

HUMAN-ROBOT INTERACTION

Teaching robots social autonomy from in situ human guidance

Emmanuel Senft^{1*}, Séverin Lemaignan², Paul E. Baxter³, Madeleine Bartlett¹, Tony Belpaeme^{1,4}

Striking the right balance between robot autonomy and human control is a core challenge in social robotics, in both technical and ethical terms. On the one hand, extended robot autonomy offers the potential for increased human productivity and for the off-loading of physical and cognitive tasks. On the other hand, making the most of human technical and social expertise, as well as maintaining accountability, is highly desirable. This is particularly relevant in domains such as medical therapy and education, where social robots hold substantial promise, but where there is a high cost to poorly performing autonomous systems, compounded by ethical concerns. We present a field study in which we evaluate SPARC (supervised progressively autonomous robot competencies), an innovative approach addressing this challenge whereby a robot progressively learns appropriate autonomous behavior from in situ human demonstrations and guidance. Using online machine learning techniques, we demonstrate that the robot could effectively acquire legible and congruent social policies in a high-dimensional child-tutoring situation needing only a limited number of demonstrations while preserving human supervision whenever desirable. By exploiting human expertise, our technique enables rapid learning of autonomous social and domain-specific policies in complex and nondeterministic environments. Last, we underline the generic properties of SPARC and discuss how this paradigm is relevant to a broad range of difficult human-robot interaction scenarios.

INTRODUCTION

In sensitive domains where social robots are expected to play a key role, such as education and therapy, the question of empowering the human user by allowing them to supervise and retain transparent control over the robot has to be constantly balanced with the contradictory expectation of an advanced level of robot autonomy. In addition, the growing expectation is that robots should behave autonomously not only at a technical, task-specific level but also in terms of social interactions.

Here, we look at one specific, yet difficult, instance of this problem: how domain experts (hereafter called human teachers) can transfer both technical and social skills to enable robots to successfully and autonomously interact with children in an educational task. The expectation is that a robot can gradually learn an adequate social behavior by observing the human teacher and will become increasingly autonomous in both task-level skills and social interactions. As the teacher starts to trust the robot's behavior, they will progressively shift their workload to the robot. In such a scenario, the robot's technical and social policies are coconstructed by the teacher during the learning phase, and the resulting (autonomous) robot behavior thus remains essentially transparent, predictable, and trustworthy to the human teacher (1). Educational social robotics is a prototypical application in this regard: To be an effective educational support, the robot needs to exhibit satisfactory technical (didactic, i.e., subject knowledge) and social (pedagogic behavior) skills, all while preserving the ability for a school teacher to oversee and, if needed, override the robot's behavior.

Learning autonomy instead of programming autonomy

Learning social policies for interactions with humans brings specific requirements not usually considered in machine learning:

R1. The robot has to exhibit, at all times, acceptable (socially and physically safe)—if not perfectly appropriate—social and task-related behavior. This must start from the onset of the learning/interaction.

R2. The robot needs to learn quickly because gathering data points from interactions with humans is a slow and costly process.

R3. To be effective in real-world scenarios, where the human experts teaching the robot are not roboticists, the learning process must be practical, integrate well with the natural human routines, and require limited technical expertise.

Traditionally, two main methods exist for teaching robots, reinforcement learning (RL) (2) and learning from demonstrations (3, 4). One of the core mechanisms of RL is the combination of exploration and learning from errors. By directly interacting with their environment and receiving feedback from it, RL agents learn online. To be effective, this requires both exploration and error recovery to be fast and cheap; thus, RL approaches typically rely on simulators to train the agent. Simulation is, however, often not an option for human-robot interaction (HRI) because simulators fail to reproduce, at meaningful levels, the complexity and unpredictability of human behaviors. This means that the robot should be trained in the real world by interacting with humans. Exploring and recovering from errors in the real world, however, are expensive and sometimes not possible. Not being able to fully recover from errors in HRI is the norm rather than the exception: HRIs almost always require a level of trust, so when the human loses trust in the robot because of poor behavior, the interaction breaks down and often cannot be recovered (5). The risk of such failures limits the general applicability of classical RL to HRI (because these failures would violate R1). In addition, learning with RL is often a slow process, thus also violating R2.

To mitigate these limitations, robots can learn from humans, which ensures that the robot's policy is appropriate to the current application during the learning process. Learning from demonstration (3, 4) is one classical approach that enables humans to teach skills to robots. However, it typically looks at kinesthetic demonstrations (6) in deterministic environments [such as manufacturing, industrial

Copyright © 2019
The Authors, some
rights reserved;
exclusive licensee
American Association
for the Advancement
of Science. No claim
to original U.S.
Government Works

Downloaded from <http://robotics.sciencemag.org/> by guest on October 24, 2019

¹Centre for Robotics and Neural Systems, University of Plymouth, Plymouth, UK.
²Bristol Robotics Laboratory, University of the West of England, Bristol, UK. ³L-CAS,
University of Lincoln, Lincoln, UK. ⁴ID Lab—imec, Ghent University, Ghent, Belgium.
*Corresponding author. Email: senft.emmanuel@gmail.com

robotics, or cobotics (3)], where the human teacher usually relinquishes control and supervision of the robot once the physical skill is deemed to have been acquired by the robot. Beyond manipulation, learning from demonstration has been applied in a few instances to the learning of scheduled tasks (7) and social, interactive behaviors. Two main methods have explored how to learn social behavior from humans: (i) by collecting data from human-human interactions and applying machine learning to derive an autonomous behavior (8–11) and (ii) by using the Wizard-of-Oz (12) method to control a robot in interactions to collect data, which are later used to create an autonomous behavior (13–16). These approaches might lead to an autonomous robot; however, in both cases, researchers approach the learning problem as gathering a static dataset and applying offline learning algorithms to create a static policy. These processes, by separating the demonstrations and the learning, are also rigid and would require substantial technical efforts to update a policy with new data points. In addition, even if the demonstrations are collected from domain experts, they are later analyzed by technical experts. This reliance on technical experts to interpret demonstration data and to create learning algorithms adapted to each environment limits the usability of such approaches for naïve users.

An alternative way is to move away from optimizing a function on a dataset to actively teaching the agent a policy. One such framework is interactive machine learning (IML) (17, 18). IML involves the end user in the learning loop and has the agent learn an appropriate behavior online through a series of small improvements. The end user becomes a teacher and can, for example, provide rewards for the robot's actions, similarly to classic RL (19). The active involvement of the teacher improves the learning (both in speed and quality) and at same time allows them to create a mental model of the robot, increasing the transparency of the robot behavior and the trust the user has in the agent (20, 21). Teachers can also be given more control over the robot by dynamically providing demonstrations, corrections, or additional information to the algorithm to improve the learning even further (22, 23). That way, teachers can even correct errors made by the algorithm before they propagate to the real world. Although promising, there are very few demonstrations of IML applied to learning for social interactions with humans (24, 25). IML, and interactive RL in particular, have had limited success so far and mostly in simple, low-dimensional and deterministic interaction domains (20, 26).

Because no learning method so far addresses the three requirements stated previously, in (27), we introduced SPARC (Supervised Progressively Autonomous Robot Competencies), an interactive framework whereby a robot interacts directly with the environment under the supervision of a human teacher who has complete control over the robot's behavior. With SPARC, initially, the robot's controller is a blank slate. The robot does not act on its own and is only teleoperated by the human teacher in a Wizard-of-Oz fashion—the teacher can select actions that the robot then executes (12). However, as soon as the teacher starts selecting actions, the robot learns from these demonstrations and uses this evolving policy to suggest actions to the teacher. The teacher can confirm or override the robot's suggestions, and this feedback is fed to the learning algorithm to progressively refine the policy. To reduce the teacher's workload, actions proposed by the robot and not cancelled by the teacher are assumed to be acceptable and are executed after a short delay. This mechanism aims to limit the need for human intervention. The teacher only has to demonstrate actions and prevent incorrect actions from being executed. Thus, as

the robot's behavior improves, the robot proposes correct actions more often, reducing the need for demonstrations and corrections and thereby the amount of input required from the teacher to achieve an effective behavior, in a process bearing similarity to the ML processes behind predictive texting (28). The novelty of SPARC lies in the in situ component of the learning: The robot learns online and in the real world, which was often not the case in prior work. When applied to HRI, for example, in the context of education, this translates into transforming a dyadic interaction (human teacher, learning child) into a triadic interaction (human teacher, robot, child), where the teacher teaches the robot how to support the child's learning on the go (Fig. 1).

SPARC was introduced in (27); however, it had never been tested to teach robots to interact with people. Previous research only considered scenarios where the robot was interacting either in a simulated environment (26) or with another robot simulating a human (27). This paper aims to evaluate SPARC in a real HRI, taking as context tutoring for children. The conceptual simplicity of the paradigm and its agnosticism with regards to the actual learning algorithm make it widely applicable to a range of social HRIs beyond the specific educational scenario that we used as support in this article.

Case study: Robots as tutors for children

Social robots have been explored as educational tools in the last decade. Because of increases in the number of pupils in the classroom and budget constraints (29), one-to-one interactions between teachers and students, known to be highly beneficial, are limited. One solution is to use a robot to supplement the teacher to offer additional individualized support to students. Recent studies have shown that social robots are typically more effective than alternative, disembodied technologies, such as tutoring software presented on a tablet or computer. The physical presence of the robot together with its social appearance fosters interactions with the learner, including increased attention and compliance, which are conducive to learning (30). However, their general lack of appropriate integration to the classroom ecosystem and to teacher's practices leads to poor adoption rates by schools (31). Having a robot that can be operated initially by the teacher but then gradually takes over control would offer a tutoring experience that is better tailored to the particular learner or context.

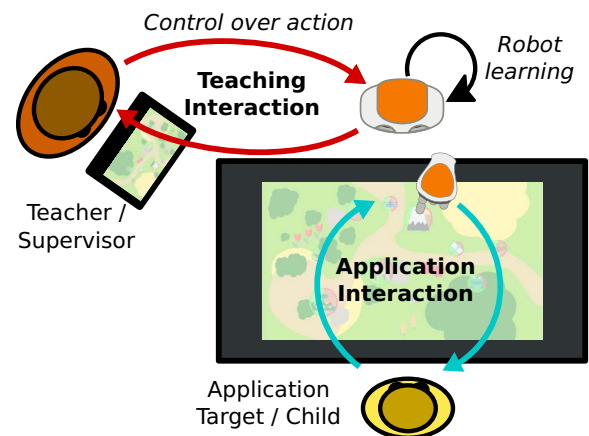


Fig. 1. Diagram of the application of SPARC to HRI. A human teacher supervises a robot learning to interact with another human (e.g., a child in the context of education).

RESULTS

Study introduction

We present a study evaluating SPARC in a high-dimensional social task where 8- to 10-year-old children learned about food webs through playing an educational game (Fig. 2). In this game, 10 animals can be moved around in a touchscreen-based game environment; animals have energy and have to consume plants or other animals to stay alive. Children have to keep the ecosystem viable as long as possible. The role of the robot tutor was to guide the child through providing advice (such as keeping track of the animals' energy or indicating what animals eat) and social prompts (e.g., encouraging the child). The game logic and the tutoring interaction were jointly modeled as an optimization problem with 210 continuous input values (last actions, distances between animals, etc.) and 655 potential output actions (motions, gestures, verbal encouragements, etc.).

The interaction consisted of four consecutive and independent game rounds and knowledge tests before the first round, between the second and the third, and after the fourth.

Our protocol included three conditions designed to assess the impact of applying the proposed approach (SPARC) to this task. The control condition (passive condition) used a passive robot that only provided initial instructions and guidelines and did not offer support during the learning game. The second, the supervised condition, involved a robot that gradually learned, from human demonstration, how to provide support during the game by using SPARC. In this condition, the robot's controller evolved with each interaction with the participants (refining its suggestions to the teacher over time). Nevertheless, the control provided to the teacher through SPARC ensured that the robot's behavior was consistent for all participants and supported their inclusion as a single group for this condition. The third, the autonomous condition, used an autonomous robot that executed the policy learned in the supervised condition but without ongoing supervision.

We ran the autonomous condition at the conclusion of the supervised condition, and the passive condition was run in parallel of the two other conditions. This allowed the trained policy learned in the supervised condition to be used in the autonomous condition. Con-



Fig. 2. Setup used in the study. A child interacts with the robot tutor with a large touchscreen sitting between them, displaying the learning activity; a human teacher provides guidance to the robot through a tablet and monitors the robot's learning. Although the picture depicts an early laboratory pilot, the main study was conducted on actual school premises.

sequently, this study was set up as a between-subject design, with a random selection of a child for each interaction.

In the supervised condition, a single person, naive about the learning mechanism and the hypotheses tested in the study, acted as a teacher for the robot in all the interactions. With 75 children in total ($n = 75$; age: $M = 9.4$, $SD = 0.71$; 37 female), each of the three conditions was allocated 25 children.

Hypotheses

Two hypotheses were explored:

H1. The autonomous robot learns a policy that produces behavior similar to that of the teacher. We hypothesized that the policies of the autonomous and supervised robots will present similarities in terms of frequency and timings of actions and that both will have a positive impact on the children compared with no behavior.

H1a. The autonomous robot will only use actions already demonstrated by the teacher, and there will be no difference in the frequency of use of each type of action between the supervised and autonomous robots.

H1b. In the teacher's policy, each type of action will have a unique dynamics (i.e., when the action is triggered). The robot will learn such dynamics, and there will be no difference in timing for each type of action between the supervised and autonomous robots.

H1c. Both robots (supervised and autonomous) will have similar and positive effects on the children: Interactions metrics and learning gains will present no differences between the supervised and autonomous robots, and both the teacher and our learning algorithm will produce robot behaviors that will lead to better results on these metrics than no behavior (e.g., a passive robot).

H2. Using SPARC, the teacher's workload decreases over time. The amount of input required from the teacher will decrease over time, and the robot's suggestions will be deemed acceptable more often (increase of accepted suggestions and decrease of the rejected suggestions).

In our protocol, the same teacher was responsible for the whole training of the robot as it was interacting with 25 children, which ensured a consistent delivery style for all participants. It would be insightful to try the same protocol with other teachers.

Example of a session

Table 1 presents an example of the first minute of a round, with suggestions by the robot and actions from the teacher. For example, at $t = 16.9$ s, the teacher accepted the suggestion by the robot. Alternatively, in some cases, such as the suggestion at $t = 20.6$ s, the teacher did not accept the action suggested by the robot and selected another action. In that case, the suggested action was not considered, and only the selected action was executed and used for learning. Last, at $t = 44.4$ s, the teacher selected the action to move the mouse closer to the wheat, and after the robot moved the mouse, the child tried other animals and then fed the mouse with the wheat. This demonstrates how actions from the robot could help the children to discover new connections between animals. As shown by this table, the teacher was able to select actions and react appropriately to the robot's suggested actions.

Policy comparison

Figure 3A presents the number of actions of each type executed by the supervised robot (in the supervised condition) and by the autonomous robot (in the autonomous condition). The first observation is that the autonomous robot based its actions on the teacher's demonstrations: The action "move away" (whereby the robot moves one animal away

Table 1. Example of events during the first minute of the first round of the interaction with the 23rd child in the supervised condition. Events beginning with “robot” represent suggestions from the robot; events beginning with “teacher” are the reactions from the teacher. “mvc” is the abbreviation of the move close action, and times are provided in seconds. Words in *italics* refer to items on the screen and number (if applicable) of the specific item interacting.

Time	Event	Time	Event
4.1	Childtouch <i>frog</i>	32.5	Childrelease <i>dragonfly</i>
4.3	Failinteraction <i>frog wheat-3</i>	34.4	Childtouch <i>wolf</i>
4.9	Animaleats <i>frog fly</i>	34.7	Robot proposes remind rules
5.8	Childrelease <i>frog</i>	35	Animaleats <i>wolf mouse</i>
6.6	Robot proposes congrats	36	Teacher selects wait
7.6	Childtouch <i>fly</i>	36	Animaleats <i>wolf mouse</i>
7.6	Teacher selects wait	37.2	Childrelease <i>wolf</i>
8	Animaleats <i>fly apple-4</i>	37.7	Childtouch <i>grasshopper</i>
8.3	Childrelease <i>fly</i>	38.3	Robot proposes congrats
9.1	Teacher selects congrats	42.1	Failinteraction <i>grasshopper apple-1</i>
9.1	Childtouch <i>frog</i>	42.7	Childrelease <i>grasshopper</i>
10.3	Childrelease <i>frog</i>	42.7	Failinteraction <i>grasshopper apple-1</i>
10.8	Childtouch <i>frog</i>	44.4	Teacher selects instead mvc <i>mouse-wheat-1</i>
11.2	Animaleats <i>frog fly</i>	44.6	Robottouch <i>mouse</i>
12.4	Failinteraction <i>frog apple-2</i>	44.7	Childtouch <i>butterfly</i>
12.5	Animaleats <i>frog fly</i>	45.1	Failinteraction <i>butterfly wheat-2</i>
13.2	Childrelease <i>frog</i>	45.6	Childrelease <i>wheat-1</i>
14.2	Childtouch <i>fly</i>	45.6	Roborelease <i>mouse</i>
14.5	Animaleats <i>fly apple-2</i>	45.7	Robottouch <i>mouse</i>
14.6	Robot proposes encouragement	48.9	Roborelease <i>mouse</i>
15	Childrelease <i>fly</i>	49.3	Childtouch <i>butterfly</i>
15.4	Animaleats <i>fly apple-3</i>	49.3	Failinteraction <i>butterfly wheat-1</i>
16.9	Teacher confirms encouragement	49.6	Childrelease <i>butterfly</i>
18.2	Childtouch <i>snake</i>	50	Childtouch <i>mouse</i>
18.4	Failinteraction <i>snake wheat-3</i>	50.3	Animaleats <i>mouse wheat-1</i>
18.7	Animaleats <i>snake bird</i>	51	Childrelease <i>mouse</i>
19.6	Animaleats <i>snake bird</i>	51.1	Animaleats <i>mouse wheat-2</i>
20.5	Childrelease <i>snake</i>	51.4	Robot proposes congrats
20.6	Failinteraction <i>snake wheat-4</i>	52.3	Teacher confirms congrats
20.6	Robot proposes congrats	52.9	Childtouch <i>snake</i>
20.9	Childtouch <i>eagle</i>	52.9	Failinteraction <i>snake wheat-3</i>
21.1	Animaleats <i>eagle bird</i>	53.2	Childrelease <i>snake</i>
22	Animaleats <i>eagle bird</i>	53.5	Childtouch <i>mouse</i>
22.4	Childrelease <i>eagle</i>	53.6	Animaleats <i>mouse wheat-3</i>
23.3	Animaldead <i>bird</i>	54.4	Robot proposes congrats

continued on next page

Time	Event	Time	Event
23.4	Teacher selects instead mvc <i>dragonfly - fly</i>	54.5	Animaleats <i>mouse wheat-4</i>
23.6	Robottouch <i>dragonfly</i>	55	Childrelease <i>mouse</i>
26.9	Robotrelease <i>dragonfly</i>	55.6	Childtouch <i>dragonfly</i>
27.7	Childtouch <i>fly</i>	56.1	Teacher selects wait
28	Childrelease <i>fly</i>	56.8	Failinteraction <i>dragonfly apple-1</i>
28.4	Childtouch <i>dragonfly</i>	57.3	Childrelease <i>dragonfly</i>
28.6	Failinteraction <i>dragonfly apple-1</i>	57.5	Failinteraction <i>dragonfly apple-1</i>
29.1	Childrelease <i>dragonfly</i>	58.6	Childtouch <i>grasshopper</i>
29.4	Failinteraction <i>dragonfly apple-1</i>	58.6	Failinteraction <i>grasshopper apple-1</i>
30.3	Childtouch <i>dragonfly</i>	58.8	Childrelease undefined
30.3	Failinteraction <i>dragonfly apple-1</i>	59.1	Childtouch <i>dragonfly</i>
30.7	Robot proposes encouragement	59.1	Failinteraction <i>dragonfly apple-1</i>
31	Failinteraction <i>dragonfly apple-1</i>	59.2	Failinteraction <i>grasshopper apple-1</i>
31.8	Teacher selects wait	59.9	Failinteraction <i>dragonfly apple-1</i>

from a prey, typically to indicate the pair is unsuitable) was almost never used, “move to” was never used (“move close” was used instead, as to hint an animal-food pair to the child), and the supportive feedback (“congratulation” and “encouragement”) was used more often than “remind rules” or “drawing attention.” This provides support for H1a. However, the number of times each action was executed for autonomous and supervised conditions was different (Bayesian t test: congratulation, $BF_{10} = 37.8$; encouragement, $BF_{10} = 5.1 \times 10^4$; drawing attention, $BF_{10} = 0.53$; remind rules, $BF_{10} = 1.6 \times 10^3$; and move close, $BF_{10} = 21.7$), failing to provide full support for H1a. These differences of action frequencies are probably linked to the type of machine learning used; with instance-based learning, some data points will be used in the action selection much more often than others, which might explain these biases.

In addition, Fig. 3B shows the time between each action executed by the robot and the last eating event (when the child fed an animal). For both conditions, there were significant differences between the times since the last eating event for each type of action [Bayesian analysis of variance (ANOVA), supervised condition: $F(4,1211) = 101$, $P < 0.001$, $B_{10} = 1.06 \times 10^{71}$; post hoc analysis in table S1—only encouragement and remind rules seem to present similarities—autonomous condition: $F(4,1385) = 81.0$, $P < 0.001$, $B_{10} = 1.53 \times 10^{58}$; post hoc analysis in table S2], providing initial support to H1b. Furthermore, we found no differences when comparing the timing for each type of action between conditions (Bayesian t test between conditions: congratulation, $BF_{10} = 0.20$; encouragement, $BF_{10} = 0.21$; remind rules, $BF_{10} = 0.13$; drawing attention, $BF_{10} = 0.21$; and move close, $BF_{10} = 0.15$), providing additional support for H1b. This means that the autonomous robot managed to capture the uniqueness of timing for each action and applied a policy using the unique timing used by the teacher. Together, these results show that the robot managed to learn social and technical policies, including their associated dynamics, that are similar to the ones demonstrated by the teacher.

Learning gains

A positive learning effect, as measured through normalized learning gain (32), was apparent in both the passive condition [$M = 0.12$; 95% confidence interval (CI), 0.07 to 0.18] and the supervised condition ($M = 0.11$; 95% CI, 0.06 to 0.16), with the performance in the autonomous condition slightly exceeding these ($M = 0.14$; 95% CI, 0.09 to 0.19). However, the robot’s behavior during the game did not have a meaningful impact on the children’s learning gain [Bayesian ANOVA: $F(2,72) = 0.34$, $P = 0.72$, $B_{10} = 0.15$], failing to provide initial support for H1c.

Game metrics

Multiple game metrics have been collected in the rounds of the game played by the children, and they can inform us of the effect of the robot’s behavior on the children during the game sessions. Figure 4A and table S3 show the evolution of the total number of different “learning units” (i.e., in our food-chain scenario, one new and correct attempt to feed one animal with one type of food) encountered by the children across the four game rounds. A Bayesian mixed ANOVA showed an impact of the repetition (i.e., progress in the rounds of the game) and the condition on the number of different eating interactions produced by the children in the game [Bayesian mixed ANOVA: repetition, $F(3,216) = 6.75$, $P < .001$, $B_{10} = 77.7$; condition, $F(2,72) = 5.19$, $P < 0.01$, $B_{10} = 5.76$]. With additional rounds of the games, the children successfully connected more animals together. Post hoc tests showed no significant difference between the supervised and the autonomous conditions (Bayesian repeated-measures ANOVA, $B_{10} = 0.15$), whereas differences were observed between the supervised and the passive conditions ($B_{10} = 512$) and between the autonomous and the passive conditions ($B_{10} = 246$). This indicates that, compared with the passive robot, the supervised robot provided additional knowledge to the children during the game, allowing them to create more useful interactions between animals and their food, receiving more information from the game and thus potentially helping them to get knowledge about

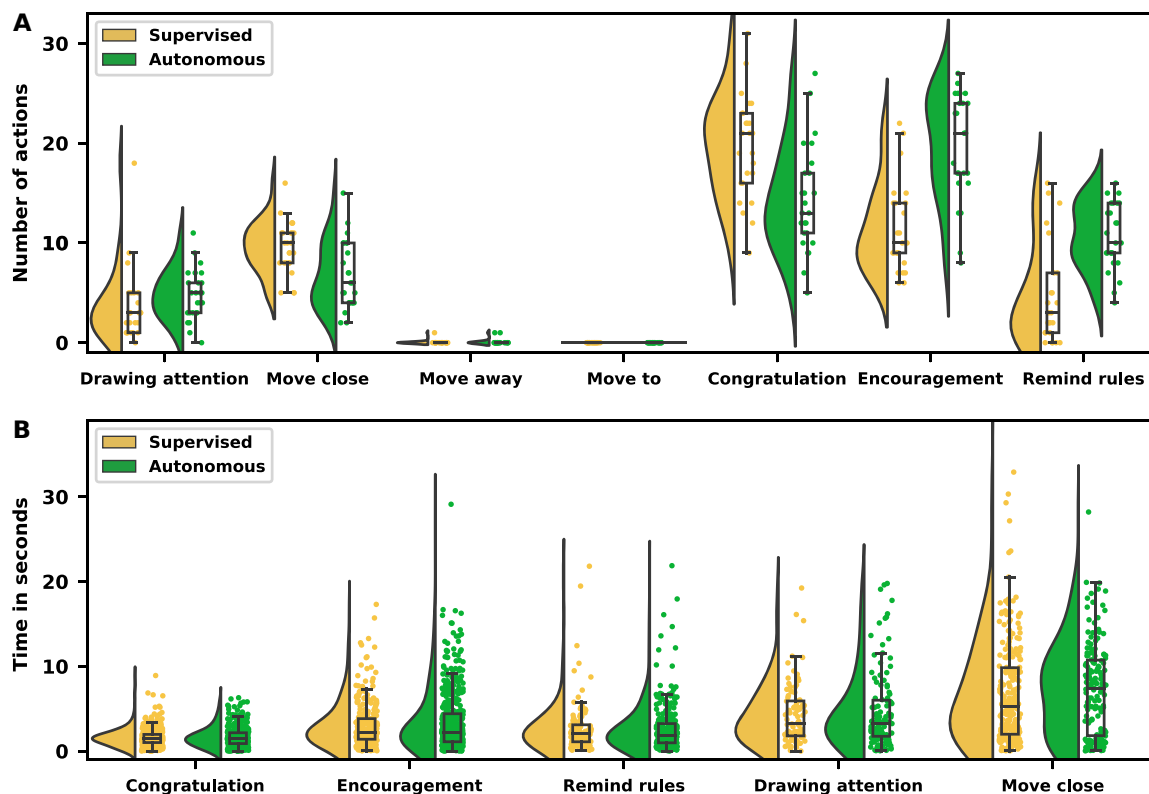


Fig. 3. Comparison of policy between the supervised and autonomous robot. (A) Comparison of the number of actions of each type executed by the robot in the autonomous and supervised conditions. Each point represents how often the robot executed an action with a child ($n = 25$ per condition). (B) Timing between each action and the last eating event (due to their low or null number of execution, the actions “move to” and “move away” were not analyzed). Each point represents one execution of an action.

what animals eat. The autonomous robot managed to recreate this effect without the presence of a human in the action selection loop.

Figure 4B and table S4 show the evolution of game duration across the four game rounds. A Bayesian mixed ANOVA showed inconclusive results on the impact of condition on game duration [Bayesian mixed ANOVA: $F(2,72) = 2.6$, $P = 0.08$, $B_{10} = 1.04$]. Post hoc tests showed no significant difference between the supervised and autonomous conditions (Bayesian repeated measure ANOVA: $B_{10} = 0.29$), whereas differences were observed between the supervised and passive conditions ($B_{10} = 118$) and a trend toward a difference between the autonomous and passive conditions ($B_{10} = 2.90$). These results indicate that children were better at the game in the supervised condition whereby animals were alive longer than in the passive condition. The autonomous robot learned and applied a policy tending to replicate this effect and without exhibiting differences with the supervised one.

However, the analysis showed no effect of the repetitions on game duration [Bayesian mixed ANOVA with Huynh-Feldt correction: $F(2.4,174.9) = 0.31$, $P = 0.78$, $B_{10} = 0.022$]; the children did not manage to keep the animals alive longer with more practice at the game. One of the reasons was a partial ceiling effect at 2.25 min (see the red line on Fig. 4B). When not fed, animals would run out of energy in 2.25 min, so if children did not manage to feed at least seven of the animals at least once before that time, then the game would stop. Because this might prove difficult to identify and achieve, many children did not manage to cross this limit. These game metrics suggest that the supervised robot managed to help the child in the game (compared with a

passive robot) from the onset, and the autonomous robot replicated this effect; thus, these results support H1c.

Teaching the robot

Figure 5 presents the teacher’s reactions to the robot’s suggestions across all the supervised interactions. Contrary to our expectations, the number of accepted and refused suggestions, as well as teacher-initiated actions, stayed roughly constant throughout the interactions with the children. No curve could be significantly fitted using a linear regression [accepted propositions: $R^2 = 0.02$, $F(1.0,23.0) = 0.54$, $P = 0.47$; rejected propositions: $R^2 = 0.09$, $F(1.0,23.0) = 2.18$, $P = 0.15$; and teacher-initiated actions: $R^2 = 0.001$, $F(1.0,23.0) = 0.01$, $P = 0.91$]. We would have expected these results to be different: With the learning, the number of accepted propositions should have increased, and both numbers of refused propositions and teacher-initiated actions should have decreased; thus, H2 is not supported. Note that, however, these results are based on a single teacher and might not be replicated with another teacher.

To provide insights on this result, we analyzed a diary that the teacher completed during the study, noting how the children responded and how she interacted with the robot. From this report and a posttraining interview, the teacher reported that her workload decreased over time, and she mentioned three phases in her teaching (session numbers are indicative, the boundaries were not clear):

First phase (sessions 1 to 3). She was not paying much attention to the suggestions, mostly focusing on having the robot execute a correct

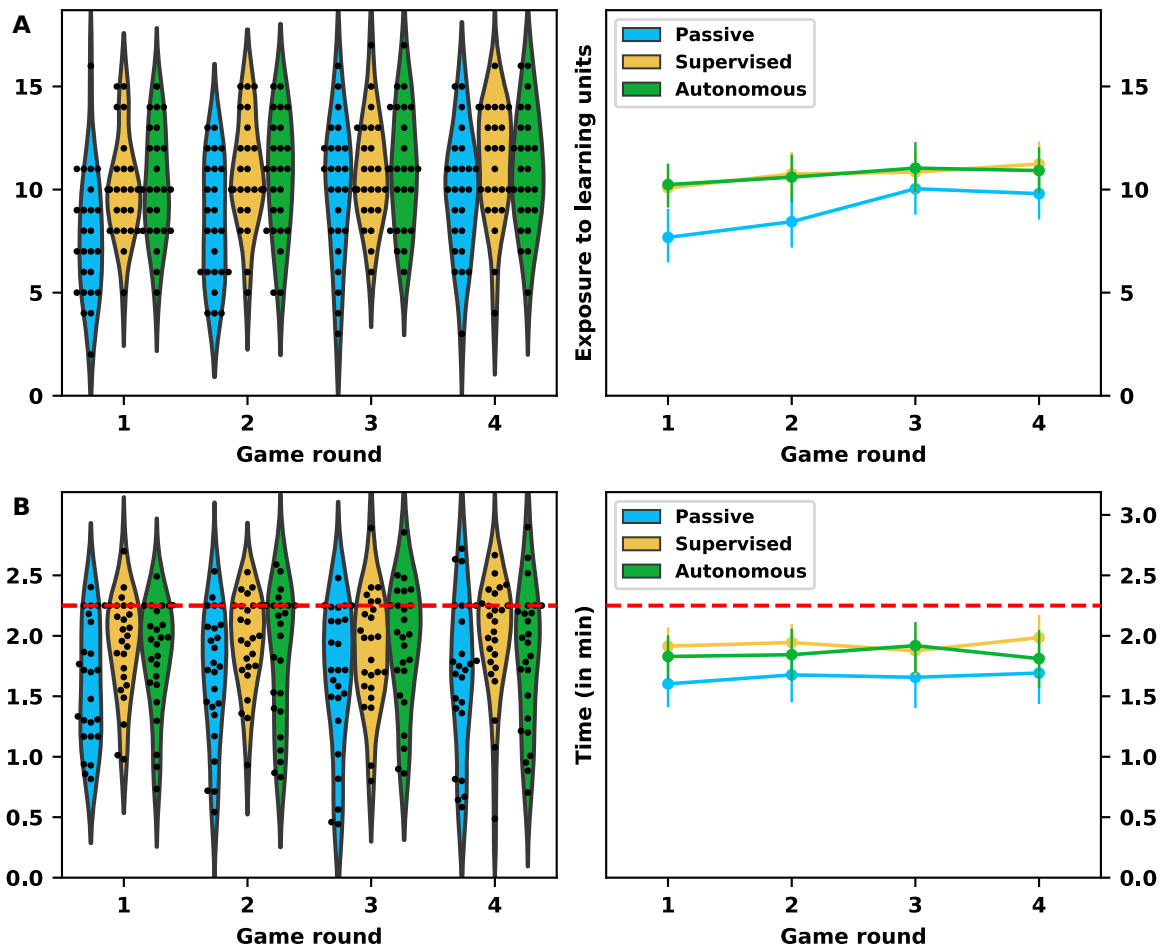


Fig. 4. Comparison of children’s behavior between the three conditions. (A) Number of different eating interactions produced by the children (corresponding to the exposure to learning units) for the four rounds of the game, for the three conditions. (B) Interaction time for the four rounds of the game for the three conditions. The dashed red line represents 2.25 min, the time at which unfed animals died without intervention, leading to an end of the game if the child did not feed animals enough.

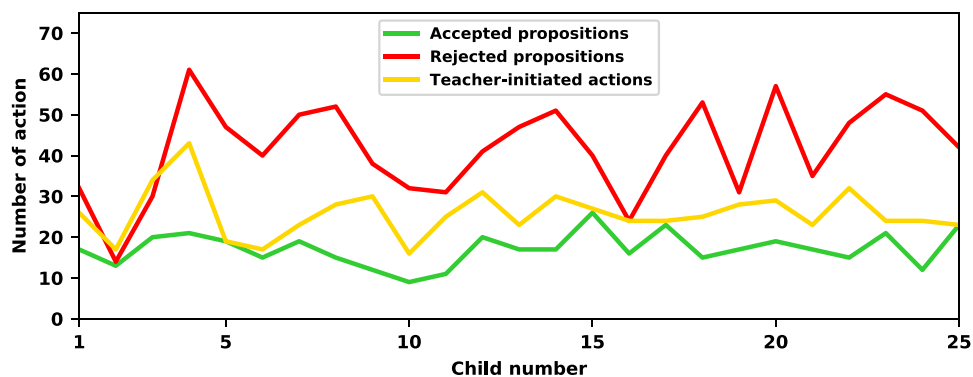


Fig. 5. Summary of the action selection process in the supervised condition. Child number 1 corresponds to the beginning of the training; child number 25 corresponds to the end of the training. The “teacher-initiated actions” label represents each time the teacher manually selected an action not proposed by the robot.

policy: She “found it difficult to know how best to respond” (session 2); “I’m dismissing robot’s suggestion more than I actually want to” (session 3); “I’m skipping/cancelling all in order to avoid inappropriate suggestions” (session 3).

Second phase (sessions 4 to 11). She was paying more attention to the suggestions but without giving them much credit: “achieving

a better balance between my own actions and robot’s suggestions” but “the robot is a bit overwhelming” (session 4); “allowed some robot suggestions but not many as I wanted to slow game-play down” (session 6); “allowing more robot suggestions” (session 7).

Third phase (sessions 12 to 25). She started to trust the robot more but without ever trusting it totally: “Let the robot carry out a lot of its

suggested behaviours” (session 12); “will try to use more robot suggestions as robot was often suggesting good things but I was auto-skipping them” (session 13); “allowed the robot to carry out more of its suggestions” (session 17); “let the robot carry out a lot of suggestions” (session 18).

It appears that the teacher reported a decrease of workload over time (as supported by behaviors such as typing her observations on a laptop, while gazing at the interface at the start of interactions). However, although controlling the robot became easier with practice, we did not observe an increase of accepted actions. Similarly, after having supervised the robot for multiple sessions, the teacher reported: “Controlling the robot is really easy now, although I still tend not to let it carry out its suggested actions even when they are valid.”

DISCUSSION

This study has demonstrated that in a little over 3 hours and only 25 independent interactions, the robot successfully learned social and pedagogical behavior to support children in an educational activity. This learning happened online, using a teacher with no knowledge of the algorithm implementation or intent of the study. Although the autonomous robot used actions with a different frequency than the teacher, it only used actions already demonstrated (partially supporting H1a), it learned the unique dynamics (i.e., timing) associated to each type of action (supporting H1b), and its behavior had a positive impact on the children similar to the supervised robot (partially supporting H1c—no effect was observed on learning gains). However, SPARC did not allow the teacher’s workload to decrease over time (invalidating H2).

In summary, this study demonstrates that the principles behind SPARC allow for an efficient teaching of social autonomy that can be achieved in the real world, on a human time scale, and while maintaining an appropriate robot behavior throughout the teaching and subsequently when the robot interacts autonomously.

Our methodology has two main facets: It learns a social behavior, and it learns in situ (both online and in the real world). We discuss hereafter these three particularities.

Learning online

Learning online offers significant advantages compared with offline learning. First, it allows a human (the teacher) to remain in the learning loop, giving them the opportunity to observe and to influence the evolution of the robot’s behavior. By receiving feedback from the robot, the teacher can estimate the robot’s policy and knowledge level. Involving the end users in the training of the system in this way facilitates an understanding of the resulting behaviors, thus increasing the transparency of complex systems and easing the decision to deploy the robot to interact autonomously.

In addition, learning online provides more flexibility to the learning system. Unlike offline learning (such as learning from demonstration), no engineering skills are required after collecting data to obtain the autonomous behavior. Technical expertise is only required during the design phase of the interaction. This key difference has two impacts. First, it implies that even with a single world representation and learning algorithm, different robot behaviors could be manifested on the basis of the specific knowledge, experience, and preference of different teachers and the specific needs of the current situation. Second, it empowers end users to design their own autonomous robotic controller without requiring technical expertise. Together, these

features might reduce the need for engineers, thus making the process of designing a policy easier and more adaptive and the resulting policy more suited to the user’s needs, potentially helping to democratize the use of robots.

Learning in real-world and sensitive environments

Although the advantages of learning online potentially apply to any IML methods, most of these approaches provide the teacher with only limited control over the behavior executed by the robot. This lack of control cannot ensure that the robot’s behavior will be appropriate and safe for the interaction partners, the robot itself, or its environment, thus reducing the applicability of such methods in sensitive environments (26). Because robots are expected to interact in the real world, directly with humans, it is critical that the learning process uses data from real interactions in the wild, in the environment where they are supposed to take place.

For example, in this study, children displayed a number of unexpected behaviors that the robot had to adapt to (such as intentional waiting, hectic play style, etc.). The robot learned in this ecologically valid (rich, under-specified, stochastic, real-world interaction) and sensitive environment (involving children, a vulnerable population) where incorrect robot behavior could have caused distress, annoyance, and/or reduced learning outcomes. The robot’s task was complex, with an input space of 210 dimensions and output action space of 655 actions. Thus, the learning situation considered in this study was realistic and more challenging than many others where IML has been evaluated [often deterministic environments, with limited risks due to failures (19, 20)] or traditional adaptive scenarios for educational HRI (24, 33).

Despite these challenges, SPARC was successful both in the teaching phase (ensuring that the robot’s behavior was safe and useful from the outset) and in the autonomous phase (by demonstrating a behavior comparable to the teacher’s policy and which had similar impacts on children). By ensuring that the teacher vets each of the robot’s actions before its execution, SPARC increases applicability of IML to sensitive real-world situations.

Learning to be social

Providing robots with social autonomy is still a challenge today. Typically, researchers either have to hard-code behaviors, or the system learns offline from demonstrations. While presenting significant advantages compared with these methods, IML had not yet been convincingly applied to social interaction.

In the specific case of education, we have demonstrated that the robot autonomously reenacted the teacher’s way of supporting the children and reached tutoring results on par with those of a human controlling the robot. The robot learned not only the didactics of the task (the actions relevant to the task) but also some elements of pedagogy, the latent dynamics of the interaction (when actions should be executed). Together, these two facets of the autonomous robot’s policy show that social autonomy can be taught to robots in situ and that SPARC is a powerful method allowing humans to teach robots to interact in social environments.

Outlook

Although our results demonstrated the opportunities provided by SPARC, some limitations remain to motivate future work. This study did not show a decrease of the teacher’s workload over time (as measured by the amount of input by the teacher). As shown in the teacher’s

diary, the main reason for this constant workload was that the robot proposed actions too often, overloading the teacher and sometimes preventing her to take time to correctly evaluate each suggestion. Future work should replicate this study with other teachers and explore ways to provide the teacher with more control not only on the overt robot behavior (the one displayed in the application) but also in the teaching interaction (such as being able to control metaparameters of the learning algorithm).

Although the learned behavior is better than having no behavior at all, it is still possible that a hand-designed or random policy is also not worse than teacher or learned behaviors. In other words, the learned policy is better than no policy at all, but whether it is better than any other policy is unclear. Last, SPARC should also be applied to other domains and in combination with more learning algorithms to properly investigate its ability to generalize.

CONCLUSION

This paper demonstrated the potential for SPARC to enable robots to learn from humans. This capability is especially useful in HRI because knowledge of the desired robot behavior typically comes from domain experts, such as teachers or therapists, rather than roboticists. The standard approach to designing robotic controllers requires multiple conversations between the engineers coding the behavior and the domain experts. Robot learning from end users (e.g., by using SPARC) would bypass these costly iterations, allowing end users to directly teach an efficient controller adapted to their specific needs in a minimally intrusive way. Furthermore, because the process fundamentally relies on having the human in the loop, it also holds considerable potential for sensitive applications of social robots, such as in eHealth, assistive robotics, or education.

The implications of this study are twofold: First, we have demonstrated that, with an appropriate methodology, IML can be successfully applied to transfer human expertise to an autonomous robot in a short period of time and in a high-dimensional and ecologically valid task. Second, we have shown that not only domain-specific technical expertise but also elements of social behaviors (such as timing between events and actions) can be taught in this way.

These two results are important. The dynamic and stochastic nature of social interactions makes learning appropriate and contingent social behaviors a challenge for which classical machine learning approaches are ill suited. We have shown here a path forward, and our approach makes it possible for autonomous social behaviors to be learned in an online manner, gradually taking over the social interaction from the human operator.

MATERIALS AND METHODS

Rational and objectives

The goal of the study is to evaluate whether SPARC can be used to teach online a robot to interact in a complex, nondeterministic, and real environment. In previous studies (27, 26), SPARC was only evaluated in simple environments and not for creating social behaviors. Consequently, this study investigated whether SPARC can be applied to HRI to teach a robot to replicate a policy demonstrated by a human. The goal was not to reach an optimal robot's policy, but one replicating the characteristics of the teacher's, thus demonstrating the potential of SPARC. In this study, a robot guided a child through a gamified tutoring session where the child had to interact with animals on a touchscreen to

learn about food webs. This study compared three conditions where the robot could be either passive (not providing any feedback or information to the child during the game), supervised (an adult, the teacher, was teaching the robot how to support the child during the game), or autonomous (the robot interacted without supervision and executed autonomously the policy learned in the supervised condition).

Apparatus

This study was based on the sandtray paradigm (34): A child interacts with a robot via a large touchscreen located between them. By interacting with the touchscreen and the robot, the child is expected to gain knowledge or improve some skills. Because of its widespread application to HRI and child tutoring (30), we used the NAO robot (35). In addition, a teacher can control and teach the robot in the "supervised" condition using a tablet. This results in a triadic interaction: A human, the teacher, knows how the robot should behave, can control it to execute an efficient behavior, and can teach it how to interact with another human in situ by using SPARC (as shown in Fig. 2).

Participants

Children from five classrooms across two different primary schools in Plymouth (United Kingdom) were recruited to take part in the study. Because both schools had an identical Office for Standards in Education evaluation (indicating that they provide similar educational environments), all the children were combined into a single pool of participants. Full permission to take part in the study and be recorded on video was acquired for all the participants via informed consent from parents. Children with special educational needs interacted with the robot but were excluded from the data collections, as well as children used in pilot versions and sessions where the protocol was breached (e.g., one child dropped out from the passive condition, two from the supervised condition, and zero in the autonomous condition). To manage the number of children available in these classes, we decided to collect data until we reached 25 children per condition. To give every child in the class the opportunity to take part in the study, the remaining children did interact with the robot but were excluded from the data collection. In total, 75 children were included in the final analysis ($n = 75$; age, $M = 9.4$; $SD = 0.72$; 37 female). Because of our protocol, we had to first collect all the participants for the supervised condition before running the autonomous condition; nevertheless, the selection of a child for each interaction was random.

In the supervised condition, the robot's teacher was a psychology PhD student from the University of Plymouth, with limited knowledge of machine learning but with an understanding of human cognition. This teacher is now part of the authors, but at the time of the study, the authorship was not considered, and she was not involved in the study design. Consequently, although knowledgeable about the protocol, she was unaware of the hypotheses tested, and the implementation and had no incentive to bias the results to fit them. The teacher was instructed on how to control the robot using a graphical user interface on the tablet and the effects of each button. She experimented controlling the robot in two interactions (not included in the results analysis) to get used to the interface and controlled the robot. After these interactions, the algorithm was reset, and the teacher started to supervise the robot for the supervised condition. No information about the learning algorithm or the representation of the state and no feedback about the optimal way of interacting or on her policy were provided before or during the study. Hence, this study involved,

as teacher, a naive user not expert in machine learning and more similar to the general population of expected robot users than an expert in computing.

Protocol

At the start of the interaction, the child was first introduced to the robot and told that they would play a game together about the food web (cf. fig. S1A). They completed a quick demographic questionnaire and a first pretest to evaluate their baseline knowledge (cf. fig. S1, B to E). After this test and before starting the game, the child completed a tutorial where they were introduced to the mechanics of the game: Animals have energy and have to eat to survive, and the child can move animals to consume other animals or plants to replenish their energy (cf. fig. S1, F and G). The teacher was sitting with the child through these steps to provide clarification if needed and was following a script. After this short tutorial, the teacher sat away from the child to supervise the robot if required. For ethical reasons, for all children, the teacher and an additional experimenter were present in the room but out of view of the children while maintaining an attitude of disinterest. The child then completed two rounds of the game where the robot could provide feedback and advice depending on the condition they were in (cf. fig. S1, H to K). Afterward, the child completed a

midtest before playing another two rounds of the game and completing a last posttest to conclude the study. Figure S1 shows examples of the screen throughout the interaction.

Implementation

The robot was controlled using the architecture presented in Fig. 6 with all the nodes communicating together using the Robot Operating System (36). The teacher interface ran on a separate tablet and was used only for the supervised condition. All the other nodes ran on the large touchscreen computer displaying the game interface that was used to guide the child through the study and to present the game rounds and the tests. The default robot behavior was simply reading the instruction on the screen, following the child’s face, and swaying lightly.

To support the children during the game rounds, the robot has access to 655 actions consisting of moving animals in relation to others on the screen (by pointing to an object and moving it on the screen), asking the child to focus on some items of the game (by pointing to them and uttering a predefined sentence), and providing social prompts and feedback such as reminding them of the rules and providing encouragements or congratulations. The robot’s policy in the game consisted of a mapping between these actions and a representation of the

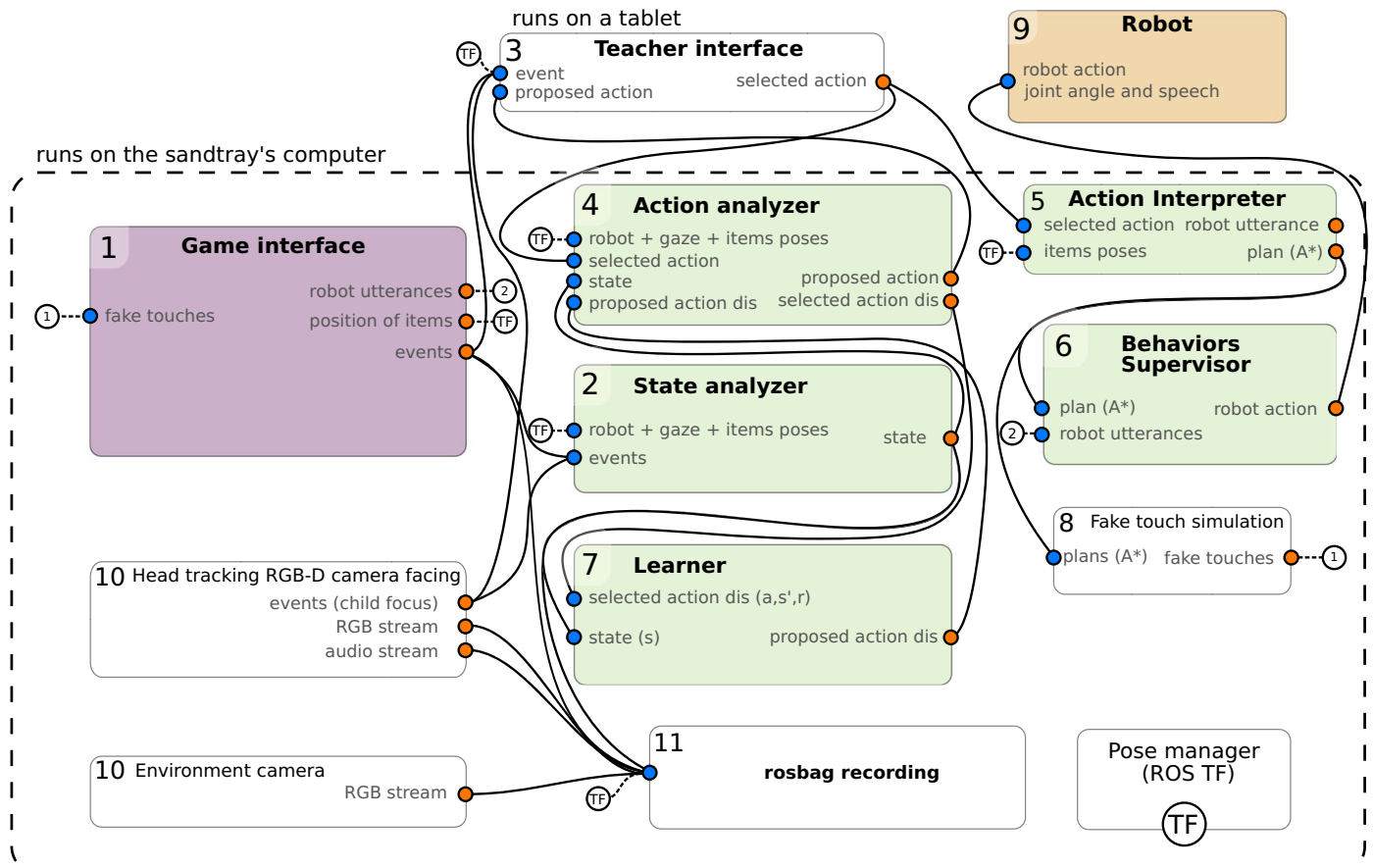


Fig. 6. Simplified schematics of the architecture used to control the robot. A game (1) runs on a touchscreen between the child and the robot. (2) analyzes the state of the game using inputs from the game and the camera. (3) is an interface running on a tablet and used by the teacher to control and teach the robot. (4) communicates actions between the interface (3) and the learner (7). (5) translates teacher’s actions into robotic commands used by (6) and (8) and executed by the robot (9). Last, (7) is the learning algorithm, which defines a policy based on the state perceived and the previous actions selected by the teacher, their substates, and their feedback on propositions. The different nodes communicate using Robot Operating System (ROS).

Downloaded from <http://robotics.sciencemag.org/> by guest on October 24, 2019

state defined in a 210-dimension vector with values ranging from 0 to 1 and corresponding features describing the state of the game (animal’s energy and distance between items) and of the interaction (how long it has been since the child or the robot touched items, when was the last action executed by the robot, etc.).

In the supervised condition, the teacher used an interface running on a tablet and replicating the graphics of the game (with the position of the animals) but with additional buttons to select actions for the robot to execute. Our algorithm, adapted from (23), used a variation of nearest neighbors to map actions selected by the teacher to a substate ($s' \in S'$, with $S' \subset S$), a sliced version of the 210-dimension state (n' dimensions of the state have a value, whereas the others, not relevant to the current action, are left as “wild cards”). This slicing was carried out by only keeping the dimensions relevant to a set of features defined by the teacher (i.e., selected on the tablet). This allowed the algorithm to consider only the dimensions of the state relevant to each action when computing the distance between instances and the current state. Consequently, this algorithm can profit from having access to a large number of state dimensions without suffering from the “curse of dimensionality” (37), thus potentially learning quickly complex behaviors. In addition, each instance in memory has a reward value (r) that allows the algorithm to avoid undesired actions (the ones with a negative reward). In summary, instances are defined as tuples: action—substate—reward (a, s', r).

This learning algorithm could propose actions to the teacher that were executed after a short delay if the teacher did not cancel them. Using the interface, the teacher could accept (rewarding positively and executing) proposed actions or refuse them (preempting the execution of an action and assigning it a negative reward). In addition, they could select actions for the robot to execute. Figure 7 shows the flowchart of the action selection process allowing mixed initiative between the teacher and the robot.

The algorithm itself did not take time into account. However, because dimensions of the state are time dependent (using exponential decreases since events), temporal effects could be captured by the learning algorithm (as shown in Fig. 3B).

In the autonomous condition, the interface used by the teacher is simply replaced by a node automatically accepting propositions after a short delay, thus applying the policy learned in the super-

vised condition. All sources are open and available online at <https://emmanuel-senft.github.io/experiment-learning-tutoring.html>.

Metrics

To address the hypotheses, we collected multiple metrics on both interactions (teacher-robot and robot-child). The goal of the study being to evaluate whether the robot can replicate the teacher’s policy, we first recorded metrics characterizing these policies: the actions executed by the robot in the supervised and autonomous conditions and the timing between these actions and game-related events. Second, we collected two groups of metrics to evaluate the application interaction: the learning metrics (corresponding to the child’s performance during the tests) and the game metrics (corresponding to the child’s behavior within the game rounds). These learning outcomes are not critical for the study but serve to characterize the impact of the robot’s policy on the children. And last, in the supervised condition, we recorded the origin of the actions executed by the robot (teacher versus algorithm) and the outcome of the proposed actions (executed versus refused).

During the game, the robot had access to 655 actions, which can be divided into seven categories: drawing attention, moving close, moving away, moving to, congratulation, encouragement, and remind rules. Because of this high number of actions, the breadth of the state space (210 dimensions), and the complex interdependence between actions and states, precisely characterizing a whole policy was nontractable. Consequently, we used the number of actions executed for each category per child and the timing between a specific event (the child feeding an animal) and the execution of actions to characterize the policy executed by the robot in the active conditions (supervised and autonomous). Although not perfectly representing the policy of each condition (e.g., complex interdependencies are missing), these metrics offer a proxy to compare these policies.

The children’s knowledge about the food web was evaluated through a graph where children had to connect animals to their food. There were 25 correct connections and 95 incorrect ones. Because the child could create as many connections as desired, the performance was defined as the number of correct connections above chance (for the total number of connection made during the test) divided by the maximum achievable performance. This resulted in a score bounded between -1 and 1 .

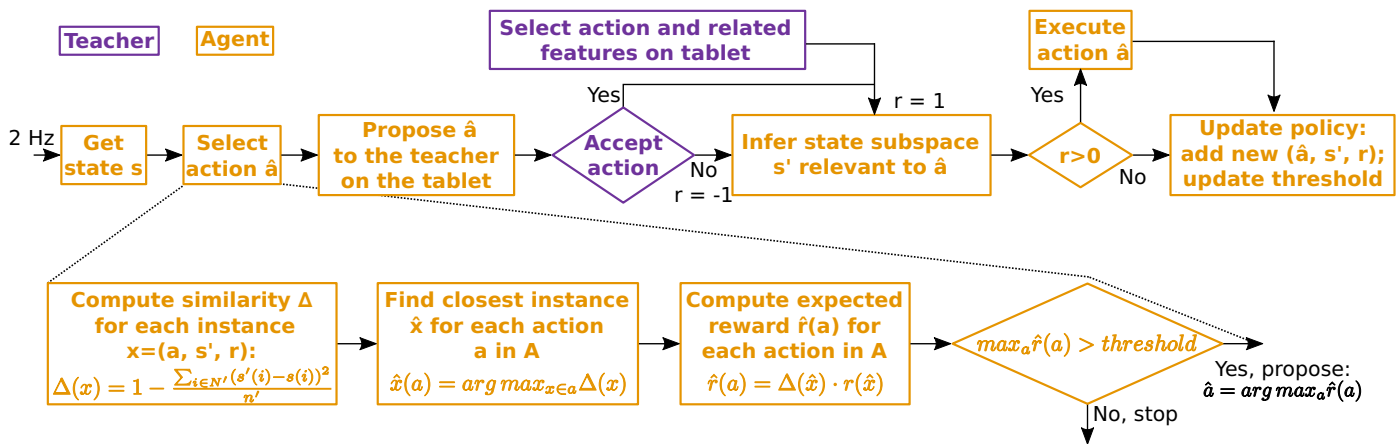


Fig. 7. Flowchart of the action selection. Mixed-initiative control is achieved via a combination of actions selected by the teacher, propositions from the robot, and corrections of propositions by the teacher. The algorithm uses instances x , corresponding to a tuple: action, a ; substate, s' ; and reward, r . s' is defined on S' with $S' \subset S$ and N' the set of the indexes of the n' selected dimensions of s' .

For example, if a child made five good connections and three bad ones, their performance would be

$$P = \frac{\#good - (\#good + \#bad) \cdot \frac{\text{totalgood}}{\text{total}}}{\text{totalgood} - \text{totalgood} \cdot \frac{\text{totalgood}}{\text{total}}} = \frac{5 - (5 + 3) \cdot \frac{25}{25+95}}{25 - 25 \cdot \frac{25}{25+95}} = 0.168 \quad (1)$$

The three tests (pre-, mid-, and post-interaction) resulted in three performance measures. To account for initial differences in knowledge and the progressive difficulty to gain additional knowledge, we computed the learning gain as proposed in (32): $g = \frac{P_{\text{final}} - P_{\text{initial}}}{P_{\text{max}} - P_{\text{initial}}}$. This learning gain indicates how much of the missing knowledge the child managed to gain from the game (values above 0 indicate learning).

In addition, game metrics were also gathered during the rounds of the game to characterize the children's behaviors:

1) Exposure to learning units, corresponding to the number of unique eating interactions between two items explored by a child in a round (range = [0,25]).

2) Interaction time, duration of game rounds, and how long a round lasted until three animals ran out of energy (typical range, 0.5 to 3 min).

An important metric in education is the engagement with the learning material, i.e., what proportion of the learning domain children explore (38). In our case, children explored a food web with 25 correct and 95 incorrect connections. Because of the imbalance between these numbers, more knowledge is acquired by discovering one of the 25 correct connections rather than the 95 incorrect ones. Hence, we defined our first game metric as the number of different eating interactions children encountered during each game. An eating interaction happens when the child moves an animal to its food (or to a predator), and the number of different eating interactions represents how many different unique correct connections the child has discovered during the game (multiple eating actions between the same animals would count only once). A game with a high number of different eating interactions represents a game where the child engaged with the learning material, encountered more learning units, and should perform better in the tests. For simplicity, we termed this metric "exposure to learning units" because it encompasses how much knowledge a child has been exposed to in one round of the game.

On the other hand, the interaction time reached in the game provides information about the children's performance in the task (keeping the animals alive as long as possible) and their engagement. A disengaged child would finish the game earlier. We expect that an active robot would encourage and support the child and allow them to reach better scores on these game metrics.

Statistical analysis

To demonstrate the presence or the absence of effects, we analyzed the data using Bayesian statistics. We report the Bayes factor B_{10} , which represents how much of the variance of the metric is explained by a parameter [if $B_{10} < 1/3$, then there is no impact; if $B_{10} > 3$, then the impact is strong; and if $1/3 < B_{10} < 3$, then the results are inconclusive (39, 40)]. We analyzed the results using the JASP software (41). We used a Bayesian mixed ANOVA as an omnibus test to explore the impact of the condition and the repetition on the metrics. Additional post hoc tests used a Bayesian repeated-measures ANOVA or Bayesian-independent t test comparing the conditions one by one and fixing the prior probability to 0.5 to correct for multiple testing.

Results are presented with graphs using violin plots featuring the kernel density estimation of the distribution, raw data points, and/or the mean and the 95% CI.

SUPPLEMENTARY MATERIALS

robotics.sciencemag.org/cgi/content/full/4/35/eaat1186/DC1

Fig. S1. Steps of the study.

Table S1. Post hoc comparison of timing of actions for the supervised condition.

Table S2. Post hoc comparison of timing of actions for the autonomous condition.

Table S3. Exposure to learning units.

Table S4. Game duration.

REFERENCES AND NOTES

1. C. Breazeal, C. D. Kidd, A. L. Thomaz, G. Hoffman, M. Berlin, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)* (IEEE, 2005), pp. 708–713.
2. R. S. Sutton, A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, 1998).
3. A. Billard, S. Calinon, R. Dillmann, S. Schaal, *Springer Handbook of Robotics* (Springer, 2008), pp. 1371–1394.
4. B. D. Argall, S. Chernova, M. Veloso, B. Browning, A survey of robot learning from demonstration. *Robot. Auton. Syst.* **57**, 469–483 (2009).
5. P. Robinette, A. M. Howard, A. R. Wagner, Effect of robot performance on human–robot trust in time-critical situations. *IEEE Trans. Hum. Mach. Syst.* **47**, 425–436 (2017).
6. Y. Liu, A. Gupta, P. Abbeel, S. Levine, in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018), pp. 1118–1125.
7. M. C. Gombolay, R. E. Jensen, J. L. Stigile, S.-H. Son, J. A. Shah, Apprenticeship scheduling: Learning to schedule from human experts, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, New York, NY, 9 to 15 July 2016.
8. H. Admoni, B. Scassellati, in *Proceedings of the 16th International Conference on Multimodal Interaction (ACM, 2014)*, pp. 196–199.
9. C.-M. Huang, B. Mutlu, in *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction (ACM, 2014)*, pp. 57–64.
10. A. Mihoub, G. Bailly, C. Wolf, F. Elisei, Learning multimodal behavioral models for face-to-face social interaction. *J. Multimod. User Interfaces* **9**, 195–210 (2015).
11. P. Liu, D. F. Glas, T. Kanda, H. Ishiguro, Data-driven HRI: Learning social behaviors by example from human–human interaction. *IEEE Trans. Robot.* **32**, 988–1008 (2016).
12. L. Riek, Wizard of Oz studies in HRI: A systematic review and new reporting guidelines. *J. Hum. Robot Interact.* **1**, 119–136 (2012).
13. P. Sequeira, P. Alves-Oliveira, T. Ribeiro, E. Di Tullio, in *Proceedings of the 11th ACM/IEEE International Conference on Human Robot Interaction* (IEEE Press, 2016), pp. 197–204.
14. M. Clark-Turner, M. Begum, in *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction (ACM, 2018)*, pp. 372–372.
15. W. B. Knox, S. Spaulding, C. Breazeal, *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, Québec City, Québec, Canada, 27 to 28 July 2014.
16. W. B. Knox, S. Spaulding, C. Breazeal, in *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems* (International Foundation for Autonomous Agents and Multiagent Systems, 2016), pp. 1309–1310.
17. J. A. Fails, D. R. Olsen Jr., in *Proceedings of the 8th International Conference on Intelligent User Interfaces (ACM, 2003)*, pp. 39–45.
18. S. Amershi, M. Cakmak, W. B. Knox, T. Kulesza, Power to the people: The role of humans in interactive machine learning. *AI Mag.* **35**, 105–120 (2015).
19. W. B. Knox, P. Stone, in *Proceedings of the Fifth International Conference on Knowledge Capture (ACM, 2009)*, pp. 9–16.
20. A. L. Thomaz, C. Breazeal, Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artif. Intell.* **172**, 716–737 (2008).
21. T. L. Sanders, T. Wixon, K. E. Schafer, J. Y. Chen, P. Hancock, in *2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)* (IEEE, 2014), pp. 156–159.
22. S. Chernova, M. Veloso, Interactive policy learning through confidence-based autonomy. *J. Artif. Intell. Res.* **34**, 10.1613/jair.2584, 1–25 (2009).
23. E. Senft, S. Lemaignan, P. Baxter, T. Belpaeme, *Proceedings of the Artificial Intelligence for Human-Robot Interaction Symposium, at AAAI Fall Symposium Series*, Westlin Arlington Gateway, Arlington, VA, 9 to 11 November 2017.
24. D. Leyzberg, S. Spaulding, B. Scassellati, in *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction (ACM, 2014)*, pp. 423–430.
25. I. Leite, G. Castellano, A. Pereira, C. Martinho, A. Paiva, in *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction (ACM, 2012)*, pp. 367–374.

26. E. Senft, P. Baxter, J. Kennedy, S. Lemaignan, T. Belpaeme, Supervised autonomy for online learning in human-robot interaction. *Pattern Recogn. Lett.* **99**, 77–86 (2017).
27. E. Senft, P. Baxter, J. Kennedy, T. Belpaeme, SPARC: Supervised Progressively Autonomous Robot Competencies, in *International Conference on Social Robotics* (Springer, 2015), pp. 603–612.
28. M. Nurmi, Predictive text input (2006). US Patent App. 11/035,687.
29. Department for Education, Schools, pupils and their characteristics: January 2018 (2018); https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/719226/Schools_Pupils_and_their_Characteristics_2018_Main_Text.pdf [accessed 18 February 2019].
30. T. Belpaeme, J. Kennedy, A. Ramachandran, B. Scassellati, F. Tanaka, Social robots for education: A review. *Sci. Robot.* **3**, eaat5954 (2018).
31. P. Dillenbourg, Design for classroom orchestration. *Comput. Educ.* **69**, 485–492 (2013).
32. D. E. Meltzer, The relationship between mathematics preparation and conceptual learning gains in physics: A possible “hidden variable” in diagnostic pretest scores. *Am. J. Phys.* **70**, 1259–1268 (2002).
33. T. Schodde, K. Bergmann, S. Kopp, in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction* (ACM, 2017), pp. 128–136.
34. P. Baxter, R. Wood, T. Belpaeme, in *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction* (ACM, 2012), pp. 105–106.
35. Nao robots, www.ald.softbankrobotics.com/en/robots/nao.
36. M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, A. Ng, ROS: An open-source robot operating system, in *ICRA Workshop on Open Source Software*, vol. 3, p. 5.
37. R. Bellman, *Dynamic Programming* (Princeton Univ. Press, 1957).
38. N. S. Podolefsky, K. K. Perkins, W. K. Adams, Factors promoting engaged exploration with computer simulations. *Phys. Rev. Sp. Top. Phys. Educ. Res.* **6**, 020117 (2010).
39. H. Jeffreys, *The Theory of Probability* (OUP Oxford, 1998).
40. Z. Dienes, Bayesian versus orthodox statistics: Which side are you on? *Perspect. Psychol. Sci.* **6**, 274–290 (2011).
41. JASP Team, JASP (Version 0.8.6)[Computer Software] (2018).

Funding: This work was supported by the EU FP7 DREAM project (grant no. 611391), the EU H2020 Marie Skłodowska-Curie Actions project DoRoThy (grant no. 657227), and the EU H2020 L2TOR project (grant no. 688014). **Author contributions:** E.S., S.L., P.E.B., and T.B. designed the study. E.S. implemented the technical components based on S.L.’s work. E.S. and M.B. ran the study. M.B. taught the robot. All the authors contributed actively to the writing. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** Sources, preprocessed data, script required to generate the graphs, and JASP file for the statistical analysis can be found at <https://zenodo.org/record/3386613>. All other data needed to evaluate the conclusions in the paper are present in the paper or the Supplementary Materials.

Submitted 24 February 2019

Accepted 16 September 2019

Published 23 October 2019

10.1126/scirobotics.aat1186

Citation: E. Senft, S. Lemaignan, P. E. Baxter, M. Bartlett, T. Belpaeme, Teaching robots social autonomy from in situ human guidance. *Sci. Robot.* **4**, eaat1186 (2019).

Teaching robots social autonomy from in situ human guidance

Emmanuel Senft, Séverin Lemaignan, Paul E. Baxter, Madeleine Bartlett and Tony Belpaeme

Sci. Robotics 4, eaat1186.
DOI: 10.1126/scirobotics.aat1186

ARTICLE TOOLS	http://robotics.sciencemag.org/content/4/35/eaat1186
SUPPLEMENTARY MATERIALS	http://robotics.sciencemag.org/content/suppl/2019/10/18/4.35.eaat1186.DC1
RELATED CONTENT	http://robotics.sciencemag.org/content/robotics/3/21/eaat5954.full http://robotics.sciencemag.org/content/robotics/3/21/eaat7451.full
REFERENCES	This article cites 14 articles, 0 of which you can access for free http://robotics.sciencemag.org/content/4/35/eaat1186#BIBL
PERMISSIONS	http://www.sciencemag.org/help/reprints-and-permissions

Use of this article is subject to the [Terms of Service](#)

Science Robotics (ISSN 2470-9476) is published by the American Association for the Advancement of Science, 1200 New York Avenue NW, Washington, DC 20005. 2017 © The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works. The title *Science Robotics* is a registered trademark of AAAS.