increase in performance should be measurable in the teams where co-learning is identified. Therefore, in future work, we will focus more on the long term effects of co-learning in embodied human-robot teams.

Secondly, this method focuses mainly on enabling human-robot co-learning by meeting theorized design aspects, as it is the first step of conceptualizing embodied human-robot co-learning. This means that, the individual effect and necessity of each design requirement can not separately be shown with this research. In team A we were for instance able to identify co-learning despite **R3** not being met, this suggests that this design requirement might be less fundamental than the other design requirements. Thus, quantitatively researching the importance and impact of each of the design requirements, is one of the next steps in understanding how co-learning can be achieved.

Lastly, in the development of this method, we focused mainly on creating an algorithm that allows a physically embodied robot to co-learn, as humans already have inherent capabilities that allow them to co-learn. However, in order to better understand how co-learning between a human and a robot can be achieved, it can be beneficial to explore how the experience of the human affects the possibilities of co-learning. For instance, exchanging the object for a fragile wine glass or a heavy weight might increase the motivation of the human to not drop the object. This could have significant effects on the responsibilities and dependencies that emerge in the team, or it could influence the amount that the human explores new strategies. In other words, the immersion and the context of the task can influence human behavior. Therefore, we plan to research how immersion and human experience affects the opportunities for human-robot co-learning in physical environments.

## VII. CONCLUSION

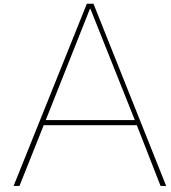
We answered the research question: “How can a human-robot team achieve co-learning, and interdependence in physically embodied tasks?” by successfully developing a method, based on five design requirements that outline the challenges of physical co-learning in human-robot teams. We showed that our method enables co-learning and interdependence between human and robot in at least three out of the six teams that performed the experiment.

To pursue the progress made in this research, future work should be dedicated to quantitatively investigate the impact of each design requirement on co-learning and interdependence.

## REFERENCES

- [1] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, “Progress and prospects of the human-robot collaboration,” *Autonomous Robots*, vol. 42, no. 5, pp. 957–975, 2018.
- [2] N. Akalin and A. Loutfi, “Reinforcement learning approaches in social robotics,” *Sensors*, vol. 21, no. 4, p. 1292, 2021.
- [3] L. Peternel, N. Tsagarakis, D. Caldwell, and A. Ajoudani, “Robot adaptation to human physical fatigue in human-robot co-manipulation,” *Autonomous Robots*, vol. 42, pp. 1011–1021, 2018.
- [4] S. Kana, S. Lakshminarayanan, D. M. Mohan, and D. Campolo, “Impedance controlled human-robot collaborative tooling for edge chamfering and polishing applications,” *Robotics and Computer-Integrated Manufacturing*, vol. 72, p. 102199, 2021.
- [5] J. de Miguel-Fernández, J. Lobo-Prat, E. Prinsen, J. M. Font-Llagunes, and L. Marchal-Crespo, “Control strategies used in lower limb exoskeletons for gait rehabilitation after brain injury: a systematic review and analysis of clinical effectiveness,” *Journal of neuroengineering and rehabilitation*, vol. 20, no. 1, p. 23, 2023.
- [6] G. J. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer, and J. Peters, “Probabilistic movement primitives for coordination of multiple human-robot collaborative tasks,” *Autonomous Robots*, vol. 41, pp. 593–612, 2017.
- [7] H. Loopik and L. Peternel, “A multi-modal control method for a collaborative human-robot building task in off-earth habitat construction.” International Conference on Advanced Robotics (ICAR), 2021, workshop on Design, Learning, and Control for Safe Human-Robot Collaboration.
- [8] C. R. B. Azevedo, K. Raizer, and R. Souza, “A vision for human-machine mutual understanding, trust establishment, and collaboration,” in *2017 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, 2017, pp. 1–3.
- [9] K. van den Bosch, T. Schoonderwoerd, R. Blankendaal, and M. Neerinx, *Six Challenges for Human-AI Co-learning*, ser. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2019, vol. 11597 LNCS, cited By :9. [Online]. Available: [www.scopus.com](http://www.scopus.com)
- [10] E. M. van Zoelen, K. van den Bosch, and M. Neerinx, “Becoming team members: Identifying interaction patterns of mutual adaptation for human-robot co-learning,” *Frontiers in Robotics and AI*, vol. 8, 2021. [Online]. Available: <https://www.frontiersin.org/article/10.3389/frobt.2021.692811>
- [11] E. M. van Zoelen, K. van den Bosch, M. Rauterberg, E. Barakova, and M. Neerinx, “Identifying interaction patterns of tangible co-adaptations in human-robot team behaviors,” *Frontiers in Psychology*, vol. 12, 2021. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fpsyg.2021.645545>
- [12] A. Shafti, J. Tjomsland, W. Dudley, and A. A. Faisal, “Real-world human-robot collaborative reinforcement learning,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 11 161–11 166.
- [13] M. Johnson, J. M. Bradshaw, P. J. Feltoovich, C. M. Jonker, M. B. van Riemsdijk, and M. Sierhuis, “Coactive design: Designing support for interdependence in joint activity,” vol. 3, no. 1, p. 43–69, Feb 2014. [Online]. Available: <https://doi.org/10.5898/JHRI.3.1.Johnson>
- [14] G. Montúfar, K. Ghazi-Zahedi, and N. Ay, “Information theoretically aided reinforcement learning for embodied agents,” *CoRR*, vol. abs/1605.09735, 2016. [Online]. Available: <http://arxiv.org/abs/1605.09735>
- [15] T. Y. Katz-Navon and M. Erez, “When collective- and self-efficacy affect team performance: The role of task interdependence,” *Small Group Research*, vol. 36, no. 4, pp. 437–465, 2005. [Online]. Available: <https://doi.org/10.1177/1046496405275233>
- [16] H. Loopik, “Comparing reinforcement learning techniques for embodied co-learning in a human-robot team,” Technische Universiteit Delft, Tech. Rep., 2022, mSc Literature Study.
- [17] H. Doorewaard, G. Van Hootegem, and R. Huys, “Team responsibility structure and team performance,” *Personnel review*, vol. 31, no. 3, pp. 356–370, June 2002. [Online]. Available: <https://doi.org/10.1108/00483480210422750>
- [18] C. S. Burke, K. C. Stagl, E. Salas, L. Pierce, and D. Kendall, “Understanding team adaptation: a conceptual analysis and model.” *Journal of Applied Psychology*, vol. 91, no. 6, p. 1189, 2006.
- [19] W. B. Knox and P. Stone, “Interactively shaping agents via human reinforcement: The tamer framework,” in *Proceedings of the fifth international conference on Knowledge capture*, 2009, pp. 9–16.
- [20] C. Celemin and J. Ruiz-del Solar, “Coach: Learning continuous actions from corrective advice communicated by humans,” in *2015 International Conference on Advanced Robotics (ICAR)*, 2015, pp. 581–586.
- [21] C. J. C. H. Watkins and P. Dayan, “Q-learning,” *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992. [Online]. Available: <https://doi.org/10.1007/BF00992698>
- [22] L. Peternel, W. Kim, J. Babič, and A. Ajoudani, “Towards ergonomic control of human-robot co-manipulation and handover,” in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 55–60.
- [23] S. Kana, S. Lakshminarayanan, D. M. Mohan, and D. Campolo, “Impedance controlled human-robot collaborative tooling for edge chamfering and polishing applications,” *Robotics and Computer-Integrated Manufacturing*, vol. 72, p. 102199, 2021.
- [24] E. Magrini, F. Ferraguti, A. J. Ronga, F. Pini, A. De Luca, and F. Leali, “Human-robot coexistence and interaction in open industrial cells,” *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101846, 2020.
- [25] L. Peternel, N. Tsagarakis, and A. Ajoudani, “Towards multi-modal intention interfaces for human-robot co-manipulation,” in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 2663–2669.
- [26] L. Wang, R. Gao, J. Vánca, J. Krüger, X. Wang, S. Makris, and G. Chryssolouris, “Symbiotic human-robot collaborative assembly,” *CIRP Annals*, vol. 68, no. 2, pp. 701–726, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0007850619301593>
- [27] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic

- actor,” in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.
- [28] S. Ramstedt and C. J. Pal, “Real-time reinforcement learning,” *CoRR*, vol. abs/1911.04448, 2019. [Online]. Available: <http://arxiv.org/abs/1911.04448>
- [29] C. Moro, G. Nejat, and A. Mihailidis, “Learning and personalizing socially assistive robot behaviors to aid with activities of daily living,” *J. Hum.-Robot Interact.*, vol. 7, no. 2, oct 2018. [Online]. Available: <https://doi.org/10.1145/3277903>
- [30] I. Papaioannou, C. Dondrup, J. Novikova, and O. Lemon, “Hybrid chat and task dialogue for more engaging hri using reinforcement learning\*,” 09 2017.
- [31] J. Hemminahaus and S. Kopp, “Towards adaptive social behavior generation for assistive robots using reinforcement learning,” in *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2017, pp. 332–340.
- [32] T. G. Dietterich, “Hierarchical reinforcement learning with the maxq value function decomposition,” *Journal of artificial intelligence research*, vol. 13, pp. 227–303, 2000.
- [33] R. S. Sutton, “Temporal credit assignment in reinforcement learning,” 1984, technical Report. [Online]. Available: <http://incompleteideas.net/papers/SS-TR-84-79.pdf>
- [34] T. G. Dietterich *et al.*, “The maxq method for hierarchical reinforcement learning,” in *ICML*, vol. 98. Citeseer, 1998, pp. 118–126.
- [35] J. Chan and G. Nejat, “Social intelligence for a robot engaging people in cognitive training activities,” *International Journal of Advanced Robotic Systems*, vol. 9, p. 1, 10 2012.
- [36] R. S. Sutton, “Introduction: The challenge of reinforcement learning,” *Reinforcement learning*, pp. 1–3, 1992.
- [37] S. P. Singh and R. S. Sutton, “Reinforcement learning with replacing eligibility traces,” *Machine learning*, vol. 22, no. 1-3, pp. 123–158, 1996.
- [38] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.
- [39] S. Kapetanakis and D. Kudenko, “Improving on the reinforcement learning of coordination in cooperative multi-agent systems,” in *Second AISB Symposium on Adaptive Agents and Multi-Agent Systems*. Citeseer, 2002.
- [40] G. Hoffman, “Evaluating fluency in human–robot collaboration,” *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 3, pp. 209–218, 2019.



## Visualization of all the episodes

The overview on the next page (Figure A.2) shows a visualization of all the learning episodes for all the human-robot teams that participated in the experiment. This is done in six sub-figures with subtitles below each one. Each dot represents an episode where the human and robot attempted a handover. The dots are displayed in order over four rows respecting the four 10 minute learning sessions. The gray dots represent failed handovers, while colored dots are succeeded handovers. The failed dots are placed slightly lower than the succeeded episodes to make a clear distinction. The colors of the successful dots correspond to the three colors used for the three distinct strategies from figure 5 in the paper. The shades of gray shows which agents were responsible for failure (see Figure A.1). Figure A.1 shows a legend to Figure A.2.






- Succeeded episodes
-  : Robot lets go (S1)
  -  : Human pulls & Robot lets go (S2)
  -  : Human seizes object (S3)
- Failed episodes
-  : Human Failed
  -  : Robot Failed

Figure A.1: A Legend to the figure on the next page

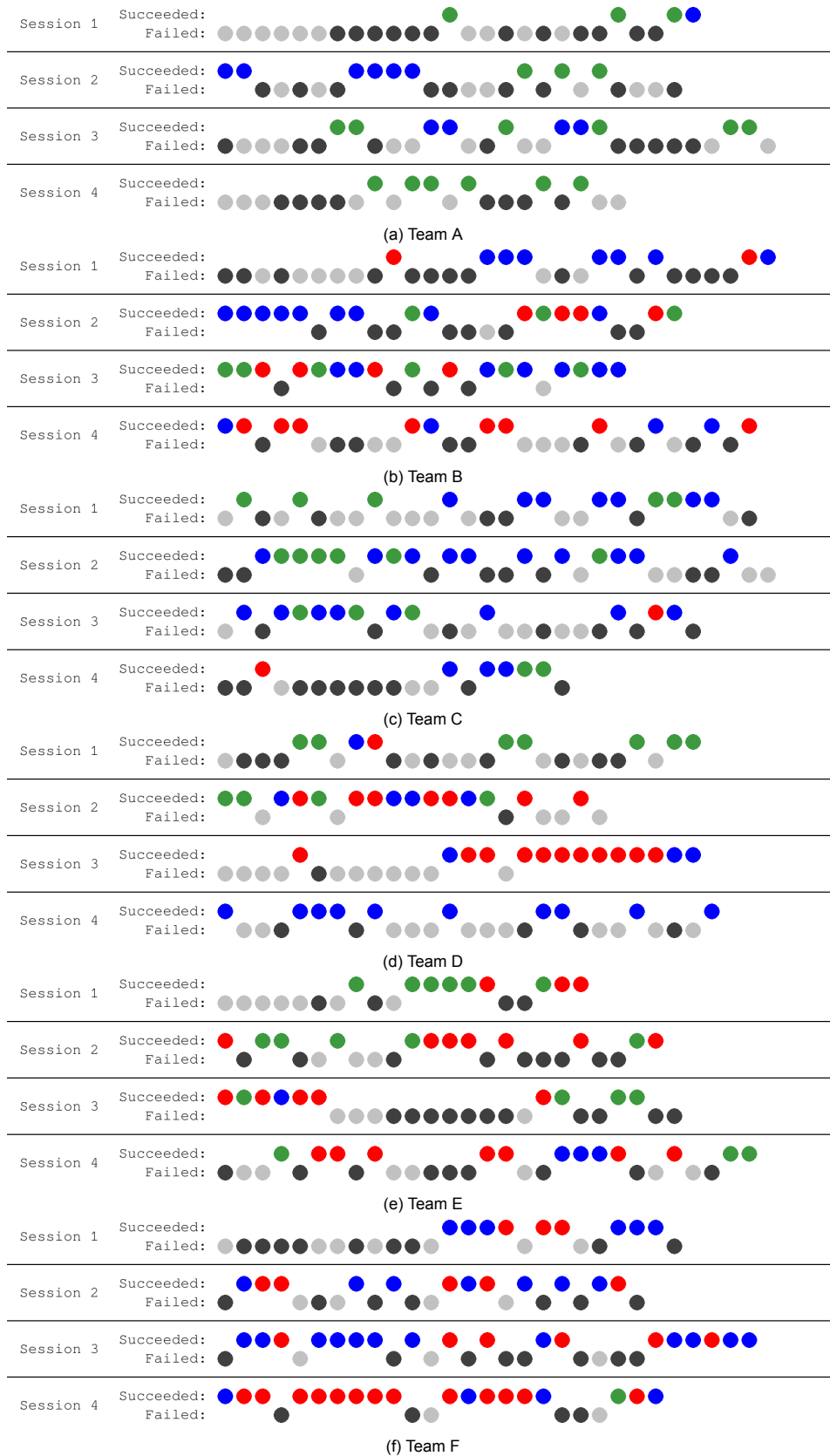


Figure A.2: A visualization of all the episodes of all the teams.

# B

## Action preference in phase 2 for all teams

In the paper the *Action preference* metric, is only displayed for team A (figure 8). In this appendix, we show the same figure for the other teams as well. In these figures, the difference between the two Q-values for the actions *Drop* and *Serve* are shown for a team. The value of this difference is plotted over the episodes for the two states that the robot can observe in phase 2. The two actions determine the orientation in which the robot will convey the object, and the state displays the predominant orientation of the human hand, as explained in subsection III-B.1. The difference between those two Q-values shows which action is preferred during which state, as explained in subsection IV-C.5 and (8). When this metric is negative, the robot learned to choose the action *Serve* in that state, while above the gray null line, it prefers the *Drop* orientation.

In the caption of each figure, a little bit of context of what can be seen in each figure is given.

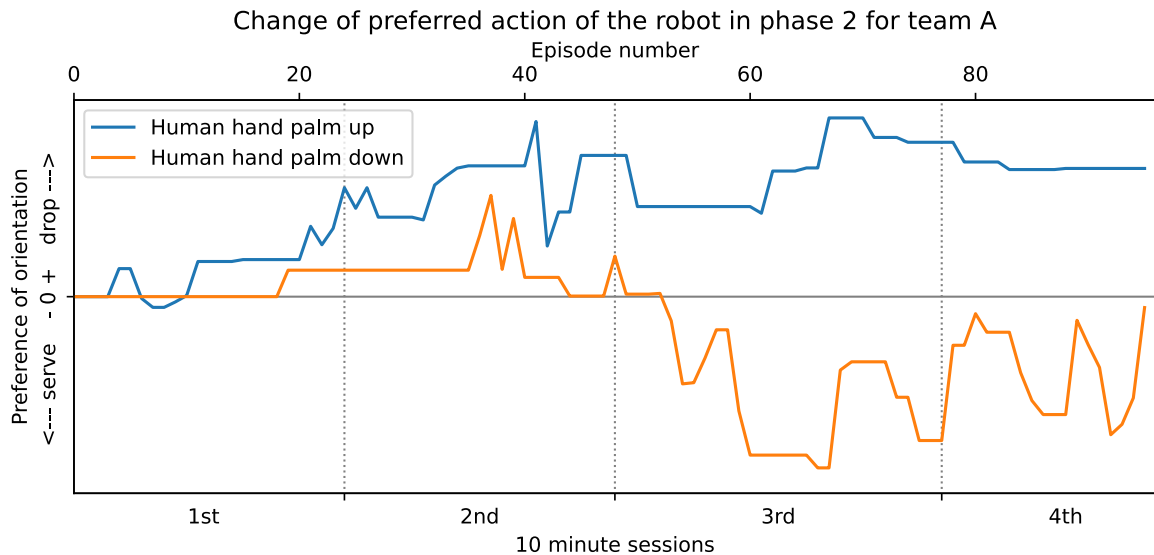


Figure B.1: The difference between the two Q-values for the actions *Drop* and *Serve* in team A for the two states that the robot can observe in phase 2. In the first two sessions, the robot preferred the *Drop* action regardless of the state. From the 3rd sessions forward, the robot preferred *Serve* when the human hand palm was facing down, and *Drop* when the human hand palm was facing up. This figure is further interpreted in the paper.



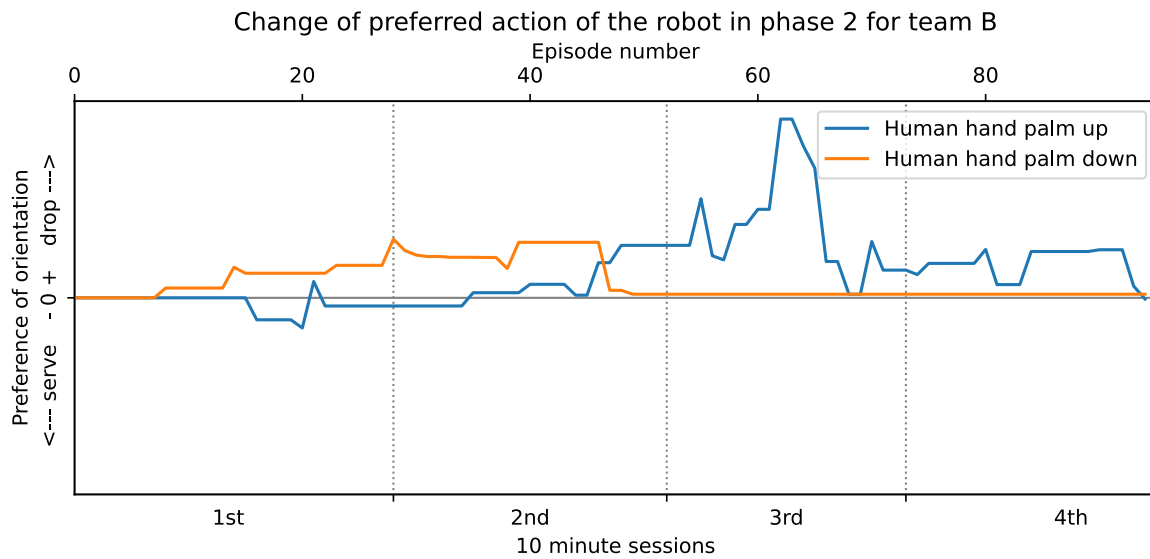


Figure B.2: The difference between the two Q-values for the actions *Drop* and *Serve* in team B for the two states that the robot can observe in phase 2. During all sessions, the robot preferred the *Drop* action regardless of the state. From the end of the 2nd sessions (around episode 45), the human did not hold their hand palm down anymore. This is why the Q-values for this state do not change from this point forward.

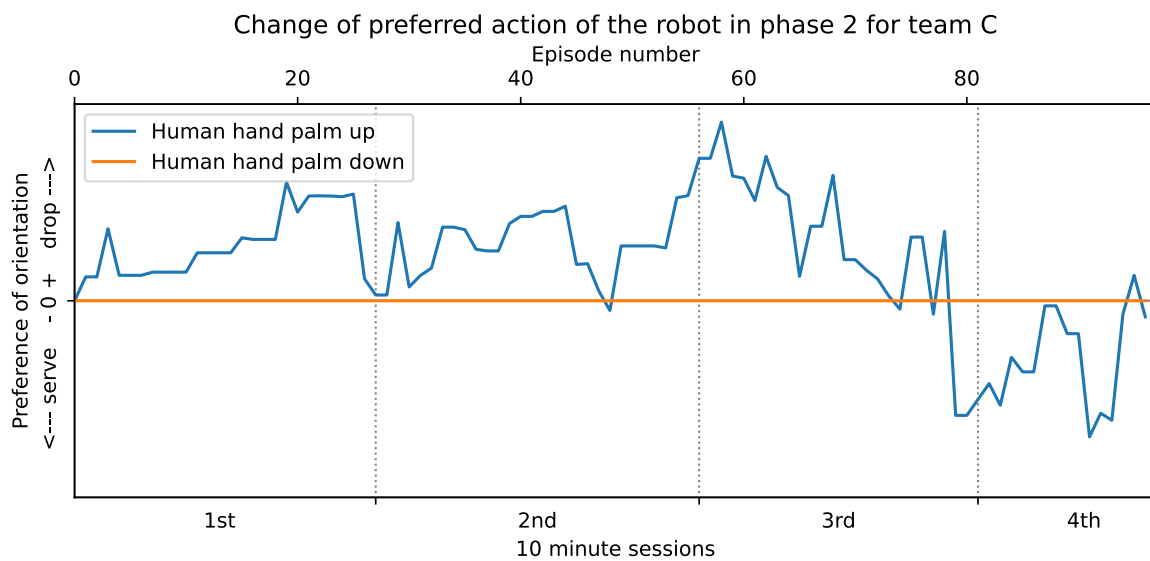


Figure B.3: The difference between the two Q-values for the actions *Drop* and *Serve* in team C for the two states that the robot can observe in phase 2. The human never held their hand palm down during the entire experiment. This is why the Q-values for this state do not change at all, and the metric stays 0 during the experiment. It can also be seen that the robot started with a preference for the *Drop* action, near the end this preference shifted to the *Serve* action. It can however not be said that this was due to the adaptability of the robot, as this policy change was caused by many failed episodes instead of exploration.

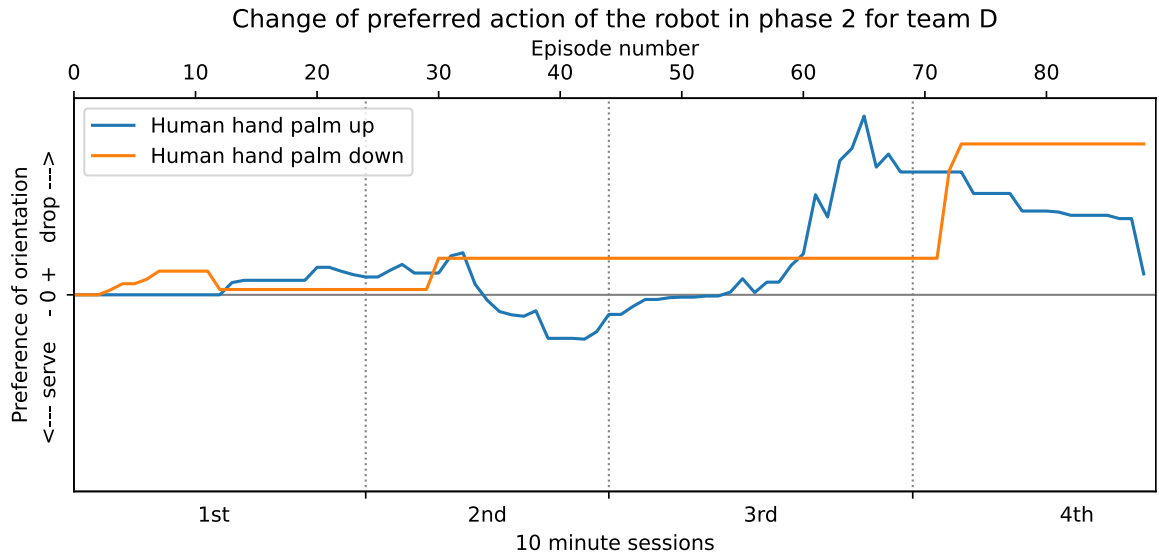


Figure B.4: The difference between the two Q-values for the actions *Drop* and *Serve* in team D for the two states that the robot can observe in phase 2. During almost all episodes, the robot preferred the *Drop* action regardless of the state. There was a brief moment between episode 35 and 50 in which the *Serve* action was preferred when the human held its hand up. This was however swiftly unlearned. The human did not often hold its hand palm down, as the orange line stays horizontal for long stretches of time. When the human did, however, the robot had a positive reinforcement to select the *Drop* action, as the line makes great jumps away from the null line every time this happened.

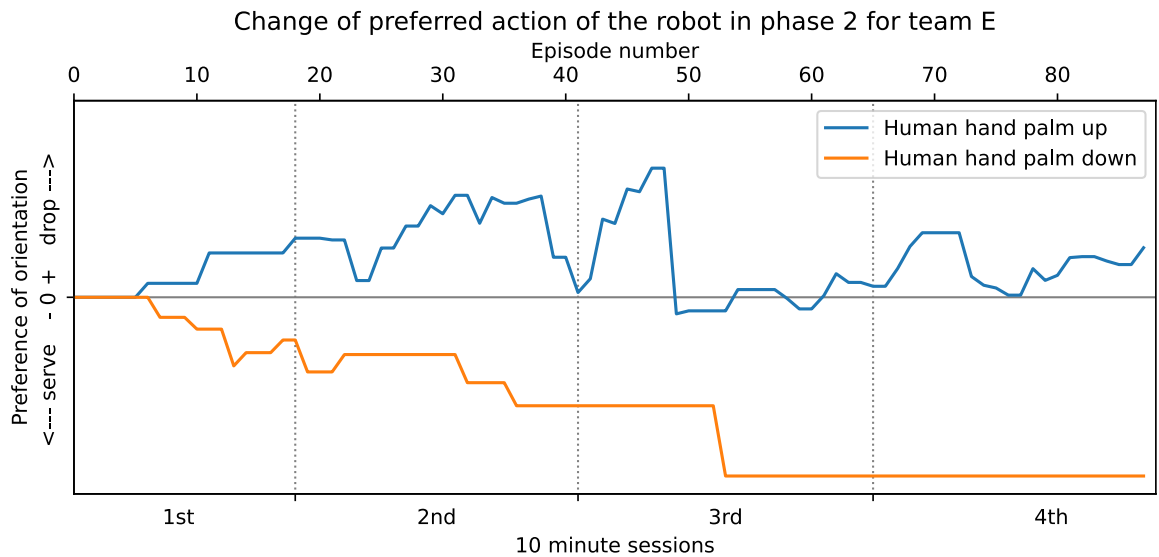
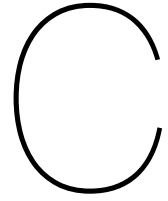


Figure B.5: The difference between the two Q-values for the actions *Drop* and *Serve* in team E for the two states that the robot can observe in phase 2. During the first few episodes, not much was learned by the robot in phase 2. This is explainable, by the fact that the human failed the secondary task often during the first few episodes, meaning that the robot never got the chance to visit phase 2. When the robot did visit this phase, the human had not yet changed any state, meaning that the robot was still in the initial state. When the first few successes were booked, however, a clear distinction was formed directly, the robot prefers *Drop* when the human hand is facing up, and it prefers *Serve*, when the human hand is facing down. Just before episode 50, a series of failed episodes changes this steady collaboration, in the hand palm up state. This tactic was however re-learned from episode 60 onwards.





## Questions of the questionnaire

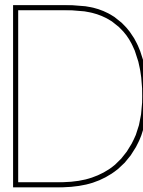
Below, a list of all the questions in the questionnaire is given. The questions are a selection from the 40 questions of the human-robot fluency questionnaire from Hoffman [40]. Questions that did not fit this project were removed. Behind each question, it is shown with a number to which subcategories each question contributed. These numbers correspond to the numbers of the categories below. An (R) behind a question indicates that the Likert scale is reversed before contributing to the average.

The categories:

- |                           |                               |
|---------------------------|-------------------------------|
| (1) Collaboration Fluency | (4) Positive Teammate Traits  |
| (2) Relative Contribution | (5) Perception of Improvement |
| (3) Trust in Robot        | (6) Perception of Shared Goal |

The questionnaire:

1. The human-robot team improved over time (5)
2. The human-robot team worked fluently together (1)
3. The human-robot team's fluency improved over time (1) (5)
4. The robot's performance improved over time (5)
5. The robot contributed to the fluency of the interaction (1)
6. I trusted the robot to do the right thing at the right time (3)
7. The robot was intelligent. (4)
8. The robot was trustworthy (3) (4)
9. The robot was committed to the task (4)
10. I had to carry the weight to make the human-robot team better (R) (2)
11. The robot contributed equally to the team performance (2)
12. I was the most important team member on the team (R) (2)
13. The robot was the most important team member on the team (2)
14. The robot does not understand what I am trying to accomplish (R) (6)
15. The robot and I are working towards mutually agreed upon goals (6)



# Complete results of the questionnaire

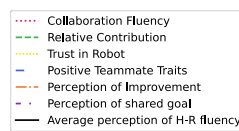


Figure D.1: A Legend to the next figure (D.2)

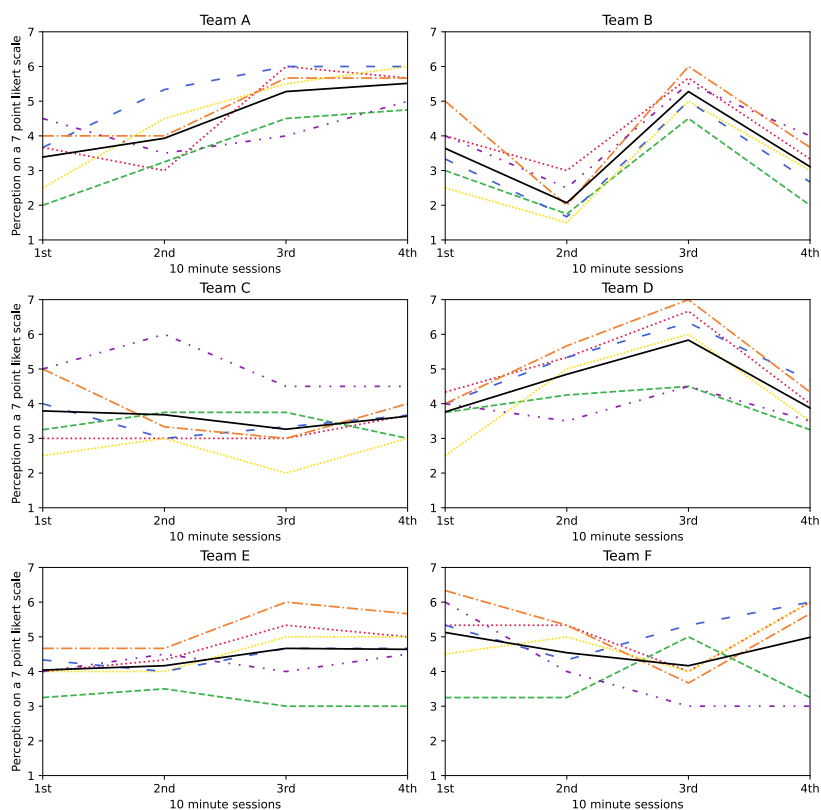
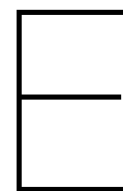


Figure D.2: This figure displays the results from six individual categories of the questionnaire for each participant. The derived average is that is shown in figure 6 in the paper is also displayed here. The legend is shown above.



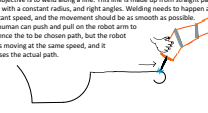


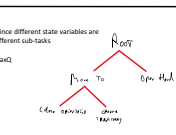
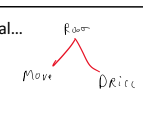
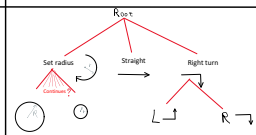
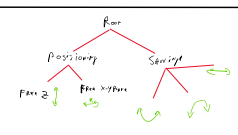
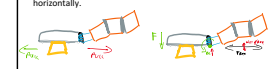


## Overview of task selection process

A part of the selection process of the task that a human and a robot can co-learn together, we used inspirations of human-robot collaborative tasks in relevant research [5],[7],[22],[23],[24],[25],[26]. We narrowed this selection down to four types of task: *Handover*, *Drilling*, *Welding* and *Sawing*. Some other task types were discarded as they are very similar to the tasks that were selected, i.e. *Polishing* is very similar to *Welding* and *Drilling*.

In the prior literature review [16] we indicated multiple challenges and solutions in achieving embodied co-learning and interdependence in human-robot teams. These challenges include giving the robot observability, defining a small state-action space, and decomposing the task so that it can be learned efficiently. The most important challenge is to design the task as such, that the human and the robot have natural hard dependencies, while there is enough room for soft dependencies to grow. We compared each of the four selected task to these challenges, and tried to formulate a solution to them. We summarized this in an overview to compare them with each other. This overview is displayed on the next page.

The final decision to select the handover task for this research, was made because of its natural hard dependence between the agents. It is the only task that can only be done with two agents. The other tasks in the comparison are not an inherent two-agent-task, but single-agent-tasks that were extended for a human and a robot. For instance, you could saw, drill, weld or carry alone, but it is more effective, faster, or takes less effort when it is done together.

	Handover	Drilling	Welding	Sawing
Short description	<p>The human (for instance a surgeon) needs an object and the robot needs to give it to the human. The human shows that it needs the object, and the robot reaches. When the human grabs the object, the robot needs to let go. The goal is to complete the task as fast as possible, without the object being dropped.</p> <p>The Optitrack can be used to localize the human hand, and the soft hand for grasping, and especially letting go at the right time.</p> 	<p>As drilling is similar to polishing, except there is movement in all three dimensions going on, so there are more responsibilities to divide.</p> <p>especially when we drill in uncontrollable positions such as in the ceiling, the robot can take over responsibilities.</p> <p>The specific place and depth of the hole can be different, and the human can decide on that.</p> 	<p>The objective is to weld along a line. This line is made up from straight parts, starts with a constant radius, and right angles. Welding needs to happen at a constant speed, and the movement should be as smooth as possible. The human can push and pull on the robot arm to influence the to be chosen path, but the robot keeps moving at the same speed, and it chooses the actual path.</p> 	<p>The robot and the human saw through a block of wood together. The robot can apply different forces by changing the vertical and horizontal stiffness, or by moving its reference position around.</p> <p>It might use to muscle activity of the human as well to decide whether it should exert forces.</p> 
State space	<p>Optitrack, location of the human hand (annotated)</p> <p>position of the end effector wrt. the goal position (to feel forces)</p>	<p>End effector of the robot</p> <p>For starting or stopping the drilling:</p> 	<p>Position of the robot arm</p>	<p>Position of the robot arm (and)</p> <p>Muscle activity</p>
Action space	<p>Move around (partially shared with human)</p> <p>open/close hand</p> <p>decide on 2 or 3 different possible orientations</p>	<p>Stiffness,</p> <p>End effector goal force</p>	<p>Goal position of the robot arm</p>	<p>Stiffnesses and/or</p> <p>Goal position</p>
Decomposition	<p>This is very useful, since different state variables are important during different sub-tasks, makes it ideal for mARD.</p> 	<p>multi modal...</p> 		
Soft dependencies	<p>This is interesting, the task has a lot of possibilities to force soft dependencies, adjust the task such that there will be possibilities. If the human would be a surgeon, it would be doing a side task, which performance would decrease if the human was looking away. Therefore it creates this dependency in the shared action space where the robot has to find the human, instead of the other way around. We could also use different orientations to create soft dependencies etc.</p>	<p>They both control the position, which is the state. This is their overlapping action space in e soft dependencies can arise. But how?:</p>	<p>They both control the position, which is the state. This is their overlapping action space in e soft dependencies can arise. But how?:</p>	<p>They both control the same movement basically, so there is lots of space for letting go, or taking over by for instance ONLY PULLING or by keeping the vertical position stiff so that the human can use it as a LEVER point while it can mover horizontally.</p> 
Hard dependencies	<p>As a passing task is a 2 agent task, the participants have hard dependencies inherent to this task. For instance:</p> <ul style="list-style-type: none"> <li>- When the robot lets go too early, it will drop the object falling the task.</li> <li>- When the human does not make clear when it has the object, the robot could never learn when to let go.</li> </ul>	<p>Human can't exert force upwards, and it can not keep the drill in place as it can not see it from the right angle without getting dust in their eyes/longs</p> <p>Robot can't place the bit in the right place in the first place, as it can't see where to drill</p>	<p>Human is imprecise</p> <p>Robot does not know what trajectory to follow</p>	<p>Robot can not see where to saw, but this is only in the place face... it can also not see how long the saw for instance is... but the human needs to be dependent of the robot as well</p>
Performance	<p>Completion time</p> <p>Success y/n (did the object drop or not)</p>	<p>Time accuracy</p>	<p>The distance to the to be followed line. The robot can not see the line, and when the line changes every time, it can not learn it by itself.</p>	<p>Muscle activity of the human should be as low as possible, or fatigue, or force or something like that....</p>
What will be learned	<p>Robot: When to open its hand When to move towards the hand What orientation to pass (2 or 3 options)</p> <p>Human: The behavior of the robot How to make the robot let go</p>	<p>Robot: When to start drilling (exhorting force) When to stop When to switch mode</p> <p>Human: The behavior of the robot</p>	<p>Robot: What mode to use</p> <p>Human: How to make the robot change behavior</p>	<p>Robot: When the human saws, and when it does place the saw</p> <p>Human: How to make the robot change behavior</p>