Learning to Interact with a Human Partner

Mayada Oudah, Vahan Babushkin, Tennom Chenlinangjia, and Jacob W. Crandall
Masdar Institute of Science and Technology
Abu Dhabi, United Arab Emirates
{myoudah,vbabushkin,xchenlinangjia,jcrandall}@masdar.ac.ae

ABSTRACT

Despite the importance of mutual adaption in human relationships, online learning is not yet used during most successful human-robot interactions. The lack of online learning in HRI to date can be attributed to at least two unsolved challenges: random exploration (a core component of most online-learning algorithms) and the slow convergence rates of previous online-learning algorithms. However, several recently developed online-learning algorithms have been reported to learn at much faster rates than before, which makes them candidates for use in human-robot interactions. In this paper, we explore the ability of these algorithms to learn to interact with people. Via user study, we show that these algorithms alone do not consistently learn to collaborate with human partners. Similarly, we observe that humans fail to consistently collaborate with each other in the absence of explicit communication. However, we demonstrate that one algorithm does learn to effectively collaborate with people when paired with a novel cheap-talk communication system. In addition to this technical achievement, this work highlights the need to address AI and HRI synergistically rather than independently.

Keywords

Human-Robot Interaction; Online Learning; Cheap Talk

1. INTRODUCTION

As in human-human relationships, many human-robot interactions are punctuated by repeated interaction amid conflicting interests. Due to information asymmetry, modeling effects, and differences between the goals of interacts and robot designers, the robot may pursue an agenda that is not fully shared by its human partner. For example, consider a robot provided by an individual to their elderly parent. The robot may provide many services the elderly parent desires, such as helping him to dress, retrieving the newspaper, etc. However, the robot may also be instructed to ensure that the elderly parent does certain things he might not want to do, such as taking beneficial medications the elderly parent dislikes. In such scenarios, the robot must skillfully interact with the elderly parent to achieve and maintain a collaborative, mutually desirable, relationship.

As in human-human relationships, collaborative outcomes are most likely to be realized when the robot and human can adapt to each other. The ability to adapt to an (also) adapting human partner, requires a robot to employ online learning. However, online-learning algorithms are not yet utilized in most human-robot interactions, particularly in scenarios with conflicting interests. We attribute this deficiency to two previously unsolved challenges. First, most online-learning algorithms designed for interactions with other individuals do not learn at time scales that support interaction with people. They either require thousands of interactions to learn effective behaviors [4, 8], or are incapable of learning collaborative solutions. Both deficiencies severely limit the possibility of effective interactions.

A second challenge to using online learning in human-robot interactions is that most online-learning algorithms rely on random exploration to learn effectively. Random exploration makes the robot unpredictable and (seemingly) irrational, which can cause human partners to form attitudes toward the robot that effectually eliminate the potential for strong collaborations. For example, random exploration could cause the human partner to mistrust the robot's motives, intelligence, and future behavior, thus causing the human to stop interacting with the robot altogether.

We hypothesize that both of these issues can now be overcome by interweaving principles of HRI with recent developments in online learning. Though past online-learning algorithms designed for interaction with others learn too slowly, several recently developed algorithms are reported to learn at time scales that support interactions with people [5, 14, 15]. These learning algorithms still use random exploration, but we anticipate that many of the negative effects of this randomness can be overcome by strengthening the degree of engagement between the human and the learning algorithm. Since *cheap talk* (i.e., non-binding, costless communication) has been shown to improve collaboration among people [17, 3], we consider interweaving cheap talk into an online learning algorithm to strengthen the human-robot interaction.

Interweaving effective cheap talk into an online learning algorithm is nontrivial, as most machine learning algorithms have representations that are not easily interpreted by people. Thus, it is difficult to infer high-level strategic plans from these algorithms' internal representations, let alone determine how to communicate these strategies to people (via

HRI'15, March 2–5, 2015, Portland, Oregon, USA.

http://dx.doi.org/10.1145/2696454.2696482.

speech acts) in arbitrary scenarios. Fortunately, the internal representations of one recently developed online-learning algorithm is more accessible than many of these past algorithms. In this paper, we describe how to generate cheap talk communication within this algorithm.

We make three primary contributions. First, we demonstrate via user study the inability of two recently developed learning algorithms (CFR [15] and MEGA-S++ [6][1]) to effectively learn to collaborate with people, despite the fact that these algorithms have relatively fast learning rates. In the absence of explicit communication, our results also show that people do not consistently learn to collaborate with each other. However, people do learn to collaborate with each other when they are allowed to talk to each other. Thus, in our second contribution, we describe how to generate cheap talk using MEGA-S++. Third, we show that the resulting learning system learns to consistently collaborate with people in two different scenarios. Our results indicate that this learning system learns to collaborate with people nearly as well as people collaborate with each other. These results demonstrate the importance of addressing AI and HRI simultaneously rather than independently.

2. BACKGROUND

This work addresses two technical challenges: online learning in repeated interactions and integrating cheap talk into learning algorithms. We discuss background information and past work related to these two topics after formally defining the domain.

2.1 Repeated Stochastic Games

Many repeated human-robot interactions can be modeled as repeated stochastic games (RSGs; also known as repeated Markov games). These games are played in episodes (or rounds). Each round consists of a sequence of *stage games* (or states) S. In each state $s \in S$, both players (denoted i and $-i$) choose an action from a finite set. Let $A(s) = A_i(s) \times A_{-i}(s)$ be the set of joint actions available in s, where $A_i(s)$ and $A_{-i}(s)$ are the action sets of players i and $-i$, respectively. When joint action $\mathbf{a} = (a_i, a_{-i})$ is played in state s, the players receive the finite rewards $r_i(s, \mathbf{a})$ and $r_{-i}(s, \mathbf{a})$, respectively. The world also transitions to some new state s' with probability defined by $P_M(s, \mathbf{a}, s')$. Each round of the RSG begins in the start state $\hat{s} \in S$ and terminates when some goal state $s_g \in G \subseteq S$ is reached.

In this paper, we consider RSGs with conflicting interests. Both players can profit by collaborating with each other, but either of the players can also possibly exploit the other and thereby obtain even higher payoffs. We assume that the transition model P_M and the reward functions $r_i(s, \mathbf{a})$ and $r_{-i}(s, \mathbf{a})$ are known to both players *a priori*, and that the players can observe each others' actions. This permits the players to focus on learning to interact with each other rather than on learning domain attributes.

2.2 Online-Learning Algorithms for RSGs

Learning in RSGs when associating with people or robots is challenging for several reasons. First, when the other player adapts it strategy over time, the environment is non-stationary, which violates assumptions made by most machine learning algorithms. Second, the behavior of the other player is often unknown. Inferring it's future behavior can be extremely difficult due to the presence of multiple (often infinite) equilibria. Thus, most existing learning algorithms for RSGs either learn too slowly to support interaction with humans or cannot learn collaborative solutions at all.

[1]This version of the paper updates the algorithm's name.

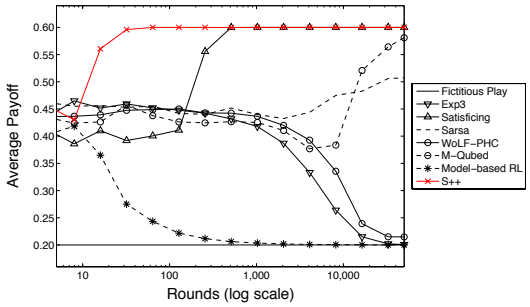


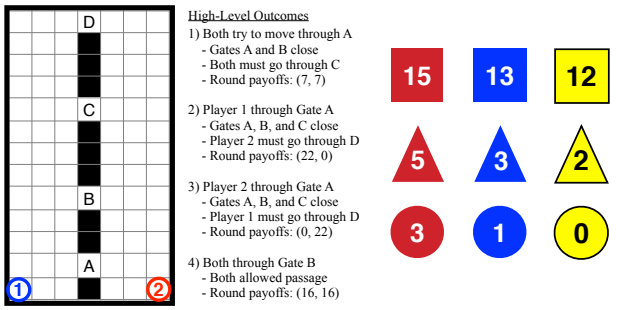
Figure 1: The performance of existing online-learning algorithms in self play in a repeated prisoners' dilemma. A payoff of 0.60 results from mutual cooperation, 0.20 from mutual defection. With the exception of (recently developed) S++, these algorithms either fail to learn to cooperate, or require hundreds to thousands of interactions to do so.

As an example, consider a repeated prisoners' dilemma [2], which comes from a special class of RSGs with only a single state (i.e., $|S| = 1$). Figure 1 shows the performance of a representative set of learning algorithms in self play. Algorithms such as Fictitious play [11] and model-based reinforcement learning quickly learn to defect in this game, which yields low payoffs. On the other hand, M-Qubed [8] and Sarsa [20] usually learn to cooperate with each other, but they require thousands of interactions to do so. Satisficing learning [16, 21] yields better results in self play, but these algorithms are somewhat exploitable [7].

Recently, two new algorithms have been developed for RSGs that appear to learn at faster time scales than previous algorithms. These algorithms are counter-factual regret (CFR) [15] and MEGA-S++ [6]. CFR has been used with substantial success in relatively large competitive games, such as poker. Prior to interaction, CFR attempts to compute an equilibrium strategy by simulating an interaction using self play. As it interacts with its associate, it continues to update this model. While CFR is rather myopic in games of conflicting interest [15], it provides an interesting baseline for what online-learning algorithms can currently achieve when interacting with a human partner.

MEGA-S++ is an expert algorithm that extends S++ [5] to general RSGs. Prior to interaction, MEGA-S++ computes a set of expert strategies, several of which are Nash equilibria of the repeated game. During interactions, it uses aspiration learning to determine which expert to follow. It has been shown to quickly learn non-myopic solutions in a number of RSGs of conflicting interest when associating with other learning algorithms [6].

We note that online learning has been used successfully in scenario-specific human-robot interactions in which the robot and human have common interests [13, 19]. Our work differs from these works in that we seek to identify more general-purpose algorithms that learn to interact with human partners in arbitrary scenarios, including those with conflicting interests.



(a) The SGPD (b) The Block Game

Figure 2: Two RSGs used in our user studies.

2.3 Cheap Talk

Cheap talk refers to non-binding, unmediated, and costless communication [9, 1]. Cheap talk has been cited as a means for equilibrium refinement [10], and has been shown to improve collaborations among people [17, 3].

In this paper, we consider whether a learning robot can employ cheap talk to help overcome the negative effects of random explorations carried out by a learning algorithm. We hypothesize that cheap talk by a robot will cause it to be held in higher regard by people, even when the robot's behavior is governed by the same learning algorithm. Thus, we anticipate that cheap talk will help the robot to better interact with people.

We are unaware of past work interweaving cheap talk with online learning. This is potentially due to the difficulty of knowing what to communicate, as most learning algorithms have representations that are difficult to articulate. However, we note parallels to the work of Thomaz and Breazeal [22]. In their work, a simulated robot signaled to a human teacher uncertainty by pausing in states in which multiple actions had similar Q-values. This helped the human to understand what the robot still needed to learn. However, in domains of conflicting interest, discussions about local Q-estimates are likely to be at too low of a level to fully resonate with potential human collaborators. Additionally, we anticipate that the competitive natures of these games will make such communications insufficient.

3. LEARNING WITH HUMANS

But is cheap talk even necessary? Given the fast learning rates of CFR and MEGA-S++, we first consider whether these algorithms can learn to effectively interact with people in the absence of explicit communication. To do this, we conducted a user study to compare the performance of people and these algorithms when paired with other people in two different RSGs. We now describe these two scenarios, after which we discuss the experimental setup and results.

3.1 Scenarios (RSGs)

The two games were consider are a stochastic game prisoner's dilemma and a block-sharing game.

3.1.1 A Stochastic Game Prisoner's Dilemma

The stochastic game prisoner's dilemma (SGPD) [12] is a maze game in which the high-level payoffs of the game equate to a standard prisoner's dilemma [2]. At the start of each round of the game, the players are placed in opposite corners of the maze as shown in Figure 2(a). The players move (simultaneously) to adjacent cells (up, down, left, or right) with the goal of reaching the other player's start position in as few moves as possible. Each move costs a player one point, but a player receives 30 points when it reaches its goal. Once both players have arrived at their respective goals, a new round begins from the original start state.

To reach their goals, the players must pass through one of four gates (Gates A-D). While the least-cost path to the goal is through Gate A, only one player can pass through Gate A in a round. When a player passes through Gate A, Gates A, B, and C close for the other player, and it must pass through Gate D. If both players attempt to pass through Gate A at the same time, neither is allowed passage and Gates A and B close. On the other hand, both players can pass through Gates B, C, and D separately or at the same time, though Gate A closes when either player passes through Gate B.

We expect players to eventually converge to one of four solutions, which we list in descending collaborative order:

1. *Both B* – Both players use Gate B, resulting in each receiving 16 points.
2. *Alt A-B* – The players take turns going through Gate A, which results in an average payoff of 11 points to each player.
3. *Both A* – Both players attempt to go through Gate A, thus requiring each player to pass through Gate C. This gives both players 7 points.
4. *Bully* – One of the players always goes through Gate A, leaving the other to go through Gate D. This gives the players 22 and 0 points, respectively.

3.1.2 A Block-Sharing Game

The Block Game is a turn-taking game in which the two players share the set of nine blocks shown in Figure 2(b). In each round, the players take turns selecting a block until they each have three blocks, with one player (typically the older sibling) going first in each round. If a player's three blocks form a valid set (i.e., she has all blocks of the same color, all blocks of the same shape, or none of her blocks have the same color or shape), then her payoff in the round is the sum of the numbers on her blocks. If she fails to collect a valid set, she loses the sum of her blocks divided by 4.

Though rather simple, this game is strategically complex. Each player would like to collect all of the squares (40 points) or all of the red blocks (23 points). However, to reach either of these outcomes, the other player would have to accept either getting all of the triangles (10 points) or all of the blue blocks (17 points), respectively. Since the other player can ensure itself 18 points by taking blocks that all differ in shape and color, it is unlikely to repeatedly accept either of those two outcomes. Thus, a player has to decide whether to try to bully the other player (possibly by punishing the other player, which might require the acceptance of negative points, in early rounds in order to get better outcomes in later rounds) or to collaborate with them in some way. This decision depends on the characteristics of the other player.

We expect the players to eventually converge to one of five solutions, which we list in descending collaborative order:

1. *Alt* □-△ – The players alternate between selecting all of the squares and all of the triangles. Thus, both players average 25 points per round $((40 + 10)/2)$.

2. *Alt rd-blu* – The players take turns selecting the red and blue blocks (respectively) in alternating rounds, Both players average 20 points $((23 + 17)/2)$.
3. *All diff* – The players both select blocks with no matching attributes. This always gives both players 18 points.
4. *Pure rd-blu* – One of the players always takes all the red blocks (23 points) while the other player always takes the blue blocks (17 points).
5. *Pure* □-△ – One player always takes the squares (40 points) while the other gets the triangles (10 points).

3.2 Experimental Design

In this initial user study, we evaluate how effectively people and two learning algorithms interact with other people in two different scenarios. We used a 3x2 mixed factorial design, in which the between-subjects variable is the player type (Humans, CFR, or MEGA-S++) and the within subjects variable is the scenario.

A convenience sample of forty-eight participants with an average age of 26.3 years were recruited from the Masdar Institute community. Twelve subjects were randomly paired with CFR, twelve with MEGA-S++, and twenty-four subjects were paired with each other. The participants took part in the study in groups of four. The study proceeded as follows:

1. Each participant was assigned an associate (Human, CFR, or MEGA-S++) without their knowledge.
2. The rules of the SGPD were explained to the participants until it was clear they understood all aspects of the game.
3. The participants played a 54-round SGPD paired with their assigned player type. The game was played on a desktop computer (using the arrow keys to select movements). The participants were not told the duration of the game or who they were paired with. They also were not allowed to talk or signal to each other.
4. Each participant filled out a post-experiment survey, which asked questions related to their experience playing the SGPD. Questions included the participant's assessments of their associate. For example, they were asked how smart they thought their associate was, whether they thought their associate was a robot or person, and how likable their associate was.
5. The participants were trained on how to play the Block Game until they understood all aspects of the game.
6. The participants played a 51-round Block Game with the same player type as before. If paired with Humans, the participant was randomly paired with a different person than in the first game. Each participant was randomly assigned to be either the first or second player to select a block – a role which was kept constant throughout the game. The participants used the mouse to select the blocks on a GUI interface.
7. The participant completed the same post-experiment survey as in Step 4.

Participants were paid a $5.00 show-up fee. To incentivize the participants to try to maximize their own payoffs, participants were also paid money proportional to the points they scored in the games (they could earn up to an additional $15.00). The GUI displaying the game interface also showed the amount of money the participant had earned.

3.3 Performance Metrics

We use three metrics to evaluate the learning algorithms in this study: average payoffs, solution quality, and humanness. We are particularly interested in comparing the algorithms' performance to that of humans with respect to each metric.

3.3.1 Average Payoffs

The average payoffs received by the algorithms are perhaps the most salient measure of successful learning. While we are interested in the payoffs obtained over the full course of the game, we are particularly interested in the payoffs achieved in later rounds, as these payoffs are indicative of how well the algorithm has learned to interact with people.

3.3.2 Solution Quality

Solution quality refers to the collaborative nature of the solutions learned by the algorithms. We count the number of participants in each condition and game that achieved each of the solutions listed in Section 3.1. This metric gives us a good idea of the ability of the learning algorithms to learn to cooperate (when beneficial) with human partners.

3.3.3 Humanness

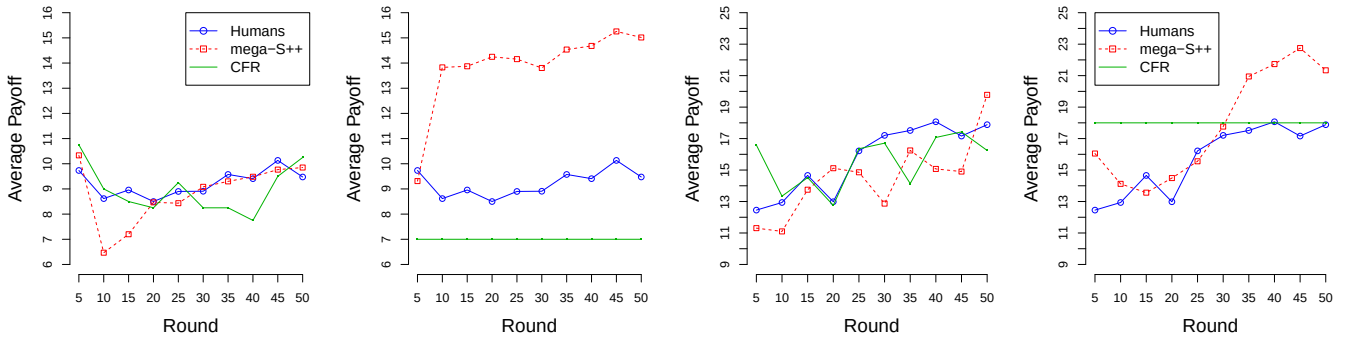
We consider how human-like the algorithms seemed to study participants. We make this assessment using the post-experiment questionnaire, in which participants were asked to guess whether their associate was a person or a robot, and to rate their confidence (on the scale one to five, with one being a complete guess and five being certainty) in their guess. Formally, let $H_{j,k}$ be the *humanness* attributed by participant j toward associate k, and let $h_j = 1$, when the participant guesses human, and $h_j = -1$ otherwise. Also, let $C_j \in \{1, 2, 3, 4, 5\}$ be participant j's confidence in his/her guess. Then, $H_{j,k}$ is given by

$$H_{j,k} = h_j * (C_j - 1). \quad (1)$$

3.4 Results

The average payoffs obtained by the three player types when paired with people are shown in Figures 3(a) and 3(c). There is no clear and substantial difference between the payoffs obtained by each player type. Additionally, neither CFR, MEGA-S++, nor Humans learned to consistently collaborate at a high level with people in either game. In the SGPD, the average round payoff received in the last few rounds by each player type was approximately 10, which is well short of the 16 points that are achieved from mutual cooperation. In the Block Game, none of the player types had an average payoff close to the 25 points they could have achieved had they learn to play *Alt* □-△. Thus, while both Humans and MEGA-S++ appear to slowly improve their payoffs with time, neither reaches a high level within 50 rounds.

Table 1 provides further insight into the ability of the learning algorithms to collaborate with people. This table shows the number of participants that achieved each outcome (by the end of 50 rounds) when paired with each player type. In the SGPD, both MEGA-S++ and Humans learned to cooperate (*Both B*) with some participants (MEGA-S++ cooperated with five, Humans with three). The rest of the participants typically learned mutual defection (*Both A*) when paired with MEGA-S++ and Humans, though MEGA-S++ did continue to attempt to pass through Gate B every few rounds in hopes of reaching a more-profitable collaboration.



(a) SGPD – With people (b) SGPD – Self play (c) Blocks – With people (d) Blocks – Self play

Figure 3: Average payoffs, grouped in 5-round chunks, when paired with people and in self play. CFR's and MEGA-S++'s payoffs in self play were obtained from the average of 25 simulation runs.

(a) SGPD (no communication)

Player	Primary Outcome				
	Both B	Alt A-B	Both A	Bully	Other
Humans	3	0.5	7.5	1	0
MEGA-S++	5	0	5	0	2
CFR	0	0	11	1	0

(b) Block Game (no communication)

Player	Primary Outcome					
	Alt □-△	Alt rd-blu	All diff	Pure rd-blu	Pure □-△	Other
Humans	2	2	5	2	0	1
MEGA-S++	1.5	0.5	4.5	2	0	3.5
CFR	0	0	5	5	0	2

Table 1: The number of subjects that reached each outcome at the end of 50 rounds in each condition.

In the Block Game, Humans and MEGA-S++ rarely reached the most collaborative solutions. CFR did not learn to play highly collaborative solutions in either game.

Humanness varied significantly by player type ($F(2, 90) = 5.96$, $p = 0.004$). Pairwise comparisons show that Humans were rated more human-like than CFR ($p = 0.005$). The difference between Humans and MEGA-S++ was marginally statistically significant ($p = 0.078$), with Humans having a higher rating. Thus, though some participants were convinced that robots were people and vice-verse, they were largely able to distinguish the behavior of people from robots.

While CFR, MEGA-S++, and Humans all failed to consistently learn collaborative solutions when interacting with people in this study, they did learn to collaborate in some instances. The potential of MEGA-S++ to collaborate with people in these two scenarios is further illustrated by Figures 3(b) and 3(d), which show the average payoffs of the players in self play. MEGA-S++ quickly achieves very high levels of cooperation in the SGPD, and also does so in the Block Game after about 30 rounds. Thus, we anticipate that MEGA-S++ could learn to consistently collaborate with people if it could better communicate its desires and intentions.

4. ADDING CHEAP TALK

As cheap talk has been shown to help people to collaborate with each other [17, 3], we now address whether interweaving cheap talk into MEGA-S++ produces a learning system that learns to effectively and consistently collaborate with people. In this section, we address two questions. First, what forms of cheap talk might be useful (if any)? Second, how can cheap talk be integrated into MEGA-S++?

4.1 Forms of Cheap Talk

We consider two kinds of cheap talk: *feedback cheap talk* and *planning cheap talk*. We refer to cheap talk that addresses assessments of past events as feedback cheap talk. As an example, a person or robot might comment on their satisfaction with past events, or comment on how past events made them feel. Additionally, feedback cheap talk includes assessing the past behaviors of the associate, perhaps expressing how the other person's actions make them feel, or expressing what they wished the other person had done instead. We anticipate that feedback cheap talk could be produced from most learning algorithms with careful thought.

Planning cheap talk is forward looking. It involves suggesting future behavior to one's associate and/or revealing one's current or future strategy. Of course, such cheap talk is non-binding – neither of the players must actually do what is spoken. However, conforming behavior with one's cheap talk can help to establish a reliable reputation, and can thus help to mitigate negative effects caused by random explorations. We anticipate that most learning algorithms are too cryptic to easily produce effective planning cheap talk, as humans typically communicate plans at higher levels than typical machine-learning algorithms reason.

4.2 Generating Cheap Talk

Unlike many learning algorithms, MEGA-S++ is structured so that its high-level strategies are expressible to people. This allows it to produce both feedback and planning cheap talk that is generic such that the same cheap talk can be used in any RSG.

MEGA-S++ is an expert algorithm that operates on a set of experts $\Phi = \{\phi_1, \cdots, \phi_n\}$. Each expert $\phi \in \Phi$ encodes a particular strategy defined over the entire state space S of the RSG. In each round, MEGA-S++ selects an expert $\phi \in \Phi$ to follow. It uses aspiration learning [16] to determine which expert to follow in each round. That is, player i encodes an aspiration α_i^t, which is updated after each round as follows:

$$\alpha_i^t = \lambda \alpha_i^{t-1} + (1 - \lambda) R_i^t. \quad (2)$$

(a) Finite state machine (with output) for "fair" leader experts.

State	State Transitions: Events						Speech Acts (Output): Events					
	sel	g	i	s	d	p	sel	g	i	s	d	p
s0	s1	s0	s0	s0	s0	s0	1	–	–	–	–	–
s1	–	s9	s1	s2	s2	s1	–	r(5-9)+4+r(10-12)	–	r(13-16)	–	–
s2	–	s10	s2	s3	s3	s2	–	r(5-9)+4+r(10-12)	–	r(13-16)	–	–
s3	–	s11	s3	s4	s4	s3	–	r(5-9)+4+r(10-12)	–	r(13-16)	–	–
s4	–	s11	s4	s5	s5	s4	–	r(5-9)+4+r(10-12)	–	r(13-16)	–	–
s5	–	s11	s5	s6	s6	s5	–	r(5-9)+4+r(10-12)	–	r(13-16)	–	–
s6	–	s11	s6	s7	s7	s6	–	r(5-9)+4+r(10-12)	–	–	–	–
s7	–	s11	s7	s8	s8	s7	–	r(5-9)+4+r(10-12)	–	3	–	–
s8	–	s11	s8	s8	s8	s8	–	r(5-9)+4+r(10-12)	–	–	–	–
s9	–	s9	s3	s10	s10	s9	–	–	2	–	–	r(17-19)
s10	–	s10	s4	s11	s11	s10	–	–	2	–	–	r(17-19)
s11	–	s11	s5	s11	s11	s11	–	–	2	–	–	r(17-19)

(b) Definitions of event symbols.

Symbol	Explanation
sel	Algorithm selects an expert.
g	Associate has profited from deviating from the "cooperative" solution (defined by the expert).
i	Associate is now innocent. It has been punished for its deviation.
s	The robot is satisfied with its round payoff.
d	The robot is dissatisfied with its round payoff.
p	Associate received a lower payoff on a move than it would have had it always cooperated.

(c) Speech acts for "fair" leader experts.

1. **Here's the deal. Let's cooperate with each other. If you do not cooperate, I'll punish you thereafter.**
2. **I forgive you. Cooperation will bring us both a higher payoff**
3. Sweet. We are getting rich. Let's continue this.
4. I trusted you to *<game specific action label*
5. You jerk!
6. You buffoon!
7. You fool!
8. Curse you!
9. You are an idiot!
10. **I'm going to teach you a lesson you will not forget.**
11. **I'm going to make sure you do not profit from this malicious act.**
12. **You will pay for this!**
13. That what I wanted.
14. That's what I'm talking about.
15. Excellent!
16. Great!
17. Take that!
18. Serves you right, jerk.
19. In your face!

(d) Speech acts for "fair" follower experts.

1. **Let's cooperate with each other.**
2. Sweet. We are getting rich. Let's continue this.
3. I thought you should *<game specific action label*
4. You betrayed me!
5. That was selfish of you
6. That was not fair!
7. Are you only thinking of yourself?
8. This is not good for our relationship.
9. For the sake of our relationship, cease this untoward behavior.
10. Friends do not do that to each other.
11. That's what I wanted.
12. That's what I'm talking about!
13. Excellent!
14. Great!

Table 2: Feedback and planning cheap talk is generated for each expert using a finite state machine. r(x-y) denotes a randomly selected speech act between the numbers of x and y in the speech-act table. '+' indicates concatenated strings. Planning speech acts are given in bold, feedback speech acts are in plain text.

Here, R_i^t is player i's total payoff in round t, and $\lambda \in [0,1]$ is a learning rate. α_i^t is player i's threshold for evaluating satisfaction. Experts that produce payoffs that exceed α_i^t are desirable (and MEGA-S++ will continue to play them), whereas experts that produce lower payoffs are not.

4.2.1 Feedback Cheap Talk

MEGA-S++ can be used to produce both high- and low-level feedback cheap talk for arbitrary RSGs. In addition to helping the robot to determine which experts to select, α_i^t provides a means for the robot to express its satisfaction with a round's outcome. Such talk could implicitly communicate whether the robot is likely to continue its same strategy, which could help the human partner to establish appropriate expectations.

Like Thomaz and Breazeal [22], MEGA-S++ also communicates its satisfaction for individual, low-level, actions using estimates of state quality or potential (such as Q-values, which are encoded by some of MEGA-S++'s experts). When the human partner executes an action that lowers the robot's potential payoffs in a round, the robot expresses its disappointment, and even explicitly states what it wishes the human had done instead. This could help the human to better distinguish which of its actions are "upsetting" the robot.

4.2.2 Planning Cheap Talk

MEGA-S++ produces planning cheap talk for two different kinds of events. First, because each expert $\phi \in \Phi$ encodes a (perhaps radically) different high-level strategy, the random selections of experts (i.e., exploration) can appear to be disjoint, irrational reasoning. Thus, when MEGA-S++ changes which expert it follows, it produces a speech act that notifies the human partner of this change. Example speech acts include "I've changed my mind," and "I've had a change of heart." Additionally, when this switch in strategies leaves some promised act undone (such as a promised punishment), the robot tries to soften the discontinuity by saying something like "I'll let you off this time."

Each expert $\phi \in \Phi$ can also produce planning cheap talk. These plans can be communicated at a high level, as most experts used by MEGA-S++ encode a high-level strategic ideal that is easily understood by the people. For example, one of MEGA-S++'s experts (called Bouncer) seeks to minimize the difference between the robot's and the human's payoffs. When Bouncer is selected, MEGA-S++ announces "I will play fair if you will play fair," and that it insists on equal payoffs and will not be cheated. Others of MEGA-S++'s experts encode trigger strategies, which carry out stages of cooperation and punishment depending on the behavior of the associate. These trigger strategies can easily be announced when they are selected. Furthermore, these trigger strategies can be modeled with simple finite state machines [18], which MEGA-S++ uses to produce speech acts that inform its associate as it switches between stages of cooperation and punishment.

As an example, Table 2 shows the state machine, event symbols, and speech acts, of a leader expert seeking to enforce a fair and pure target solution (e.g., the *Both B* strategy in the SGPD). When initiated, the expert announces that it would like to "cooperate" with its associate, and that it will "punish" deviations from the cooperative solution. If the associate fails to cooperate (indicated by event `g`), the robot curses, states what the person did wrong, and then says that it will punish him. Once the punishment has been carried out, the robot states "I forgive you," reminds its associate that cooperation will bring them both a higher payoff, and then returns to cooperating. Speech acts for a similar follower expert [5], generated using the same state machine, is shown in Table 2d.

We generated similar state machines for leader strategies that target different target solutions (e.g., the *Alt* □-△ solution in the Block Game). The only difference is that this solution requires the players to take turns getting a higher

Figure 4: A Nao delivered speech acts to subjects.

payoff. Thus, our algorithm produces a speech act stating the turn-taking nature of the desired solution, and announces who's turn it is to get the higher payoff.

Since the concepts of "cooperation," "punishment," and "forgiveness" apply to arbitrary situations, this same speech-generation system can be used in any RSGs. Thus, we use the same speech system for both games we consider.

5. THE IMPACT OF CHEAP TALK

To determine to what extent cheap talk helps a robot to learn to effectively collaborate with a human partner, we conducted a second user study. In this section, we describe the user study and discuss the results.

5.1 Experimental Setup

To understand how cheap talk impacts how well MEGA-S++ learns to collaborate with people, we tested two different cheap talk systems. The first system produced only feedback cheap talk. The second system was identical to the first, except that it also produced planning cheap talk. In both cases, the cheap talk was delivered by a Nao robot, who was placed before the participants as they played (Figure 4).

Forty-eight people, with an average age of 25.1 years, were recruited from the Masdar Institute community to participate in this study. Twelve subjects were paired with MEGA-S++ using each of the two cheap-talk systems, which we denote MEGA-S++(f) and MEGA-S++(fp), respectively. So as to provide a baseline condition, we also paired twenty-four participants with each other. These participants were allowed to talk to each other as they played the games.

Except for the communication (and, in turn, the knowledge of who one was paired with), this study was carried out in the same manner as the initial study (Section 3.2).

5.2 Results

Cheap talk greatly enhanced the performance of Humans (when they were paired together) in both games (Figures 5). All human-human pairings in the SGPD converged to *Both B* (Table 3a). Convergence in the Block Game was a little more diverse (Table 3b), though more highly collaborative outcomes were observed than without communication.

Cheap talk also greatly enhanced the ability of MEGA-S++ to learn to interact with study participants. An ANOVA shows that cheap talk substantially increased the payoffs of MEGA-S++ when paired with people in the last ten rounds of the games ($F(2, 66) = 6.04; p = 0.004$). Pairwise comparisons reveal that MEGA-S++(fp) performed significantly higher then MEGA-S++ (with no cheap talk; $p = 0.003$), but MEGA-S++(f) did not ($p = 0.234$).

(a) SGPD

Player	Primary Outcome				
	Both B	Alt A-B	Both A	Bully	Other
Humans	3	0.5	7.5	1	0
Humans (fp)	12	0	0	0	0
MEGA-S++	5	0	5	0	2
MEGA-S++ (f)	5.5	0	5.5	1.0	0
MEGA-S++ (fp)	9.5	0	2	0.5	0

(b) Block Game

Player	Primary Outcome					
	Alt □-△	Alt rd-bl	All diff	Pure rd-bl	Pure □-△	Other
Humans	2	2	5	2	0	1
Humans (fp)	7	2	1	0	2	0
MEGA-S++	1.5	0.5	4.5	2	0	3.5
MEGA-S++ (f)	3.5	0	3	2	1.5	2
MEGA-S++ (fp)	9	0	0	1	2	0

Table 3: The number of subjects that reached each outcome by 50 rounds. f - feedback cheap talk only, fp – both feedback and planning cheap talk.

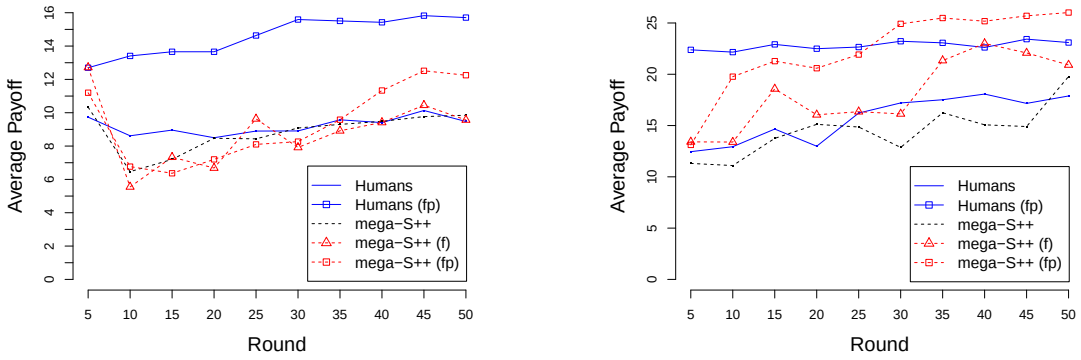
In the SGPD, combined feedback and planning cheap talk led to higher average payoffs after round 35 (Figure 5a). In the Block Game (Figure 5b), feedback and planning cheap talk produced substantially higher payoffs throughout the game, even exceeding the payoffs achieved by Humans in the last 25 rounds on average. Feedback cheap talk alone, however, produced no discernible increase in payoffs in the SGPD, and less substantial increases in the Block Game.

Convergence characteristics provide further insight into how cheap talk impacted MEGA-S++'s ability to learn to interact with people. While feedback cheap talk alone produced little increase in the number of highly collaborative outcomes, feedback and planning cheap talk together led to substantial increases in profitable collaborations (Table 3). In the SGPD, about 80% of all participants learned to cooperate (*Both B*) when paired with MEGA-S++(fp), and 9 out of 12 participants converged to the most collaborative solution in the Block Game. These results are similar to those observed in human-human pairings.

The participants in our study also held much higher opinions of MEGA-S++ when it produced both feedback and planning cheap talk. In the post-experiment questionnaires, participants were asked to indicate how likable and how intelligent their associate was on the scale 1 to 5. Cheap talk had a statistically significant impact on both of these ratings ($F(2, 66) = 6.99; p = 0.002$ and $F(2, 66) = 8.19; p < 0.001$, respectably). Pairwise comparisons show that participants thought the robot was more intelligent when it employed feedback and planning cheap talk than when it employed just feedback cheap talk ($p = 0.005$) or no cheap talk at all ($p = 0.001$). However, when the robot only produced feedback cheap talk, it was not seen as more intelligent than when the algorithm produced no cheap talk ($p = 0.912$).

6. CONCLUSIONS AND DISCUSSION

In this paper, we studied (via user study) how robots can learn to collaborate with human partners when the goals of the robot and the human partner are not fully aligned. Our results indicate that feedback and planning cheap talk



(a) SGPD (b) Block Game

Figure 5: Average payoffs when paired with humans when communication was permitted. (f) indicates feedback cheap talk, (fp) indicates both feedback and planning cheap talk.

can substantially improve a robot's ability to learn to interact with a human partner. An online-learning algorithm that can quickly learn collaborative solutions is insufficient. Humans appear to require communication in the form of forward planning to establish collaborations with an adapting robot. This highlights the need to address AI and HRI synergistically rather than independently.

This work highlights a number of unsolved challenges. For example, in our study, the robot produced cheap talk, but it did not consider what the human said. We anticipate that the ability to engage in two-way communication would substantially improve a robot's ability to learn to collaborate with a human partner. Future work involves determining how cheap talk from the human can be utilized by a learning algorithm to improve its ability to collaborate with people.

7. REFERENCES

[1] R. J. Aumann and S. Hart. Long cheap talk. *Econometrica*, 71 (6):1619–1660, 2003.

[2] R. Axelrod. *The Evolution of Cooperation*. Basic Books, 1984.

[3] O. Bonroy, A. Garapin, and D. Llerena. Repeated cheap talk, imperfect monitoring and punishment behavior: An experimental analysis. *Working Paper GAEL: 2011-2012*, 2012.

[4] C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of AAAI*, pages 746–752, 1998.

[5] J. W. Crandall. Towards minimizing disappointment in repeated games. *Journal of Artificial Intelligence Research*, 49:111–142, 2014.

[6] J. W. Crandall. Robust learning in repeated stochastic games using meta-gaming. In *Proceedings of IJCAI*, 2015.

[7] J. W. Crandall, A. Ahmed, and M. A. Goodrich. Learning in repeated games with minimal information: The effects of learning bias. In *Proceedings of AAAI*, pages 650–656, 2011.

[8] J. W. Crandall and M. A. Goodrich. Learning to compete, coordinate, and cooperate in repeated games using reinforcement learning. *Machine Learning*, 82(3):281–314, 2011.

[9] V. P. Crawford and J. Sobel. Strategic information transmission. *Econometrica*, 50 (6):1431–1451, 1982.

[10] J. Farrell and M. Rabin. Cheap talk. *Journal of Economic Perspectives*, 10 (3):103–118, 1996.

[11] D. Fudenberg and D. K. Levine. *The Theory of Learning in Games*. The MIT Press, 1998.

[12] M. A. Goodrich, J. W. Crandall, and J. R. Stimpson. Neglect tolerant teaming: Issues and dilemmas. In *AAAI Spring Symposium*, 2003.

[13] G. Hoffman and C. Breazeal. Achieving fluency through perceptual-symbol practice in human-robot collaboration. In *Proceedings of HRI*, pages 1–8, 2008.

[14] F. Ishowo-Oloko, J. W. Crandall, M. Cebrian, S. Abdallah, and I. Rahwan. Learning in repeated games: Human versus machine. *arXiv:1404.4985*, 2014.

[15] M. Johanson, N. Bard, M. Lanctot, R. Gibson, and M. Bowling. Efficient Nash equilibrium approximation through Monte Carlo counterfactual regret minimization. In *Proceedings of AAMAS*, pages 837–846, 2012.

[16] R. Karandikar, D. Mookherjee, D. R., and F. Vega-Redondo. Evolving aspirations and cooperation. *Journal of Economic Theory*, 80:292–331, 1998.

[17] J. Y. Kim. Cheap talk and reputation in repeated pretrial negotiation. *RAND Journal of Economics*, 27 (4):787–802, 1996.

[18] M. L. Littman and P. Stone. A polynomial-time Nash equilibrium algorithm for repeated games. *Decision Support Systems*, 39:55–66, 2005.

[19] S. Nikolaidis, K. Gu, R. Ramakrishnan, and J. Shah. Efficient model learning for human-robot collaborative tasks. *arXiv:1405.6341v1*, 2014.

[20] G. A. Rummery and M. Niranjan. On-line Q-learning using connectionist sytems. Technical Report CUED/F-INFENG-TR 166, Cambridge University, 1994.

[21] J. R. Stimpson, M. A. Goodrich, and L. C. Walters. Satisficing and learning cooperation in the prisoner's dilemma. In *Proc. of IJCAI*, pages 535–544, 2001.

[22] A. L. Thomaz and C. Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172:716–737, 2008.