Article

The International Journal of Robotics Research
2015, Vol. 34(14) 1711–1730

DOI: 10.1177/0278364915609673
ijr.sagepub.com

Improved human–robot team performance through cross-training, an approach inspired by human team training practices

Stefanos Nikolaidis, Przemyslaw Lasota, Ramya Ramakrishnan and Julie Shah

Abstract

We design and evaluate a method of human–robot cross-training, a validated and widely used strategy for the effective training of human teams. Cross-training is an interactive planning method in which team members iteratively switch roles with one another to learn a shared plan for the performance of a collaborative task.

We first present a computational formulation of the robot mental model, which encodes the sequence of robot actions necessary for task completion and the expectations of the robot for preferred human actions, and show that the robot model is quantitatively comparable to the mental model that captures the inter-role knowledge held by the human. Additionally, we propose a quantitative measure of robot mental model convergence and an objective metric of model similarity. Based on this encoding, we formulate a human–robot cross-training method and evaluate its efficacy through experiments involving human subjects ($n = 60$). *We compare human–robot cross-training to standard reinforcement learning techniques, and show that cross-training yields statistically significant improvements in quantitative team performance measures, as well as significant differences in perceived robot performance and human trust. Finally, we discuss the objective measure of robot mental model convergence as a method to dynamically assess human errors. This study supports the hypothesis that the effective and fluent teaming of a human and a robot may best be achieved by modeling known, effective human teamwork practices.*

Keywords

Human-robot collaboration, human-robot teaming, cross-training, shared mental model

1. Introduction

Traditionally, industrial robots used in manufacturing and assembly function in isolation from humans; when this is not possible, the work is performed manually. We envision a new class of manufacturing processes that achieve significant economic and ergonomic benefits through robotic assistance in manual processes. For example, mechanics in the fields of automotive and aircraft assembly spend a significant portion of their time retrieving and staging tools and parts for each job. A robotic assistant could improve productivity by performing these non-value-added tasks for the worker. Other concepts for human and robot co-work envision large industrial robotic systems that operate as efficient and productive teammates for human mechanics while sharing the same physical space.

When humans work in teams, it is crucial that the participants develop fluent team behavior; we believe that the same holds for human–robot teams, if they are to perform in a similarly fluent manner. Learning from demonstration (Argall et al., 2009) is one robot training technique that has received significant attention. In this approach, a human explicitly teaches the robot a skill or specific task (Abbeel and Ng, 2004; Akgun et al., 2012; Atkeson and Schaal, 1997; Chernova and Veloso, 2008; Nicolescu and Mataric, 2003). However, the focus is on one-way skill transfer from human to robot, rather than a mutual adaptation process for learning fluency in joint action. In many other works, the human interacts with the robot by providing high-level feedback or guidance (Blumberg et al., 2002; Doshi and Roy, 2007; Kaplan et al., 2002; Thomaz and Breazeal, 2006), but this kind of interaction does not resemble the teamwork processes naturally observed when human team members train together to accomplish interdependent tasks (Marks et al., 2002).

Department of Aeronautics and Astronautics, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, MA, USA

Corresponding author:
Stefanos Nikolaidis, Department of Aeronautics and Astronautics, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA.
Email: snikol@alum.mit.edu

In this paper, we propose a training framework that leverages methods used in human-factors engineering, with the goal of achieving convergent human–robot team behavior during training and fluency at task execution, as perceived by the human partner and assessed by quantitative team performance metrics. Contrary to prior work, the training is conducted in a virtual environment, as training to perform shared-location collaborative tasks with an actual robot could be dangerous or cost-prohibitive. We then evaluate team fluency after training by having the human perform the task with an actual robot. (In this work, a "human–robot team" refers to a robot and a single human user.)

We computationally encode a mental model that captures knowledge about the role of the robot and expectations on the human reaction to robot actions. We refer to this inter-role knowledge as a "robot mental model." This encoded model is quantitatively comparable to the human mental model, which represents the human's preference for his or her own actions, as well as the expectation that the human has regarding the actions of the robot. Additionally, we propose quantitative measures to assess robot mental model convergence as it progresses through a training process, as well as similarity between the human and robot mental models.

We then introduce a human–robot interactive planning method that emulates cross-training, a training strategy widely used in human teams (Marks et al., 2002). We compare human–robot cross-training to standard reinforcement learning algorithms through a human subject experiment incorporating 36 human subjects. Using two-tailed, unpaired t-tests with unequal variance, we show that cross-training improves quantitative measures of human–robot mental model convergence ($p = 0.04$) and similarity ($p < 0.01$). We also present results from experimental analysis indicating that the proposed metric of mental model convergence could be used for dynamic human error detection. Findings from two-tailed Mann–Whitney–Wilcoxon tests on a post-experimental survey indicate statistically significant differences in perceived robot performance and trust in the robot ($p < 0.01$). Additionally, using two-tailed, unpaired t-tests with unequal variance, we observe a significant improvement in team fluency metrics, including a 71% increase in concurrent motion ($p = 0.02$) and a 41% decrease in human idle time ($p = 0.04$) during the actual human–robot task execution phase following the human–robot interactive planning process. This observed improvement in team fluency is indicative of a transfer of the learning experience within a virtual environment to working with an actual robot. Finally, we posited that even after removing the learning component from both algorithms, cross-training would improve the perceived robot performance and team fluency compared with standard reinforcement learning algorithms. We conducted a follow-up experiment to test this hypothesis ($n = 24$), but we did not observe any statistically significant difference in either subjective or objective measures.

In Section 2, we discuss examples of human–robot interaction that motivate our work, and place it in the context of other related work in Section 3. Section 4 presents our computational formulation of the human–robot teaming model (first introduced in Nikolaidis and Shah, 2012), as well as methods to assess mental model convergence and similarity. Section 5 introduces human–robot interactive planning using cross-training, and Section 6 describes our own experiments with human subjects. Section 7 presents and discusses the experiment results, which indicate a significant improvement in team performance using cross-training compared with standard reinforcement learning techniques. We also discuss the applicability of the proposed metric of mental model convergence to dynamic human error detection, and provide more information on the algorithmic performance of human–robot cross-training. Finally, we describe an additional experiment designed to assess the benefit of human adaptation in the cross-training process in Section 8.

We conclude with recommendations for the direction of future research in Section 9.

2. Motivating examples

Although important concepts such as tolerances and completion times are well-defined, many of the details of assembly tasks are largely left up to individual mechanics. Each worker has his or her own preference for how to perform a task, and a robotic assistant should be able to adapt to the preferences of its human partner in order to be an effective teammate. Our aim is to develop a capability that supports efficient and productive interaction between a worker and a robotic assistant, such as the YuMi robot. Potential applications for this capability include time-critical domains, where the capabilities of humans and robots can be harnessed to improve task efficiency. There is a variety of procedures, particularly in manufacturing, that require fast, repetitive execution or physical exertion and are more efficiently performed by robots. Other procedures, such as hand-finishing and assembly of machine parts, require qualities that are difficult to program robots for: human judgment and experience, for example.

Hand-finishing of machine parts is frequently required in various manufacturing processes. In aerospace manufacturing, for instance, this can range from small, coffee-cup-sized parts to large flap tracks and landing gear beams. These parts require post-processing in order to break sharp edges and blend mismatches. In these circumstances, a universal manipulator that holds the part in an ergonomically friendly position while the mechanic refinishes the surfaces can yield benefits for productivity and ergonomics. The most efficient and beneficial position and orientation of the part would depend on the physical characteristics of the mechanic, such as his height and the length of his arms, as well as his particular preference for how to hold the tool.

The assembly of airplane spars is another manual process in which mechanics often develop highly individualized styles for performing their tasks. Consider a mechanic assembling a spar composed of two pieces that must be physically manipulated into alignment. After alignment, wet, sealed bolts are hammered into pre-drilled holes and fastened with collars. Excess sealant is then removed and the collars are re-torqued to final specifications. The sequencing of these tasks is flexible, but subject to the constraint that the sealant must be applied within a specific amount of time after being opened. A robot such as YuMi can assist a mechanic by picking bolts and fasteners from a singulator, rotating them in front of a stationary sealant end-effector and inserting them into the bores. This would allow the mechanic to focus on wiping the sealant, hammering the bolts and placing and torquing the collars, resulting in a productivity benefit through the division of labor and parallelization of tasks.

Our aim is to enable a robotic assistant to adapt to person-specific workflow patterns during training in order to achieve fluent team behavior during actual task execution. We enable the robot to learn a model of human behavior by iteratively switching roles with the human worker. The robot uses this model to learn a sequence of actions necessary for task completion that matches the human preference and is directly comparable to the human mental model. The robot also refines its expectation according to the actions of the human partner. The training occurs in a simulation environment, and is succeeded by work with the real robot in an actual environment. We examine the transfer of the human learning experience from the simulation environment to the actual robot, and compare the fluency of human–robot teams that underwent cross-training to teams that trained using standard interactive reinforcement learning algorithms.

3. Related work

Our work is heavily inspired by human team training practices, applied prior to the execution of tasks or missions with the goal of improving human team performance. We first present an overview of human team training techniques, and then review previous work on human–robot training.

3.1. Human team training practices

In high-intensity domains, such as manufacturing, military and medical operations, there is a variety of tasks that are too complex or cognitively demanding to be performed by individuals working alone. To function as a team, individuals must coordinate their activities; simply bringing together several people to accomplish a task is not enough. Adaptive teams are able to coordinate their activities, not only under routine conditions, but also under novel conditions for which the teams have not been explicitly trained. Poor team coordination has been related to major system failures, such as in the cases of Three Mile Island and Chernobyl (Davis et al., 1986), where deficiencies in interaction and coordination resulted in failure to adapt to changes in the task environment. Studies of team training practices have mainly focused on improving team performance, particularly in response to novel event patterns.

One such technique is *procedural training*, a form of process training in which "operators in complex systems are positively reinforced (through feedback) to follow a standard sequence of actions each time a particular stimulus is encountered" (Gorman et al., 2010). Trainees practice by repetitively following pre-specified procedures, with the goal of learning to respond automatically to stimuli. The underlying assumption is that training in this manner reduces the incidence of errors and enhances performance (Hockey et al., 2007). Procedural training is prevalent in medical, manufacturing and military settings, for tasks during which deviations from complicated procedures can be catastrophic. Whereas this type of training enables team members to reflexively react under stressful conditions and a heavy workload, it is argued that it can also limit a team's ability to transfer their training to novel situations, leading to poor performance when the actual task execution conditions do not match the training conditions (Gorman et al., 2010).

In *cross-training*, another common technique, team members are trained to perform each other's roles and responsibilities in addition to their own (Blickensderfer et al., 1998). There are three types of cross-training: (a) positional clarification, (b) positional modeling and (c) positional rotation. Positional clarification involves verbally presenting team members with information about their teammates' jobs through lecture or discussion. Positional modeling includes observations of team members' roles through video footage or direct observation. Positional rotation is the most in-depth form of cross-training. Study results (Cannon-Bowers et al., 1998; Marks et al., 2002) suggest that positional rotation cross-training, defined as "learning interpositional information by switching work roles," is strongly correlated with improvement in human team performance, as it provides individuals with hands-on knowledge about the roles and responsibilities of their teammates (Marks et al., 2002). Positional rotation cross-training has been used by military tactical teams, as well as aviation crews. It has been argued that shared expectations, resulting from the development of shared knowledge, allow team members to generate predictions for appropriate behavior under novel conditions and in cases when there is uncertainty in the information flow (Marks et al., 2002).

The proposed human–robot cross-training algorithm is inspired by the positional-rotation type of training practice. Whereas in this work we do not examine novel situations in tasks performed by human–robot teams, uncertainty is present due to the inherent lack of transparency between human and robot. Additionally, task execution following

training is conducted in an actual environment, which is inherently different to the virtual environment where training takes place. From an algorithmic point of view, switching roles has the additional benefit of enabling the human to directly demonstrate his preference, as explained in Section 7.5.

3.2. Shared mental models in human teams

The objective of team training is to foster similar or shared mental models, as empirical evidence suggests that mental model similarity improves coordination processes which, in turn, enhance team performance (Marks et al., 2002). The literature presents various definitions for the concept of "shared mental models" (Langan-Fox et al., 2000). Marks et al. (2002) state that mental models represent "the content and organization of inter-role knowledge held by team members within a performance setting." According to Mathieu et al. (2000), mental models are "mechanisms whereby humans generate descriptions of system purpose and form, explanations of system functioning and observed system states and prediction of future system states … and they help people to describe, explain and predict events in their environment." Most researchers agree that there are multiple types of mental models shared among team members. Mathieu et al. (2000) state that one such type is technology/equipment mental models that capture the dynamics and control of the technology among team members. Task mental models describe and organize knowledge about how a task is accomplished in terms of procedures and task strategies, whereas team interaction models describe the roles and responsibilities of team members. Finally, team mental models capture team-specific knowledge of teammates, such as their individual skills and preferences. In this work, we refer to "robot mental model" as the learned sequence of robot actions toward task completion, as well as the expectation that the robot has regarding human actions. We computationally encode this model as a Markov decision process (MDP) (Russell and Norvig, 2003).

3.3. Human–robot team training

While there has been extensive work conducted on human team training techniques, in human–robot team settings, training has focused on one-way knowledge given by a human teacher to a robot apprentice. An example of this method is the SARSA(λ) reinforcement learning approach, where the reward signal is interactively assigned by the human. This technique falls into the category of learning wherein the **human and machine engage in high-level evaluation and feedback**. The general context of an agent learning from human reward is also referred to in literature as *interactive shaping*. While in this section we briefly describe different approaches in this field, we refer the reader to Knox (2012) for a more comprehensive survey.

In some approaches within this category, a human trainer assigns positive reinforcement signals (Blumberg et al., 2002) to a virtual character, a method also known as "clicker training." The state space is represented by a percept tree, which maintains a hierarchical representation of sensory input. The leaf nodes represent the highest degree of specialization, and the root node matches any sensory input. Similarly, state-action pairs consisting of percepts that generate the same action are organized hierarchically, according to the specificity of each percept. Each state-action pair is assigned a reward depending on whether it has good, bad or neutral consequences. The structure of the percept tree and the rewards are refined interactively by a human trainer.

A similar approach is detailed in Kaplan et al. (2002), wherein clicker training is used to train four-legged robots. In this proposed system, the behavior of the robot is implemented through a hierarchical tree of schemata, where each schema is constituted by a set of activation conditions and a set of executable actions. Human feedback is then used to create new behaviors through the combination of existing ones. The robot maintains a user-specific model of human behavior that is updated through interaction and affects the probability of transitions between different schemata.

A user model is also learned in Doshi and Roy (2007), simultaneously with a dialog manager policy in a robotic wheelchair application. The model is encoded in the transition and observation functions and rewards of a partially observable MDP framework. The hidden state represents the user's intent: in this case, the places where the user would like the wheelchair to go. The human interacts with the system by issuing verbal commands, as well as providing a scalar reward after each robot action.

Other methods, such as TAMER-RL (Knox and Stone, 2010, 2012), support the use of human input to guide a traditional reinforcement learning agent in maximizing an environmental reward. The TAMER framework is based upon two insights into how humans assign rewards: first, the reward is delayed according to the time it takes the trainer to evaluate behavior and deliver feedback. Second, a human assigns rewards after considering their long-term effects; in that sense, the reward value more closely resembles a state-action value than an environmental reward in the manner of a MDP framework (Sutton and Barto, 1998). SARSA(λ) is augmented by multiple different approaches to incorporating human reward in TAMER-RL, and their effectiveness is tested through experiments involving a mountain-car and cart-pole task. Q-learning with interactive rewards (Thomaz and Breazeal, 2006) is identical to our version of SARSA(λ) if we remove eligibility traces on SARSA and set a greedy policy for both algorithms. In this case, the algorithm has been applied to teach a virtual agent to cook by following a recipe, with the human assigning rewards to the agent by moving the green slider along a vertical bar. A modified version (Thomaz and Breazeal, 2006) incorporating human guidance has been empirically shown to significantly improve several dimensions of learning. That version

of the algorithm resulted in fewer failures, as the learning process was focused on smaller, more relevant parts of the state space.

The other category for learning in human–robot teams involves **a human providing demonstrations to the machine**. Work involving learning from demonstration includes systems that learn a general policy for a task by passively observing a human expert as he or she executes that task. Following Argall et al. (2009), we can identify three core approaches to deriving policies from demonstration data: learning an approximation to state-action mapping, extracting task constraints and invoking a planner to produce action sequences, and learning a model of the world and computing a policy that optimizes an objective metric.

For example, in Dillmann et al. (1995), skills represented through human demonstrations are captured in a neural network. First, demonstrated trajectories are segmented into time intervals corresponding to different motion classes. A preprocessing phase improves the quality of the demonstrated data, which is then used to train the network offline. The authors demonstrate the applicability of the learned controller on a peg-in-hole task using an industrial robot.

In Chernova and Veloso (2007), the system learns a Gaussian mixture model for each action class, using human demonstrations as training data. Each new data point is assigned to a mixture class according to maximum likelihood. The algorithm also returns a confidence measure, used by the agent to request additional demonstrations. This proposed algorithm is improved through the automatic selection of multiple confidence thresholds in Chernova and Veloso (2008).

More recently, Gaussian mixture models (Akgun et al., 2012) have been used to teach a skill to a robot during experiments in which a human physically guides the robot through a trajectory; this approach is known as "kinesthetic teaching." These experiments have shown that guiding a robot arm through keyframes is a more effective method of teaching means-oriented skills (such as performing gestures) than guiding the robot through the entire trajectory. This is partially due to the difficulty of smoothly manipulating a heavy robot arm. However, demonstrating an entire trajectory has been more successful for goal-oriented skills, such as the performance of pick-and-place tasks.

The second approach to a robot learning from demonstration is to teach a plan to the robot. In Kuniyoshi et al. (1994), a robot extracts a description of the task by tracking human hand motions through a vision system. The observed motions are segmented and clustered into action classes. A symbolic representation of the task hierarchy and operation dependency is then extracted from the operation sequences. The authors show that the proposed framework enables the execution of a pick-and-place assembly task in a different workspace and with a different initial state.

In Nicolescu and Mataric (2003), the authors assume that the robot has an available set of low-level behaviors. Given this assumption, the goal is then for the robot to build a high-level task representation of a more complex, sequentially structured task using its existing behavior set. The robot learns the necessary tasks by creating a link between observations and robot behaviors that achieve the observed effects. Used in addition to human demonstrations, instructional feedback focuses the learning process on the relevant aspects of a demonstration. In the Nicolescu study, experiments in which a Pioneer 2-DX mobile robot attempts to complete a pick-and-place task validate the correctness of learned representations.

In another training method, the robot learns a system model that consists of a transition model from state s given action a, $T(s'|, a)$, and a reward function $R(s)$, which maps states to a scalar reward. Using this system model, a policy that maps states to actions can maximize the finite- or infinite-horizon accumulated reward.

Atkeson and Schaal (1997) consider the problem of having a robot arm follow a demonstrated trajectory. In their paper, the robot learns the transition model through repeated attempts to execute the task, and the reward function is modeled so as to quadratically penalize deviation from the desired trajectory. A priori human knowledge was used to divide a vertical balancing task into a swing-up component and a balancing component. Experiment results indicate improved performance using this approach compared with simply mimicking demonstrated motions.

Billard et al. (2006) reduce the dimensionality of the demonstrations to a subset of relevant features using a hidden Markov model (HMM). They then use these features in a cost function, which produces a measure of the discrepancy between demonstrated and reproduced trajectories. The robot then generates a trajectory that optimizes the cost function while respecting kinematic constraints. The proposed method is validated in a series of experiments wherein a human teaches a manipulation task to a humanoid robot.

Apprenticeship learning (Abbeel and Ng, 2004) generalizes to task planning applications, employing a MDP framework. In this method, the algorithm assumes that the expert tries to maximize a "true" unknown reward function that can be expressed as a linear combination of known "features." A quadratic program is solved iteratively to find feature weights that attempt to match the expected feature counts of the resulting policy with those of the expert demonstrations. Experiment results using this approach indicate that robot performance is similar to that of the expert, even though the expert reward function may not be recovered. This work falls into the category of inverse reinforcement learning, wherein the MDP state reward function is derived from observed expert demonstrations (Ng and Russell, 2000). In multi-agent settings, state-of-the-art behavior modeling based on the game-theoretic notion of regret and the principle of maximum entropy has accurately predicted future behavior in new domains (Waugh et al., 2011).

Our proposed human–robot cross-training algorithm focuses on a team consisting of a robot and a single human user. Rather than asking the human to explicitly provide feedback to the robot, the algorithm allows the human to directly demonstrate robot actions through role-switching, in a manner similar to effective human team training practices. This part of the training resembles inverse reinforcement learning (Ng and Russell, 2000), as the state reward function is learned by human demonstrations when the human and robot switch roles. A key difference from previous work, however, is that we focus on collaborative tasks during which the robot and human actions are interdependent. Therefore, the outcome of the robot actions depends on the human actions, and leads to a learned model of human actions encoded in the transition probabilities of a MDP framework similar to that observed in Doshi and Roy (2007). By following a human-team-inspired approach, we support the mutual co-adaptation of both the human and robot, and focus on team fluency in shared-location, joint-action collaborative tasks rather than on the optimization of agent performance metrics.

4. Mental model formulation

In this section, we computationally encode a mental model for the robot as a MDP. Based on this encoding, we then introduce an objective measure: the entropy rate of the Markov chain to evaluate the convergence of the robot mental model. (The term "robot mental model convergence" refers to the reduction of the uncertainty of the robot on the human actions.) Finally, we propose a metric for human–robot mental model similarity inspired by shared mental model elicitation techniques in human teams.

4.1. Robot mental model formulated as a MDP

Here, we describe how a robot teaming model can be computationally encoded as a MDP. A MDP is a tuple $\{S, A, T, R\}$, wherein:

- S is a finite set of world states; it models the set of world environment configurations.
- A is a finite set of actions; this is the set of actions the robot can execute.
- $T : S \times A \longrightarrow \Pi(S)$ is the state transition function, giving a probability distribution over world states for each world state and action; the state transition function models the uncertainty that the robot has about the human action. For a given robot action a, the human's next choice of action yields a stochastic transition from state s to a state s'. We write the probability of this transition as $T(s, a, s')$. In this formulation, human behavior is the cause of randomness in our model, although this can be extended to include stochasticity from the environment or the robot actions.
- $R : S \times A \longrightarrow \mathbb{R}$ is the reward function, giving the expected immediate reward gained by performing each action in each state. We write $R(s, a)$ for the expected reward for taking action a while in state s.

The policy π of the robot is the assignment of an action $\pi(s)$ at every state s. The optimal policy π^* can be calculated using dynamic programming (Russell and Norvig, 2003). Under this formulation, we define the following terms:

- *Robot mental model of its own role*: the optimal policy π^*, which represents the assignment of robot actions at every state toward task completion. The computation of the optimal policy π^* that captures the robot role takes into account the current estimate of the human behavior, as represented in T.
- *Robot mental model of the human*: the robot's knowledge about the actions of its human co-worker, as represented by the transition probabilities T. The transition matrix represents the probability of a human action, given a state s and a robot action a, and therefore enables the robot to generate predictions about human actions, and, subsequently, future states.
- *Human mental model of his or her own role*: the humans' preference for their own actions.
- *Human mental model of the robot*: the human's expectation regarding the robot action while in a given state.

4.2. Evaluation of robot mental model convergence

As human and robot update their expectations about their teammates' actions over the course of the training process, we expect the human and robot to perform similar patterns of actions. This means that the same states will be visited frequently, and robot uncertainty about human action selection will decrease. In order to evaluate the convergence of the robot's mental model about the human actions, we assume a uniform prior and compute the *entropy rate* (Ekroot and Cover, 1993) of the Markov chain (equation (1)). The Markov chain is obtained by specifying a policy π in the MDP framework.

For π, we use the robot actions that match the preference elicited by the human after training with the robot. Additionally, we use the states $s \in S$ that match the preferred sequence of configurations for task completion. For a finite state Markov chain X with initial state s_0 and transition probability matrix T, the entropy rate is always well-defined (Ekroot and Cover, 1993). It is equal to the sum of the entropies of the transition probabilities $T(s, \pi(s), s')$, for all $s \in S$, weighted by the probability of the occurrence of each state according to the stationary distribution μ of the chain (equation (1)):

$$H(X) = -\sum_{s \in S} \mu(s) \sum_{s' \in S} T(s, \pi(s), s') \log \left[T(s, \pi(s), s') \right] \tag{1}$$

The conditional entropy given by equation (1) represents the uncertainty of the robot about the action selection of

the human, which we expect to decrease as the human and robot train together. This measure can be generalized to encode situations in which the human has multiple preferences or acts stochastically. In Section 7.4, we conduct a post hoc analysis indicating entropy evolution over time in such cases. The entropy rate appears to be particularly sensitive to changes in human strategy, and reflects the resulting increase in robot uncertainty about the next actions of the human. We propose that these results provide intriguing first support for the potential use of entropy rate as a component of a human error detection mechanism.

4.3. Human–robot mental model similarity

Given the formulation of the robot mental model, we propose a similarity metric between human and robot mental models based on prior work (Langan-Fox et al., 2000) on shared mental model elicitation for human teams. In a military simulation study (Marks et al., 2000), each participant was asked to annotate a sequence of actions that he and his teammates should follow in order to achieve mission completion. The degree of mental model similarity was then calculated by assessing the overlap in action sequences selected by each of the team members. We generalize this approach in the setting of a human–robot team: in our study, the participant annotates a sequence of actions that he or she thinks the human and robot should perform in order to complete an assigned task. We then obtain the similarity between the human and robot mental models by determining the ratio of annotated robot actions matching the actions assigned by the optimal policy to the total number of robot actions required for task completion. This describes how well human expectations about robot actions match the actual optimal policy for the MDP.

5. Human–robot interactive planning

Expert knowledge about task execution is encoded into the assignment of rewards R, and in the priors on the transition probabilities T that encode the expected human behavior. This knowledge can be derived from task specifications or from observation of expert human teams. However, rewards and transition probabilities that have been finely tuned to one human worker are not likely to generalize to another human worker, since each worker develops his or her own highly individualized method for performing manual tasks. In other words, a robot that works with one person according to another person's preferences is not likely to be a good teammate.

Empirical evidence suggests that mental model similarity improves coordination processes, which in turn enhance team performance (Marks et al., 2002). Mental model similarity is particularly important under conditions in which communication is difficult due to excessive workload, time pressure or another environmental feature, as teams are unable to engage in necessary strategizing under these circumstances (Mathieu et al., 2005). Shared or similar mental models are important in such cases, as they allow team members to predict the information and resource requirements of their teammates.

In the case of a human–robot team, communication is difficult for different reasons: transparency in the interaction between human and robot is an unsolved problem, mainly due to the technical challenges inherent in the exchange of information about high-level goals and intentions between a human and robot. We therefore hypothesize that a shared mental model for a human–robot team will improve team performance during actual task execution. Cross-training (Marks et al., 2002) is a validated and widely used mechanism for conveying shared mental models in human teams; we emulate the cross-training process that takes place among human team members by having the human and robot train together in a virtual environment. We use a virtual environment because it is infeasible or cost-prohibitive for a robot to perform the human's role in an actual environment, and vice versa, especially in high-intensity applications.

5.1. Cross-training emulation in a human–robot team

We emulate positional rotation in human teams by having the human and robot iteratively switch roles. We name the phase in which the human and robot roles match those of the actual task execution as the *forward phase*, and the phase in which the human and robot roles are switched as the *rotation phase*. In order for the human and robot to develop a shared plan on the collaborative task, the following criteria must be met:

1. The robot must have an accurate estimate of the role the human will have while performing the task. We use the human–robot forward phase of the training process to update our estimation of the transition probabilities that encode the expected human behavior.
2. The actions of the robot must match the preference of the human. We accomplish this by including human inputs in the rotation phase to update the reward assignments.

After each training round, we use the updated transition and reward function of the MDP to compute a new policy for the robot. As the human and robot update their expectations for their teammates' actions throughout the training process, we expect the robot's uncertainty about human action selection to decrease. We can evaluate robot mental model convergence by using the updated transition matrix after each training round, as described in Section 4.2. Additionally, we can assess human and robot mental model similarity after the end of the training process (Section 4.3). The human and robot then perform the task by executing their predefined roles, and team fluency is assessed using the fluency metrics presented in Section 7.1.2. A flowchart of

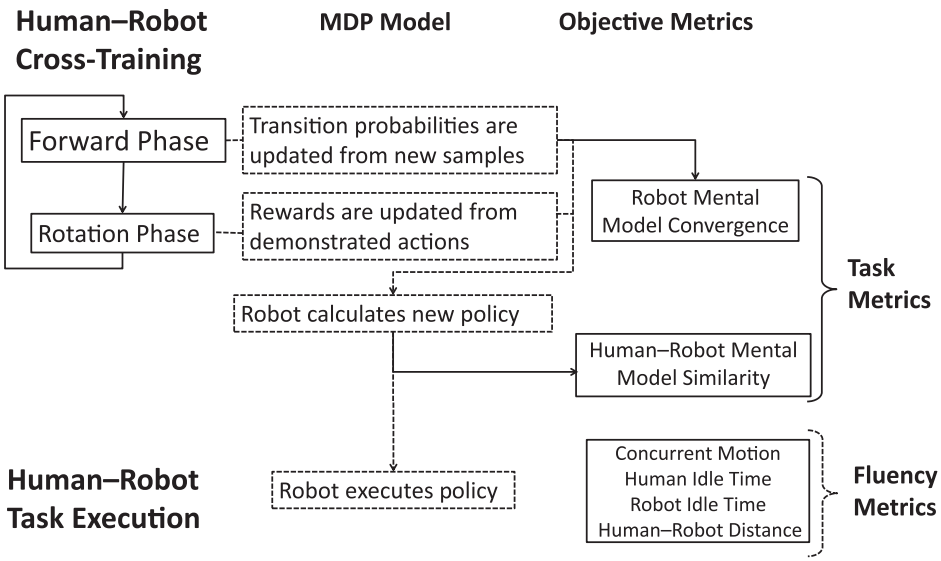

Fig. 1. Cross-training and task execution flowchart.

Algorithm: Human–Robot Cross-training
1. Initialize $R(s, a)$ and $T(s, a, s')$ from prior knowledge
2. Calculate initial policy π
3. **while**(number of iterations < *MAX*)
4. Call Forward-phase(π)
5. Update $T(s, a, s')$ from observed sequence $s_1, a_1, s_2, \ldots, s_{M-1}, a_{M-1}, s_M$
6. Call Rotation-phase()
7. Update $R(s_i, a_i)$ for observed sequence $s_1, a_1, s_2, a_2, \ldots, s_N, a_N$
8. Calculate new policy π
9. **end while**

Fig. 2. Human–robot cross-training algorithm.

the cross-training process and task execution, together with the proposed objective metrics for evaluation, is depicted in Figure 1.

5.1.1. Human–robot cross-training algorithm. The human–robot cross-training algorithm is summarized in Figure 2. In Line 1, rewards $R(s,a)$ and transition probabilities $T(s,a,s')$ are initialized from prior knowledge about the task. In Line 2, an initial policy π is calculated for the robot; we used value iteration (Russell and Norvig, 2003) in our implementation. In Line 4, the forward-phase function is called, where the human and robot train for the task. The robot chooses its actions depending on the current policy π, and the observed state-action sequence is recorded. In Line 5, $T(s,a,s')$ are updated based on the observed state-action sequence. $T(s,a,s')$ describes the probability that, for a task configuration modeled by state s and robot action a, the human will perform an action such that the next state will be s'.

In the rotation phase (Line 6), the human and robot switch task roles. During this phase, the observed actions $a \in A$ are performed by the human worker, whereas the states $s \in S$ remain the same. In Line 7, the rewards $R(s,a)$ are updated for each observed state s and human action a. We then use the new estimates for $R(s,a)$ and $T(s,a,s')$ to update the current policy (Line 8). The new optimal policy is computed using standard dynamic programming techniques (Russell and Norvig, 2003).

In our implementation, we update the rewards (Line 7) as follows:

$$R(s,a) = R(s,a) + r \tag{2}$$

The value of the constant r must be large enough compared with the initial values of $R(s,a)$ for the human's actions to affect the robot's policy. Note that our goal is not to examine the best way to update the rewards, as this has proven to be task-dependent (Knox and Stone, 2012). Instead, we aim to provide a general human–robot training framework, and use the reward update of equation (2) as an example. Knox and Stone (2010) evaluate eight methods for combining human inputs with MDP reward in a reinforcement learning framework. Alternatively, inverse reinforcement learning algorithms could be used to estimate the MDP rewards from human input (Abbeel and Ng, 2004).

We iterate the forward and rotation phases for a fixed number of *MAX* iterations, or until a convergence criterion is met.

5.1.2. Forward phase. The pseudocode of the forward phase is presented in Figure 3. In Line 1, the current state is initialized to the start step of the task episode. The FINAL_STATE in Line 2 is the terminal state of the task episode. In Line 3, the robot executes an action a assigned to a state s, based on the current policy π. The human action is observed (Line 4) and the *next_state* variable is set according to the *current_state*, the robot action a and the human action. In our implementation, we use a look-up table that sets the next state for each state/action combination. Alternatively, the next state could be directly observed after the human and robot finish executing their actions.

```
Function: Forward-phase(policy π)
1. Set current_state = START_STATE
2. while(current_state != FINAL_STATE)
3.     Execute robot action a according to current policy π
4.     Observe human action
5.     Set next_state to the state resulting from current_state, robot
       and human action
6.     Record current_state, a, next_state
7.     current_state = next_state
8. end while
```

Fig. 3. Forward phase of the cross-training algorithm.

```
Function: Rotation-phase()
1. Set current_state = START_STATE
2. while (current_state != FINAL_STATE)
3.     Set action a to observed human action
4.     Sample robot action from T(current_state, a, next_state)
5.     Record current_state, a
6.     current_state = next_state
7. end while
```

Fig. 4. Rotation phase of the cross-training algorithm.

The state, action, and next state of the current time-step are recorded (Line 6).

5.1.3. Rotation phase. The pseudocode of the rotation phase is presented in Figure 4. In Line 3, the action a is the observed human action. In Line 4, a robot action is sampled from the transition probability distribution $T(s, a, s')$.

We note that if the robot has high uncertainty in T, it will sample actions that will drive the training to parts of the state space that may not match the preference of the human. On the other hand, if the robot were to simply repeat the actions that the human performed during the previous forward phase, the learned model would be susceptible to errors or variations on the human actions during the training. By initializing the transition matrix T with a pre-observation count, based on prior knowledge about the human action and gradually updating it after each training round, we allow the robot to learn the human's preference for his or her own actions. This process also achieves robustness to variations on the human actions during the training process.

Just as the transition probability distributions of the MDP are updated after the forward phase, the robot policy is updated to match the human's expectations after the rotation phase. This process emulates how a human mental model would change while working with a partner. A key feature of the cross-training approach is that it also provides an opportunity for the human to adapt to the behavior of the robot.

5.1.4. Time complexity. In the forward training phase, the robot executes its pre-computed policy during each training round using a look-up table. This operation has a constant time complexity $\mathcal{O}(1)$. During the rotation phase, the robot samples actions from the current estimate of transition matrix T, which has a time complexity of $\mathcal{O}(|A_h|)$, where A_h is the number of human actions. After each training round, the robot updates the reward function and transition matrix from the demonstrated human and robot actions performed during the two phases, and computes the optimal policy using value iteration. The complexity of value iteration is $\mathcal{O}(H|A||S|^2)$, where H is the number of iterations until convergence (Kaelbling et al., 1996). Therefore, for I training rounds, the time complexity is $\mathcal{O}(IH|A||S|^2)$. The number of training rounds I required for a human to demonstrate his preference to the robot is task-specific and depends on the size of the state space, and the number of human and robot actions.

We demonstrate the applicability of our approach on a simple place-and-drill task, described in Section 6. We believe that the proposed framework can be used for training a human–robot team in other manufacturing tasks, with well-defined procedures that involve high-precision industrial robots in a constrained environment. One important point is that role switching is beneficial when there are actions that are distinct between the human and the robot. This occurs frequently in the manufacturing setting, where humans and robots have different strengths and weaknesses, as described in Section 2. Additionally, we assume no stochasticity in the robot actions, although the proposed framework can be easily extended to include uncertainty about the robot's actions, as well.

Another assumption we make is that the state space is fully observable. We find this assumption to be reasonable in a constrained manufacturing setting, but it could be limiting within other domains, such as a home environment. Extending human–robot cross-training for partially observable domains is a subject for future work.

5.2. Reinforcement learning with human reward assignment

Here, we compare the proposed formulation to the interactive reinforcement learning approach, wherein the reward signal of an agent is determined by interaction with a human teacher (Thomaz et al., 2005). We use SARSA(λ) with a greedy policy (Sutton and Barto, 1998) as the reinforcement learning algorithm, due to its popularity and applicability to a wide variety of tasks. In particular, SARSA(λ) has been used to benchmark the TAMER framework (Knox and Stone, 2009), as well as to test TAMER-RL (Knox and Stone, 2010, 2012). Variations of SARSA have been used to teach a mobile robot to deliver objects (Ramachandran and Gupta, 2009), for navigation of a humanoid robot (Navarro et al., 2011) and within an interactive learning framework wherein the user administers rewards to a robot through verbal commands (Tenorio-Gonzalez et al., 2010). Furthermore, our implementation of SARSA(λ) would be identical to Q-learning with interactive rewards (Thomaz and Breazeal, 2006) if we removed eligibility traces on SARSA and, in the case of a greedy policy, for both algorithms. The

Function: Reinforcement Learning With Human Reward Assignment (policy π)
1. Set *current_state* = START_STATE
2. **while**(*current_state* != FINAL_STATE)
3. Execute robot action *a* according to current policy π
4. Observe human action
5. Set *next_state* to the state resulting from *current_state*, robot and human action
6. Human enters reward *r*
7. Update robot policy π using SARSA(λ)
8. Record *current_state*, *a*, *next_state*
9. *current_state* = *next_state*
10. **end while**

Fig. 5. Reinforcement learning with human reward assignment.

"reinforcement learning with human reward assignment" algorithm is illustrated in Figure 5. After each human and robot action, the human is asked to assign a good, neutral or bad reward $\{+r, 0, -r\}$. In our current implementation, we set the value of r (the reward signal assigned by the human) to be identical to the value of the reward update in cross-training (equation (2) in Section 5.1) for comparison purposes. In contrast to the cross-training algorithm, where the policy is computed at the end of each training iteration, SARSA(λ) updates the policy online after each human and robot action (Sutton and Barto, 1998). To calculate the entropy rate in the human reward assignment algorithm and compare it with the entropy rate of the human–robot cross-training algorithm, we record the observed state and action sequences and update the transition probability matrix T after each training iteration, identically to the forward phase of the cross-training algorithm.

6. Human–robot teaming experiments

We conducted a human subject experiment ($n = 36$) to compare human–robot cross-training to standard reinforcement learning techniques.

6.1. Experiment hypotheses

The experiment tested the following four hypotheses about human–robot team performance:

- Hypothesis 1: Human–robot interactive planning with cross-training will improve quantitative measures of **robot mental model convergence** and **human–robot mental model similarity** compared with human–robot interactive planning using reinforcement learning with human reward assignment. We base this hypothesis on prior work indicating that cross-training improves the similarity of mental models among human team members (Cannon-Bowers et al., 1998; Marks et al., 2002).
- Hypothesis 2: Participants who cross-trained with the robot will agree more strongly that **the robot acted according to their preferences** compared with participants who trained with the robot by assigning rewards. Furthermore, we hypothesize that they will agree more strongly that **the robot is trustworthy**. We base this hypothesis upon prior work (Shah et al., 2011) that indicated that humans find a robot more trustworthy when it emulates the effective coordination behaviors observed in human teams.
- Hypothesis 3: Human–robot interactive planning with cross-training will improve **team fluency metrics on task execution** compared with human–robot interactive planning using reinforcement learning with human reward assignment. We base this hypothesis on the wide use of cross-training to improve performance in human teams (Marks et al., 2002).
- Hypothesis 4: The **learning experience within a virtual environment** of training with a robot will **transfer** to an improvement in team fluency metrics and subjective performance measures, when **working with the actual robot** at the task execution phase.

6.2. Experiment setting

As a proof of concept, we applied the proposed framework to train a team consisting of a human and a robot to perform a simple place-and-drill task. In the task, there were three holes that could either remain empty or have a screw placed and/or drilled into them. Each screw could be either placed, drilled or not placed, resulting in a state-space size of $3^3 = 27$ states. The robot actions are the predefined task actions {Wait, DrillA, DrillB and DrillC}, where A, B and C correspond to the three possible screw positions. The human actions included either the placement of a screw in one of the empty holes or waiting (no action), while the robot could either drill each placed screw or wait.

Although this task is simple, we found it adequate for the testing of our framework, as there is a sufficient variety of potential ways to accomplish the task among different persons. For example, some participants preferred to place all the screws in sequence from right to left and then have them drilled in the same sequence, while others preferred to place and drill each screw before moving on to the next. For the humans who preferred to place all three screws before drilling, there are $3! \times 3! = 36$ different potential orderings for the placement and drilling of the screws. For those who stated a preference to have each screw drilled immediately after placement, there are $3! = 6$ different possible orderings. Therefore, there are a total of $36 + 6 = 42$ different potential orderings for these two high-level strategies. This is a lower bound on the possible human preferences for this task, as it does not include the case of a mixed strategy, where the human preferred to have the robot drill one screw immediately after placement, but only drill the remaining screws after they had all been placed. The participants consisted of 36 subjects recruited from MIT. Videos of the experiment can be found at http://tiny.cc/5q685w.

6.3. Human–robot interactive training

Before initiating training, all participants were asked to describe, both verbally and in written form, their preferred method of executing the task. We then initialized the robot policy using a set of pre-specified policies, in a way clearly different from the stated preference. We did this in order to avoid the potential trivial case in which the initial policy of the robot matches the preferred policy of the user, and also to evaluate the effectiveness of the training process starting from different human and robot mental models.

To initialize the robot policy, we ran a script that automatically generated reward functions corresponding to the "opposite" of a large variety of possible human preferences. For instance, for the human preference "Drill a screw as soon as it is placed, in the sequence C–B–A", we initialized the robot policy as follows: for every state s where not all screws were placed, we ranked the reward function $R(s, a)$ so that the Wait action would have the highest reward, followed by DrillA (if available), then DrillB, and finally DrillC. For all states s where all screws were placed (and possibly drilled), the highest reward was assigned to the action DrillA, if available, followed by DrillB, then DrillC. Therefore, the resulting initial robot policy was "wait until all screws are placed, then drill them in the sequence A–B–C."

Participants were randomly assigned to one of two groups: Group A or Group B. Each participant then underwent a training session within the ABB RobotStudio virtual environment, where they controlled the white anthropomorphic YuMi robot depicted on the left in Figure 6 while working with the orange industrial robot, "Abbie," on the right. The human chose an action in discrete time steps and observed the outcome by watching YuMi move concurrently with Abbie. The motions of both the human and robot actions were predefined, with a single motion for each action.

Depending on the assigned group, the participant underwent one of the following training sessions:

1. Cross-training session (Group A): the participant iteratively switched positions with the virtual robot, placing the screws during the forward phase and drilling during the rotation phase.
2. Reinforcement learning with human reward assignment session (Group B): this was the standard reinforcement learning approach, wherein the participant placed screws and the robot drilled at all iterations, with the participant assigning a positive, zero, or negative reward after each robot action (Doshi and Roy, 2007).

For the cross-training session, the policy update (Line 8 of Figure 2, Section 5.1) was performed using value iteration with a discount factor of 0.9. The SARSA(λ) parameters in the standard notation of SARSA (Sutton and Barto, 1998) were empirically tuned ($\lambda = 0.9, \gamma = 0.9, \alpha = 0.3$) for optimal task performance.

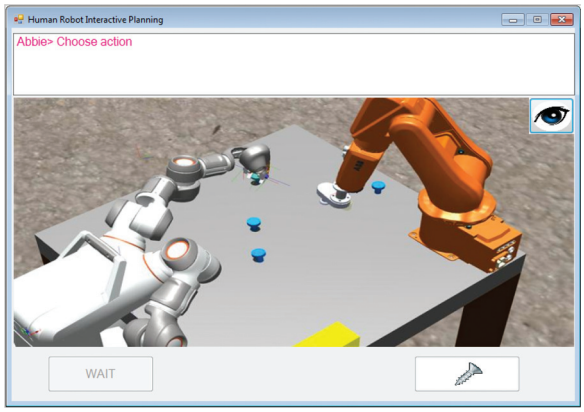

Fig. 6. Human–robot interactive planning using ABB RobotStudio virtual environment. The human participant controls the white anthropomorphic YuMi robot on the left, to work with the orange industrial robot, Abbie, on the right.

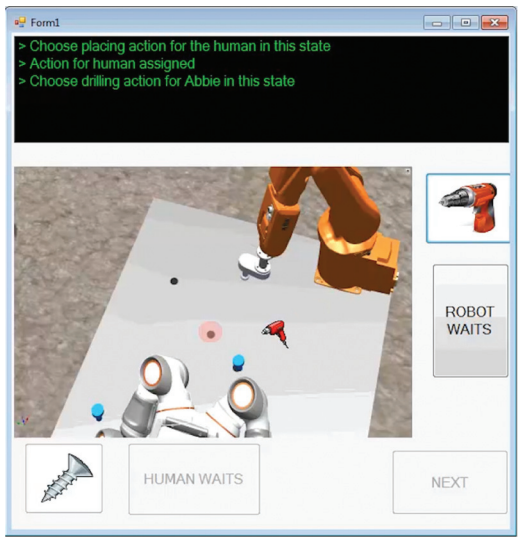

Fig. 7. Human–robot mental model elicitation tool.

After the training session, the mental model of all participants was assessed using the method described in Section 4.3. For each workbench configuration through task completion, participants were asked to choose a placing action and their preference for an accompanying robot drilling action, based on the training they had experienced together (Figure 7).

6.4. Human–robot task execution

Upon completion of the simulation, we asked all participants to perform the place-and-drill task with the actual robot, Abbie. To enable the robot to recognize the actions of the human, we used a PhaseSpace motion-capture system of eight cameras (see http://www.phasespace.com) that tracked the motion of a PhaseSpace glove worn by the participant (Figure 8). Abbie executed the policy as learned from the training sessions. The task execution was recorded

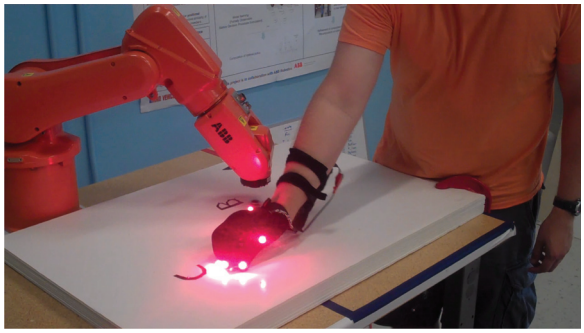

Fig. 8. Human–robot task execution.

and later analyzed for team fluency metrics. Finally, all participants were asked to respond to a post-experiment survey.

7. Results and discussion

The results from the human subject experiments indicated that the proposed cross-training method outperforms standard reinforcement learning in a variety of quantitative and qualitative measures. This is the first evidence that human–robot teamwork is improved when a human and robot train together by switching roles in a manner similar to effective human team training practices. Unless stated otherwise, all the p-values in this section are computed for *two-tailed, unpaired t-tests with unequal variance*. We additionally discuss the applicability of the proposed metric of mental model convergence to dynamic human error detection, and provide more information on the algorithmic performance of human–robot cross-training.

7.1. Objective measures

7.1.1. Task metrics. Here, we evaluate the mental model similarity after the training process, as described in Section 4.3, and the robot mental model convergence, as it evolved over the course of the training (Section 4.2).

Mental model similarity. As described in Section 4.3, we computed the mental model similarity metric as the ratio of human drilling actions matching the actions assigned by the robot policy to the total number of drilling actions required for task completion. Participants in Group A had an average ratio of 0.96, compared with an average ratio of 0.75 in Group B ($p < 0.01$). This shows that participants who cross-trained with the robot developed mental models more similar to the robot teaming model than the participants who trained with the robot by assigning rewards.

Robot mental model convergence. The term "robot mental model convergence" refers to the reduction in the uncertainty of the robot about the actions of the human, as computed by the entropy rate of the MDP (Section 7.1.1). We computed the entropy rate in each training round using the preferred robot policy elicited from the human with the mental model elicitation tool (Figure 7 of Section 6.3). Since the initial value of the entropy rate varies for different robot policies, we used the mean percentage decrease across all participants within each group as a metric to compare cross-training to reinforcement learning with human reward assignment. To calculate the entropy rate during the human reward assignment session, we updated the transition probability matrix T from the observed state and action sequences in a manner identical to how we calculated the entropy rate for the cross-training session. We did so for comparison purposes, as SARSA(λ) is a model-free algorithm and does not incorporate T in the robot action selection (Sutton and Barto, 1998).

Figure 9 shows the entropy rate after each training round for participants in both groups. We considered only the 28 participants who did not change their preference. The difference between the two groups after the final training round was statistically significant ($p = 0.04$), indicating that the robot's uncertainty about the human's actions after training was significantly lower in the cross-training group than in the group that used reinforcement learning with human reward assignment.

We noticed that the cross-training session lasted slightly longer than the session involving reinforcement learning with human reward assignment, as switching roles took more time on average than assigning a reward after each robot action. Since participants often interrupted training in order to interact with the experimenters, we were unable to reliably measure the training time for the two groups.

The above results support our first hypothesis: cross-training improves quantitative measures of human–robot mental model convergence.

7.1.2. Fluency metrics on task execution. We elicited teamwork fluency by measuring the concurrent motion of the human and robot and human idle time during the task execution phase, as proposed in Hoffman and Breazeal (2007). The measurements of the above metrics were evaluated by an independent analyst who did not know the purpose of the experiment, nor whether each participant had been a member of Group A or Group B. Additionally, we automatically computed robot idle time and human–robot distance. Since these metrics are affected by the human's preferred way of performing the task, we used only the subset of 20 participants who self-reported their preferred strategy as "while Abbie is drilling a screw, I will place the next one." (This was the largest subset of participants who reported the same preference on task execution.)

Concurrent motion. We measured the duration for which both the human and robot were concurrently in motion during the task execution phase, and found that participants in Group A who preferred to "finish the task as fast as possible, placing a screw while Abbie was drilling the previous one" had a 71% increase in the time of concurrent motion

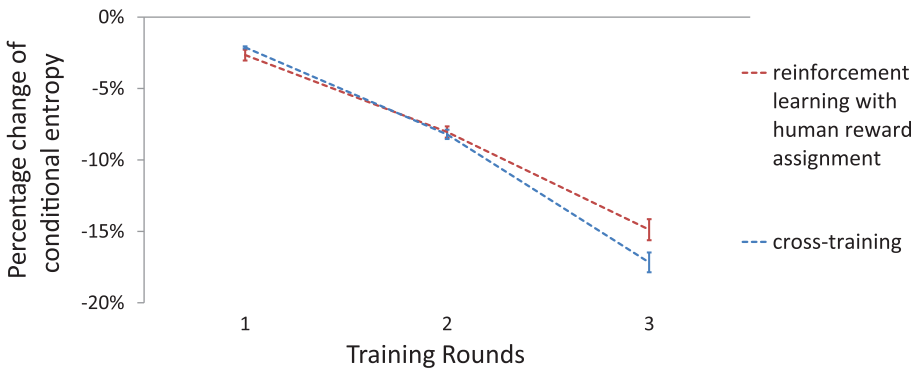

Fig. 9. Robot mental model convergence. The graph depicts the percentage decrease of entropy rate over training rounds. The error bars represent standard errors.

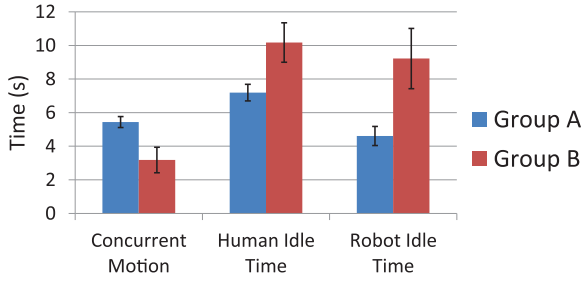

Fig. 10. Team fluency metrics for task execution. The error bars represent standard errors.

with the robot compared with participants in Group B who reported the same preference (A: 5.44 s [SD = 1.13 s]; B: 3.18 s [SD = 2.15 s]; $p = 0.02$). One possible explanation for this difference is that cross-training engendered more trust in the robot, and therefore participants in Group A had more confidence to act while the robot was moving. This possibility is supported by subjective results presented in Section 7.2.

Human idle time. We measured the amount of time each human spent waiting for the robot to perform an action. Participants in Group A spent 41% less time idling, on average, than those in Group B: a statistically significant difference (A: 7.19 s [SD = 1.71 s]; B: 10.17 s [SD = 3.32 s]; $p = 0.04$). In some cases, the increase in idle time occurred because the participant was waiting to see what the robot would do next. In other cases, the robot had not correctly learned the human preference and did not act appropriately, confusing the human team member or forcing them to wait.

Robot idle time. Our task-execution software automatically calculated the time during which the robot remained idle while waiting for the human to perform an action, such as placing a screw. Idle time was significantly shorter in Group A than Group B (A: 4.61 s [SD = 1.97 s]; B: 9.22 s [SD = 5.07 s]; $p = 0.04$).

Human–robot distance. The distance from the human hand to the robot base, averaged over the time the robot spent moving and normalized to the baseline distance from the participant, was significantly shorter in Group A than Group B (A: 23 mm [SD = 26 mm]; B: 80 mm [SD = 73 mm]; $p = 0.03$). This difference occurred because some participants in Group B "stood back" while the robot was moving. Prior work using physiological measures has indicated that mental strain among operators is strongly correlated with the distance of a human worker from an industrial manipulator moving at high speed (Arai et al., 2010). We therefore suggest that cross-training with the robot may have a positive impact on emotional aspects such as fear, surprise and tension, and leave further investigation to be conducted in future studies.

The concurrent motion, human idle time and robot idle time for participants in both groups are shown in Figure 10. The above results support our third hypothesis: human–robot interactive planning with cross-training improves team fluency metrics on task execution, compared with human–robot interactive planning using reinforcement learning with human reward assignment.

7.2. Subjective measures

After each training round, each participant was asked to rate his or her agreement with the following statement on a five-point Likert scale: "In this round, Abbie performed her role exactly according to my preference, drilling the screws at the right time and in the right sequence." Participants were also asked to respond to a survey about Abbie's performance upon completion of the experiment. Subjects who cross-trained and then executed the task with Abbie (Group A) selected a significantly higher mark on the Likert scale than those who trained with Abbie using the standard reinforcement learning method (Group B) for the following statements:

- "In this round, Abbie performed her role exactly according to my preference, drilling the screws at the right time and in the right sequence."

(For the final training round) Group A: 4.52 [SD = 0.96]; Group B: 2.71 [SD = 1.21]; $p < 0.01$.

- "In the actual task execution, Abbie performed her role exactly according to my preference, drilling the screws at the right time and in the right sequence."
  Group A: 4.74 [SD = 0.45]; Group B: 3.12 [SD = 1.45]; $p < 0.01$.
- "I trusted Abbie to do the right thing at the right time."
  Group A: 3.84 [SD = 0.83]; Group B: 2.82 [SD = 1.01]; $p < 0.01$.
- "Abbie is trustworthy."
  Group A: 4.05 [SD = 0.71]; Group B: 3.00 [SD = 0.93]; $p < 0.01$.
- "Abbie does not understand how I am trying to execute the task."
  Group A: 1.89 [SD = 0.88]; Group B: 3.24 [SD = 0.97]; $p < 0.01$.
- "Abbie perceives accurately what my preferences are."
  Group A: 4.16 [SD = 0.76]; Group B: 2.76 [SD = 1.03]; $p < 0.01$.

The p values above are computed for a two-tailed Mann–Whitney–Wilcoxon test. The results show that participants in Group A agreed more strongly that Abbie had learned their preferences than participants in Group B. Furthermore, cross-training had a positive impact on their degree of trust in Abbie, in accordance with prior work (Shah et al., 2011). This supports Hypothesis 2 of Section 6.1, that participants who cross-trained with the robot would agree more strongly that the robot acted according to their preferences and was trustworthy, than those who trained with the robot by assigning rewards. The two groups did not differ significantly when subjects were asked whether they themselves were "responsible for most of the things that the team did well on this task," whether they were "comfortable working in close proximity with Abbie" or whether they and Abbie were "working toward mutually agreed upon goals."

7.3. Transfer of learning experience from virtual to actual environment

The significant differences in team fluency metrics and subjective measures observed between the two groups are indicative of a transfer of the learning experience from the virtual environment to the actual environment. In fact, we observed an intermediate correlation between the mental model similarity metric calculated after the training process, and the time of human–robot concurrent motion at task execution ($r = 0.37$), where r is the Spearman's rank correlation coefficient (Neter et al., 1996). Additionally, we observed an intermediate correlation between the entropy rate after the final training round and the concurrent motion ($r = 0.59$), as well as human idle time ($r = -0.69$), during task execution. Finally, an intermediate correlation was observed between the entropy rate and the participants' Likert-scale response to the statement "In the actual task execution, Abbie performed her role exactly according to my preference, drilling the screws at the right time and in the right sequence" ($r = 0.49$). While these results do not fully support our fourth hypothesis, they are indicative of a transfer of learning from virtual to actual environment and warrant further investigation.

7.4. Dynamic error detection using entropy rate

In this section, we use data collected from the human subject experiment to discuss the entropy rate as a method to dynamically assess change in human preference or to detect a human mistake. A change in human preference during task execution could mean, for instance, that a new user has arrived, and therefore the new human and robot should cross-train together before performing their task. Dynamic detection of human mistakes could serve as an automated inspection mechanism to encourage the human to self-correct, while increased detection of inconsistencies in human actions could be a sign of fatigue.

There is great potential for robots to use this information to improve team efficiency and safety. For instance, the robot could adapt its action selection and motion generation to avoid areas where there is greater uncertainty about human behavior. First, we explain the reason for the entropy rate sensitivity in the human strategy, and then we show the evolution of the entropy rate for two participants in Group A and one participant in Group B who changed their strategy during task execution.

After the training session, we asked all participants to annotate their preferred sequences of human and robot actions toward task completion. We calculated the entropy rate using equation (1) of Section 4.2, taking into account only the states that appeared in the annotated sequence. In addition to the training iterations, we recorded the sequence of states and actions during task execution, updated T and then computed the entropy rate at the end of the execution.

As the human and robot follow a mutually agreed-upon sequence of actions during training, the uncertainty that the robot has about the human actions in these states decreases. On the other hand, if the human changes the sequence of their screw placement at some point, the transition probability distribution over the next states becomes flatter, and the entropy increases. However, this increase appears only when the robot has already correctly learned the user's preference to some degree.

For instance, if the user annotates as their preferred sequence to "have a screw drilled as soon as it is placed in the order A–B–C", the states in the annotated sequence used for the calculation of the entropy rate are "no screw placed," "screw A placed," "screw A drilled and screw B placed," and so on. However, if the robot has not yet learned that it should drill after the user places a screw, most of these states are not reached, and therefore any change in the placement sequence will not affect the entropy calculation.

We present three examples of participants who changed their strategy while working with the robot:

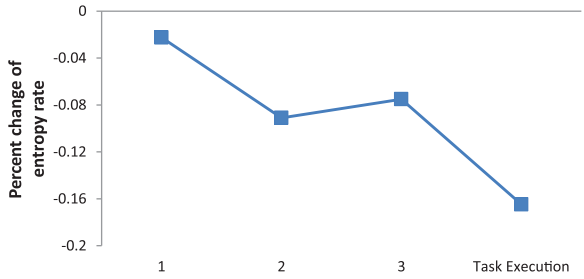

Fig. 11. Entropy percentage rate of Subject 1. The change in the participant's strategy is illustrated by an increase in the entropy rate during the third round.

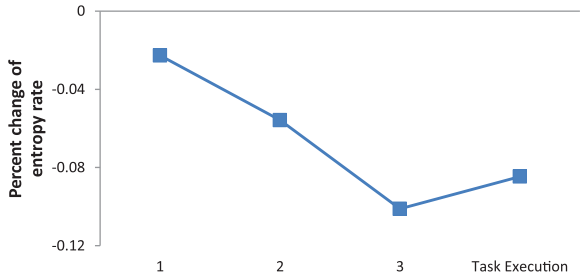

Fig. 12. Entropy percentage rate of Subject 2. The change in the participant's strategy is illustrated by an increase in the entropy rate at task execution.

1. Subject 1, Group A: this user's stated preference was to "place the screws down in the order B–A–C, and Abbie drills them immediately after each one is placed." The user followed this preference during the first two rounds, but changed the sequence from B–A–C to A–C–B during the third round, causing an increase in the entropy rate. At task execution, the user then switched back to the predefined sequence B–A–C, and the entropy decreased again (Figure 11).
2. Subject 2, Group A: this participant followed her initial stated preference of placing the screws in the order C–B–A during training, but switched to the sequence A–B–C during task execution without realizing it. The robot had correctly learned her preference of C–B–A during training, and the result of the change of strategy at execution was a sharp entropy increase, as illustrated in Figure 12.
3. Subject 3, Group B: this user started with the preference of placing the screws in the order C–B–A, with Abbie drilling them as they were put in place. During the first round, the robot did not learn the user's preference, and instead waited for them to finish placing all screws. For the second round, the participant changed the sequence to A–B–C. However, this change affected states that were not included in entropy rate calculation, as explained at the beginning of the session, and therefore entropy remained constant during the second round (Figure 13).

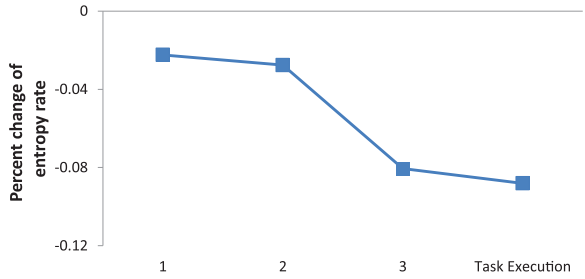

Fig. 13. Entropy percentage rate of subject 3. The entropy rate does not increase when a change in the sequence occurs in states irrelevant to the user preference.

In conclusion, we observed an increase in the entropy rate when there were changes or inconsistencies in execution, and when these changes occurred after the human and robot had converged to a mutually agreed-upon sequence of actions toward task completion. Practical use of this metric as an informative measure at task execution would require confirmation that the robot had correctly learned the human's preference; this confirmation could be obtained by the human upon completion of the training. After each task execution, the entropy rate decrease can be compared to that of a consistent user via a distance metric, and a large deviation can signify a change in human behavior. We leave the testing of this hypothesis for future investigation.

7.5. *Algorithmic performance*

The results presented in Section 7 imply that the robot learned human preferences better through cross-training than training using reward assignment. Upon analysis of the experimental data, we identified three main reasons for this difference.

First, if the robot performs an action that does not match the human's preference, the human will then assign a negative reward, and the SARSA(λ) algorithm will update the value of the corresponding state-action pair that estimates the expected return. When the same state is visited again, the algorithm will most likely not result in the same action, as its value has been reduced. However, the robot will not have any information about which of the other available actions best matches the preference of the human. On the other hand, in the cross-training algorithm, when the human switches roles with the robot, he directly demonstrates his preferred robot action, and the rewards of the visited states are updated. Therefore, the values of the most relevant states are affected to the greatest extent after each iteration, speeding up the learning process for the robot. To verify the above, we calculated for each algorithm the ratio of the number of visited states during training that matched the human preference, as elicited after the training process, to the total size of the state space. For participants in Group A, this ratio was 76%, compared with 67% for participants in Group B, supporting our explanation.

Second, we observed that some participants in Group B had a tendency toward more neutral reward values, even when the actions performed by the robot were very different from their stated preferences. This slowed the performance of SARSA(λ) for these participants. This finding is consistent with those from previous work, which noted that people may be reluctant to give negative rewards to the agent, or use the reward signal as a motivational channel (Thomaz and Breazeal, 2008). Knox and Stone wrote that the general positivity of human reward can actually lead to "positive circuits" (Knox and Stone, 2013): repeatable sequences of behavior that lead to accumulated reward values higher than that of the goal state. Although we did not observe this phenomenon due to the sequential nature of the place-and-drill task, the authors have shown that converting an episodic task into a continuing task can overcome this problem and improve performance (Knox and Stone, 2013).

Third, even though we explicitly asked participants in Group B to evaluate the action that the robot performed after each state, some participants treated the reward as a future-directed signal. For example, one participant preferred that the robot drill a screw as he was placing the next one, in a direction from left to right. During the first training round, the robot's policy was initialized so that it was very different from human preference. Therefore, the human placed the first screws and the robot waited instead of drilling. When the human finished placing all screws, the robot began drilling at the leftmost screw. The participant then assigned a positive reward to the robot, assuming that this would encourage the robot drilling behavior. This resulted in an increase in the estimated value of the state-action pair, "drill screw" at "screw A placed, screw B placed, screw C placed". However, the state "all screws have been placed" would never appear if the robot had followed the human's stated preference of drilling a screw as soon as it was placed. By assigning a positive reward to the aforementioned state-action pair, the human increased the estimated value of a state that did not appear in his preferred sequence of states, and therefore misled the learning algorithm.

In conclusion, the proposed cross-training algorithm outperformed standard approaches incorporating human reward assignment, as it enabled updating of the values of the most relevant parts of the state space, and switching roles is more intuitive to a human participant than assigning rewards. However, reinforcement learning with human reward assignment has proven very effective when a human teacher guides an agent toward maximizing an objective performance metric (Knox and Stone, 2012). We believe that the above observations are helpful for effectively designing the user interface and reward assignment method in such a case.

8. Benefit of human adaptation

The previous experiment did not disambiguate whether the actual benefit observed in human–robot team fluency was derived solely from the learning component of the algorithm, or whether it was also derived from the improved transparency achieved when the human and robot switch roles. Therefore, we added the following hypothesis and conducted an additional experiment involving 24 participants.

8.1. Experiment hypothesis

- Hypothesis 5: Human–robot interactive planning with cross-training will improve objective and subjective measures assessing team fluency and participants' satisfaction compared with human–robot interactive planning using reinforcement learning with human reward assignment, even after removing the learning component of both algorithms. Apart from the better adaptation achieved by the robot, the improved transparency within the interaction played a significant role in the results of the previous section.

8.2. Experiment setting

To test the proposed hypothesis, we conducted an additional experiment involving the same place-and-drill task. In this task, there were three positions that could either remain empty or have a screw placed or drilled into them. Each screw could be either placed, drilled or not placed, resulting in a state-space size of $3^3 = 27$ states. The robot actions were the predefined task actions {Wait, DrillA, DrillB and DrillC}, where A, B and C corresponded to the three screw positions. The human actions included either the placement of a screw in one of the empty holes, or waiting (no action), while the robot could either drill each placed screw or wait.

8.3. Human–robot interactive training

As in the previous experiment, all participants were asked to describe, both verbally and in written form, their preferred method of executing the task prior to beginning the training. We then initialized the robot policy using a set of pre-specified policies, in a way clearly different from the participants' stated preferences, as in Section 6. Also as in the previous section, all participants were divided into two groups:

1. Cross-training session (Group A): the participant iteratively switched positions with the virtual robot, placing the screws during the forward phase and drilling during the rotation phase.
2. Reinforcement learning with human reward assignment session (Group B): this was the standard reinforcement learning approach, wherein the participant placed screws and the robot drilled at all iterations, with the participant assigning a positive, zero, or negative reward after each robot action (Doshi and Roy, 2007).

Contrary to Section 6, we removed the learning component from both algorithms: the change in the reward function from the human actions during the rotation phase in the cross-training session (Group A), and from the reward assignment in the reinforcement learning with human reward assignment session (Group B), was always zero. Therefore, the robot continued to follow its initial policy throughout the training process. This generated an additional design challenge: if the preference of a participant is for the robot to drill a screw as soon as it is placed (efficiency preference), then the initial policy of the robot will be to wait for all screws to be placed first, as explained in Section 6.3. Since the robot will not learn from the human, it will maintain that policy throughout task execution and will always wait. For this particular robot policy, the human idle time and concurrent motion are not indicative of team fluency, as the robot does not move while the human is moving. In the first experiment, 28 out of 36 participants preferred that the robot drill a screw as soon as it was placed. Therefore, in order to have a large enough number of participants interact with a robot that acts while they are moving, we prompted all participants to start with the preference of the robot waiting until all screws are placed first, so that the robot would start with the "opposite" efficient policy. We notified the participants that they could change their preferences during the training. Of the 24 subjects, only one insisted upon following the efficiency preference, and we removed the data for this subject from analysis. Additionally, we had to disregard the data of one other participant due to an error in the experiment process, resulting in a total of 22 samples.

After the training session, the mental model of all participants was assessed using the method described in Section 4.3.

Table 1. Subjective measures for cross-training and SARSA without learning.

Question	Cross-Training No Learning Mean(Std)	SARSA No Learning Mean(Std)
Q1	2.82(1.78)	2.73(1.62)
Q2	2.73(1.01)	3.18(1.25)
Q3	3.36(1.12)	3.55(1.29)
Q4	3.00(1.48)	2.91(1.45)
Q5	3.73(1.27)	3.64(1.21)
Q6	3.27(1.01)	3.09(1.41)
Q7	3.91(1.38)	3.36(1.43)
Q8	3.36(1.03)	3.27(1.35)

Q1: "In the final execution, Abbie did the task according to my preference."
Q2: "I am responsible for most of the things done in this task."
Q3: "Abbie seems more like an assembly tool than a team member."
Q4: "I trusted Abbie to do the right thing at the right time."
Q5: "Abbie is trustworthy."
Q6: "Abbie does not understand how I am trying to do the task."
Q7: "Abbie and I are working towards mutually agreed upon goals."
Q8: "The robot perceives accurately what my preferences are."

Table 2. Objective measures for cross-training and SARSA without learning.

Metric of Team Fluency	Cross-Training No Learning Mean(Std)	SARSA No Learning Mean(Std)
Concurrent Motion	4.41(2.24)	4.74(2.61)
Human Idle Time	10.04(3.54)	9.44(4.52)

8.4. Human–robot task execution

Following Section 6, we asked all participants to perform the place-and-drill task with the actual robot, Abbie. Abbie executed the same policy as in the training session. The task execution was recorded and later analyzed for team fluency metrics. Finally, all participants were asked to respond to a post-experiment survey.

8.5. Results

The results of this experiment were inconclusive, since we did not observe any statistically significant difference in either objective or subjective measures. We present the mean values of participants' ratings of different statements in Table 1, and the mean human idle time and concurrent motion duration for each condition in Table 2. For both conditions, the lowest ratings were given to statements assessing Abbie's ability to learn, which is expected since we removed the learning component of the algorithms. When participants were asked to comment on their overall experience, they responded that "it was annoying that Abbie did not learn what I wanted," that "[Abbie] kept ignoring my suggestion," and wished that "Abbie completed the task in the way I [wanted] her to." It appears that when participants are instructed to execute the task with a specified preference, and the robot is commanded to ignore that preference, participants tend to fixate on the absence of learning throughout the training. However, we believe mutual adaptation can be observed when a human is cross-training with the robot, as implied by the large difference in team fluency metrics during the first experiment and from visual inspection of the videos recorded during task execution. Therefore, we plan to further investigate this hypothesis by conducting an additional experiment in which the robot will learn at the same rate when both switching roles and assigning rewards, using a Wizard-of-Oz process. This will result in isolation of the human adaptation factor of the cross-training, while also providing a realistic training experience.

Table 3. Subjective measures for cross-training with and without learning.

Question	Cross-Training With Learning Mean(Std)	Cross-Training No Learning Mean(Std)
Q1	4.74(0.45)	2.82(1.78)
Q2	3.16(0.76)	2.73(1.01)
Q3	2.84(1.12)	3.36(1.12)
Q4	3.84(0.83)	3.00(1.48)
Q5	4.05(0.71)	3.73(1.27)
Q6	1.89(0.88)	3.27(1.01)
Q7	4.11(0.81)	3.91(1.38)
Q8	4.16(0.76)	3.36(1.03)

Table 4. Subjective measures for SARSA with and without learning.

Question	SARSA With Learning Mean(Std)	SARSA No Learning Mean(Std)
Q1	3.12(1.45)	2.73(1.62)
Q2	3.06(0.43)	3.18(1.25)
Q3	3.29(1.05)	3.55(1.29)
Q4	2.82(1.01)	2.91(1.45)
Q5	3.00(0.93)	3.64(1.21)
Q6	3.24(0.97)	3.00(1.41)
Q7	3.71(0.92)	3.36(1.43)
Q8	2.76(1.03)	3.27(1.35)

Finally, in Tables 3 to 6 we report the results from each of the two conditions alongside the results from the previous experiment. It is challenging to draw conclusions when comparing the algorithms across the two experiments, due to the differences in the experimental design mentioned in Section 8.3. However, the comparison yields interesting insights for future work. In Table 3 we observed the largest differences between the cross-training with learning and without learning conditions for the statements that assessed the robot's understanding of the participants' preferences. This is expected, since the learning component was manipulated across the two experiments. The similarity in the average numerical responses to the statements "Abbie is trustworthy" and "Abbie and I are working towards mutually agreed upon goals" motivates future investigation of the relationship between the robot learning ability and human's trust in the robot when switching roles. In response to open-ended questions, subjects reported feeling "uncomfortable when [Abbie] moved too early," ignoring their preference to wait. A subject also stated that "I was not sure what Abbie was going to do when I started working with the actual robot," since "I did not know if Abbie had learned what I wanted by the end." In fact, the mean human idle time in the cross-training with learning condition was significantly lower than the corresponding value in the cross-training without learning condition, as shown in Table 5. We attribute this to the participants' reported discontent with the robot's inability to learn their preferred way of doing the task during the training, despite having repeatedly demonstrated their preference in the rotation phase. We observed no large differences in the mean ratings and team fluency metrics across the two experiments for the SARSA(λ) condition. In the first experiment, SARSA(λ)'s performance in matching the human preference was affected by the ratio of the state space explored and the manner in which the participants assigned rewards, as explained in Section 7.5. Therefore, the subject ratings were quite low for SARSA(λ) in both experiments.

Table 5. Objective measures for cross-training with and without learning.

Metric of Team Fluency	Cross-Training With Learning Mean(Std)	Cross-Training No Learning Mean(Std)
Concurrent Motion	5.44(1.13)	4.41(2.24)
Human Idle Time	7.19(1.71)	10.04(3.54)

Table 6. Objective measures for SARSA with and without learning.

Metric of Team Fluency	SARSA With Learning Mean(Std)	SARSA No Learning Mean(Std)
Concurrent Motion	3.18(2.15)	4.74(2.61)
Human Idle Time	10.17(3.32)	9.44(4.52)

9. Conclusion

We designed and evaluated a method of cross-training, a strategy widely used and validated for effective training in human teams, for a human–robot team. We first presented a computational formulation of the robot mental model and showed that it is quantitatively comparable to that of a human. Based on this encoding, we formulated human–robot cross-training and evaluated it in a human subject experiment of 36 subjects. We found that cross-training improved quantitative measures of robot mental model convergence ($p = 0.04$) and human–robot mental model similarity ($p < 0.01$), while post hoc experimental analysis indicated that the proposed metric of mental model convergence could be used for dynamic human error detection. A post-experimental survey indicated statistically significant differences between groups in perceived robot performance and trust in the robot ($p < 0.01$). Finally, we observed a significant improvement to team fluency metrics, including an increase of 71% in concurrent motion ($p = 0.02$) and a decrease of 41% in human idle time ($p = 0.04$), during the human–robot task execution phase in the cross-training group. These results provide the first evidence that human–robot teamwork is improved when a human and robot train together by switching roles in a manner similar to that used in effective training practices for human teams.

For this experiment, we focused on a simple place-and-drill task as a proof of concept. We are currently extending

the cross-training algorithm to a complex hand-finishing task, wherein the robot manipulator lifts and places a heavy load at an ergonomically friendly position for the human, whose role is to refinish the surfaces of the load. The best position and orientation of the load depend on the size of the human, his arm length, his age and other physical characteristics, and therefore should be different for each individual worker. Additionally, there is a wide variety of different potential preferences for the sequence of surface refinishing and the velocity of robot motion. As this task must be encoded in a very large state space, we will need to use value-function approximation methods for the reward update of the rotation phase, rather than the currently implemented tabular approach.

Additionally, while cross-training is feasible for small teams, it can become impractical as teams grow in size. Recently, Gorman et al. (2010) introduced *perturbation training*. Using this approach, standard coordination procedures are disrupted multiple times during the training process, forcing team members to coordinate in novel ways to achieve their objective. Perturbation training aims to counteract habituation associated with task processes, a possible outcome of procedural training. It is inspired by prior work in motor and verbal learning and intended to improve team performance under novel post-training conditions (Schmidt and Bjork, 1992). We believe that it would be interesting to introduce perturbation training in a human–robot team setting.

Future work will include extending the computational formulation of the robot's teaming model to a partially observable MDP framework (Kaelbling et al., 1998) incorporating information-seeking behavior, and testing this framework using more complex tasks. Although cross-training is applicable to a wide range of manufacturing tasks with well-understood task procedures, there are also tasks that are difficult to model and simulate in a virtual environment, such as robot-assisted surgery. For these cases, other team training techniques, could be more suitable; however, we leave this assessment for future work.

Acknowledgements

This work is being conducted in collaboration with Thomas Fuhlbrigge, Gregory Rossano, Carlos Martinez, and Biap Zhang of ABB Inc., USCRC – Mechatronics. We would also like to acknowledge the Onassis Foundation.

This work is a revised and expanded version of the paper "Human–robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy" by Stefanos Nikolaidis and Julie Shah, published in the Proceedings of the 8th ACM/IEEE International Conference on Human–Robot Interaction (HRI 2013).

Funding

This work is supported in part by ABB (grant number 6928774) and also in part by the NSF National Robotics Initiative (award number 1317445).

References

Abbeel P and Ng AY (2004) Apprenticeship learning via inverse reinforcement learning. In: *Proceedings of the twenty-first international conference on machine learning*.

Akgun B, Cakmak M, Yoo JW, et al. (2012) Trajectories and keyframes for kinesthetic teaching: A human–robot interaction perspective. In: *Proceedings of the seventh annual ACM/IEEE international conference on human–robot interaction*, New York, NY, pp. 391–398.

Arai T, Kato R and Fujita M (2010) Assessment of operator stress induced by robot collaboration in assembly. *CIRP Annals – Manufacturing Technology* 59(1): 5–8.

Argall BD, Chernova S, Veloso M, et al. (2009) A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57(5): 469–483.

Atkeson CG and Schaal S (1997) Robot learning from demonstration. In: *Proceedings of the fourteenth international conference on machine learning*, San Francisco, CA, pp. 12–20.

Billard AG, Calinon S and Guenter F (2006) Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robotics and Autonomous Systems* 54(5): 370–384.

Blickensderfer E, Cannon-Bowers JA and Salas E (1998) Cross-training and team performance. In: Cannon-Bowers JA and Salas E (eds) *Making Decisions Under Stress: Implications for Individual and Team Training*. Washington, DC: American Psychological Association, pp. 299–311.

Blumberg B, Downie M, Ivanov Y, et al. (2002) Integrated learning for interactive synthetic characters. *ACM Transactions on Graphics* 21(3): 417–426.

Cannon-Bowers JA, Salas E, Blickensderfer E, et al. (1998) The impact of cross-training and workload on team functioning: A replication and extension of initial findings. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 40: 92–101.

Chernova S and Veloso M (2007) Confidence-based policy learning from demonstration using Gaussian mixture models. In: *Proceedings of the 6th international joint conference on autonomous agents and multiagent systems*, New York, NY, pp. 233:1–233:8.

Chernova S and Veloso M (2008) Teaching multi-robot coordination using demonstration of communication and state sharing. In: *Proceedings of the 7th international joint conference on autonomous agents and multiagent systems*, Richland, SC, pp. 1183–1186.

Davis LT, Gaddy CD, Turney JR, et al. (1986) Team skills training. *Performance + Instruction* 25(8): 12–17.

Dillmann R, Kaiser M and Ude A (1995) Acquisition of elementary robot skills from human demonstration. In: *International symposium on intelligent robotics systems*, pp. 185–192.

Doshi F and Roy N (2007) Efficient model learning for dialog management. In: *Proceedings of human–robot interaction (HRI 2007)*, Washington, DC.

Ekroot L and Cover T (1993) The entropy of Markov trajectories. *IEEE Transactions on Information Theory* 39(4): 1418–1421.

Gorman JC, Cooke NJ and Amazeen PG (2010) Training adaptive teams. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 52(2): 295–307.

Hockey G, Sauer J and Wastell D (2007) Adaptability of training in simulated process control: Knowledge versus rule-based guidance under task changes and environmental stress. *Human*

Factors: The Journal of the Human Factors and Ergonomics Society 49(1): 158–174.

Hoffman G and Breazeal C (2007) Effects of anticipatory action on human–robot teamwork efficiency, fluency, and perception of team. In: *Proceedings of the ACM/IEEE international conference on human–robot interaction*, New York, NY, pp. 1–8.

Kaelbling LP, Littman ML and Cassandra AR (1998) Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101: 99–134.

Kaelbling LP, Littman ML and Moore AW (1996) Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* 4: 237–285.

Kaplan F, Oudeyer PY, Kubinyi E, et al. (2002) Robotic clicker training. *Robotics and Autonomous Systems* 38(3): 197–206.

Knox WB (2012) *Learning from human-generated reward.* PhD Dissertation, The University of Texas at Austin, Austin, TX.

Knox WB and Stone P (2009) Interactively shaping agents via human reinforcement: The TAMER framework. In: *The fifth international conference on knowledge capture 2009*.

Knox WB and Stone P (2010) Combining manual feedback with subsequent MDP reward signals for reinforcement learning. In: *Proceedings of the 9th international conference on autonomous agents and multiagent systems (AAMAS 2010)*.

Knox WB and Stone P (2012) Reinforcement learning from simultaneous human and MDP reward. In: *Proceedings of the 11th international conference on autonomous agents and multiagent systems (AAMAS)*.

Knox WB and Stone P (2013) Learning non-myopically from human-generated reward. In: *Proceedings of the 2013 international conference on intelligent user interfaces*, pp. 191–202.

Kuniyoshi Y, Inaba M and Inoue H (1994) Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation* 10: 799–822.

Langan-Fox J, Code S and Langfield-Smith K (2000) Team mental models: Techniques, methods, and analytic approaches. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 42(2): 242–271.

Marks MA, Zaccaro SJ and Mathieu JE (2000) Performance implications of leader briefings and team-interaction training for team adaptation to novel environments. *Journal of Applied Psychology* 85: 971–986.

Marks M, Sabella M, Burke C, et al. (2002) The impact of cross-training on team effectiveness. *Journal of Applied Psychology* 87(1): 3–13.

Mathieu JE, Heffner TS, Goodwin GF, et al. (2005) Scaling the quality of teammates' mental models: Equifinality and normative comparisons. *Journal of Organizational Behavior* 26(1): 37–56.

Mathieu JE, Heffner TS, Goodwin GF, et al. (2000) The influence of shared mental models on team process and performance. *Journal of Applied Psychology* 85(2): 273–283.

Navarro N, Weber C and Wermter S (2011) Real-world reinforcement learning for autonomous humanoid robot charging in a home environment. In: *Proceedings of the 12th annual towards autonomous robotic systems conference*, pp. 231–240.

Neter J, Kutner MH, Nachtsheim CJ, et al. (1996) *Applied Linear Statistical Models*, vol. 4. Chicago, IL: Irwin.

Ng AY and Russell S (2000) Algorithms for inverse reinforcement learning. In: *Proceedings of the 17th international conference on machine learning*, pp. 663–670.

Nicolescu MN and Mataric MJ (2003) Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In: *Proceedings of the second international joint conference on autonomous agents and multi-agent systems*, pp. 241–248.

Nikolaidis S and Shah J (2012) Human–robot interactive planning using cross-training: A human team training approach. In: *AIAA Infotech@Aerospace*.

Ramachandran D and Gupta R (2009) Smoothed Sarsa: Reinforcement learning for robot delivery tasks. In: *Proceedings of the 2009 IEEE international conference on robotics and automation*, Piscataway, NJ, pp. 3327–3334.

Russell SJ and Norvig P (2003) *Artificial Intelligence: A Modern Approach*. New York, NY: Pearson Education.

Schmidt RA and Bjork RA (1992) New conceptualizations of practice: Common principles in three paradigms suggest new concepts for training. *Psychological Science* 3(4): 207–217.

Shah J, Wiken J, Williams B, et al. (2011) Improved human-robot team performance using Chaski, a human-inspired plan execution system. In: *Proceedings of the 6th international conference on human-robot interaction*, New York, NY, pp. 29–36.

Sutton RS and Barto AG (1998) *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

Tenorio-Gonzalez AC, Morales EF and Villaseñor Pineda L (2010) Dynamic reward shaping: Training a robot by voice. In: *Proceedings of the 12th Ibero-American conference on advances in artificial intelligence*, pp. 483–492.

Thomaz AL and Breazeal C (2006) Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In: *Proceedings of the 21st national conference on artificial intelligence*, pp. 1000–1005.

Thomaz AL and Breazeal C (2008) Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence* 172(6): 716–737.

Thomaz AL, Hoffman G and Breazeal C (2005) Real-time interactive reinforcement learning for robots. In: *Proceedings of the AAAI workshop on human comprehensible machine learning*.

Waugh K, Ziebart BD and Bagnell JAD (2011) Computational rationalization: The inverse equilibrium problem. In: *Proceedings of the international conference on machine learning 2011*.