Teaching robots social autonomy from in situ human guidance

Emmanuel Senft[1]*, Séverin Lemaignan[2], Paul E. Baxter[3], Madeleine Bartlett[1], Tony Belpaeme[1,4]

[1]Centre for Robotics and Neural Systems, University of Plymouth, UK,
[2]Bristol Robotics Lab, University of the West of England, Bristol, UK,
[3]L-CAS, University of Lincoln, UK
[4]ID Lab - imec, University of Ghent, Belgium

*Corresponding author; E-mail: emmanuel.senft@plymouth.ac.uk

Striking the right balance between robot autonomy and human control is a core challenge in social robotics, both in 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, a novel approach addressing this challenge whereby a robot progressively learns appropriate autonomous behaviour from *in situ* human demonstrations and guidance. Using online machine learning techniques, we demonstrate that the robot can 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 non-deterministic environments. Finally, 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. Additionally, the growing expectation is that robots should behave autonomously not only at a technical, task-specific level, but also in terms of social interactions.

In this article, 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 behaviour 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 behaviour, they will progressively shift their workload to the robot. In such a scenario, the robot's technical and social policies are co-constructed by the teacher during the learning phase, and the resulting (autonomous) robot behaviour 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 behaviour) skills, all while preserving the ability for a school teacher to oversee and, if needed, override the robot's behaviour.

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 behaviour. This starting from the onset of the learning/interaction.

R2 The robot needs to learn quickly, as 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 in their environment and receiving feedback from it, RL agents learn online. To be effective, this requires both the 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, as simulators fail to reproduce, at meaningful levels, the complexity and unpredictability of human behaviours. 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, is expensive, and sometimes not possible at all. Not being able to fully recover from errors in HRI is the norm rather than the exception: when one observes that human-robot interactions almost always require a level of trust, it becomes clear that if the human loses trust in the robot due to poor behaviour, the interaction breaks down and can often not be recovered (*5*). The risk of such failures limits the general applicability of classical RL to HRI (as this violates R1). Additionally, 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 which enables humans to teach skills to robots. However, it typically looks at kinaesthetic demonstrations (*6*) in deterministic environments (such as manufacturing, industrial 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 behaviours. Two main methods have explored how to learn social behaviour from humans. Firstly, by collecting data from human-human interactions and applying machine learning to derive an autonomous behaviour (*8, 9, 10, 11*). Secondly, 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 behaviour (*13, 14, 15, 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 datapoints. Additionally, even if the demonstrations are collected from domain experts, they are later analysed by technical experts. This reliance on technical experts to interpret demonstration data and create learning algorithms adapted to each environment limits the usability of such approaches solely by naïve users.

An alternative way is to move away from optimising 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 behaviour 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 behaviour 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. However, while holding promise, 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*).

As no learning method so far addresses the three requirements stated previously, in (*27*) we introduced SPARC (Supervised Progressively Autonomous Robot Competencies), a new 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 behaviour. 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 which 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. In order 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 behaviour 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 behaviour, 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 is often not the case of 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. Indeed, previous research only considered scenarios where the robot was either interacting in a simulated environment (*26*) or with another robot simulating a human (*27*). This paper aims to evaluate SPARC in a real human-robot interaction, 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 human-robot interactions beyond the specific educational scenario that we use as support in this article.

Case study: Robots as Tutors for Children Social robots have been explored as educational tools in the last decade. Due to increases in the number of pupils in the classroom and budget constraints (*29*), one-to-one interactions between teachers and students, while known to be highly beneficial, are limited. One solution is to use a robot to supplement the teacher to offer additional individualised 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 practises leads to poor adoption rates by schools (*31*). Having a robot which can be operated initially by the teacher but then gradually takes over control, would offer a tutoring experience which is better tailored to the particular learner or context.

Results

Study Introduction We present a study evaluating SPARC in a high-dimensional social task where 8 to 10 years 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 is 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 are jointly modelled as an optimisation problem with 210 continuous input values (last actions, distances between animals, etc.) and 655 potential output actions (motions, gestures, verbal encouragements, etc.).

The interaction consists 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 includes three conditions, designed to assess the impact of applying the proposed approach (SPARC) to this task. The control condition (*Passive condition*) uses a passive robot that only provides initial instructions and guidelines, but does not offer support during the learning game. The second, the *Supervised condition*, involves a robot which gradually learns from human demonstration how to provide support during the game by using SPARC. In this condition, the robot's controller evolves with each interaction with the participants (refining its suggestions to the teacher over time). Nevertheless, the control provided to the teacher through SPARC ensures that the robot's behaviour is consistent for all participants, and supports their inclusion as a single group for this condition. The third, the *Autonomous condition*, uses an autonomous robot which executes the policy learned in the supervised condition, but without ongoing supervision.

We run 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. Consequently, this study is 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 which produces behaviour similar to that of the teacher.** We hypothesise that the policies of the autonomous and supervised robots will present similarities in term of frequency and timings of actions and will both have a positive impact on the children compared to no behaviour.

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 of 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 behaviours that will lead to better results on these metrics than no behaviour (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 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. Suggestions by the robot are in blue, and actions from the teacher in orange. For example, at t=16.9s, the teacher accepted the suggestion by the robot. Alternatively, in some cases, such as the suggestion at t=20.6s, 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. Finally, at t=44.4s, 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 Fig. 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 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') were used more often than 'Remind rules' or 'Drawing attention'. This provides support for H1a. However, the number of times each action was executed for the autonomous and supervised condition was different (Bayesian T-Test: Congratulation: BF_{10}=37.8, Encouragement: BF_{10}=5.1x10^4, Drawing attention: BF_{10}=.53, Remind rules: BF_{10}=1.6x10^3 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.

Additionally, 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 time since the last eating event for each type of action (Bayesian ANOVA: supervised condition: $F(4, 1211)$=101, $p <$ .001, B_{10}=1.06x10^{71}, post-hoc analysis in Table S1, only Encouragement and Remind rules seem to present similarities - autonomous condition: $F(4, 1385)$=81.0, p < .001, B_{10}=1.53x10^{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 condition: 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 normalised learning gain (*32*), was apparent in both the passive condition (M=0.12, 95% CI: [0.07, 0.18]) and supervised condition (M=0.11, 95% CI: [0.06, 0.16]), with the performance in the autonomous condition slightly exceeding these (M=0.14, 95% CI: [0.09, 0.19]). However, the robot's behaviour during the game did not have a meaningful impact on the children's learning gain (Bayesian ANOVA: $F(2, 72)$=0.34, p=.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 on the effect of the robot's behaviour on the children during the game sessions.

Fig. 4A and Table S3 show the evolution of the total number of different 'learning units' (ie., 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,

$p < .01$, 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-Measure ANOVA: B_{10}=0.15), whilst 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 to 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, thus potentially helping them to get knowledge about what animals eat. Importantly, the autonomous robot managed to recreate this effect without the presence of a human in the action selection loop.

Fig. 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=.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), while differences were observed between the supervised and passive conditions (B_{10}=118) and a trend towards a difference between the autonomous and passive conditions (B_{10}=2.90). This indicates 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=.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 minutes (see the red line on Fig. 4B). When not fed, animals would run out of energy in 2.25 minutes, so if children did not manage to feed at least 7 of the animals at least once before that time, the game would stop. As 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 to a passive robot) from the onset, and the autonomous robot replicated this effect, thus these results support H1c.

Teaching the Robot Fig. 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=.47, Rejected propositions: R^2=0.09, $F(1.0, 23.0)$=2.18, p=.15 and Teacher-initiated actions: R^2=0.001, $F(1.0, 23.0)$=0.01, p=.91). We would have expected these results to be different: with the learning, the number of accepted propositions should have increased and both the number of refused propositions and teacher-initiated actions should have decreased, thus H2 is not supported. It should, however, be noted that these results are based on a single teacher, and might not be replicated with another teacher.

To provide insights on this result, we analysed 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 post-training 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 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 behaviours such as typing her observations on a laptop, while gazing at the interface at the start of interactions). However, while 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 three hours and only 25 independent interactions, the robot successfully learned social and pedagogical behaviour to support children in the educational activity. This learning happened online, using a teacher with no knowledge about the algorithm implementation or intent of the study. While 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 behaviour 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 timescale, and while maintaining an appropriate robot behaviour throughout the teaching and subsequently when the robot interacts autonomously.

Our methodology has two main facets: it learns a *social* behaviour; 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 to offline learning. First, it allows a human (the teacher) to remain in the learning loop, giving them the opportunity to observe and influence the evolution of the robot's behaviour. 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 behaviours, thus increasing the transparency of complex systems and easing the decision to deploy the robot to interact autonomously.

Additionally, 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 behaviour. 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 behaviours could be manifested based on 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 democratise the use of robots.

Learning in Real-World and Sensitive Environments While the advantages of learning online potentially apply to any IML methods, most of these approaches provide the teacher with only limited control over the behaviour executed by the robot. This lack of control cannot ensure that the robot's behaviour 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*). As 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 behaviours 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 behaviour 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 behaviour was safe and useful from the outset) and in the autonomous phase (by demonstrating a behaviour 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 prior to 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 behaviours, or the system learns offline from demonstrations. While presenting significant advantages compared to 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 re-enacted the teacher's way of supporting the children, and reached tutoring results on par with those of a human controlling the robot. Not only did the robot learn 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 overtime (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 others teachers and should explore ways to provide the teacher with more control not only on the overt robot behaviour (the one displayed in the application) but also in the teaching interaction (such as being able to control meta parameters of the learning algorithm).

While the learned behaviour is better than having no behaviour at all, it is still possible that a hand-designed or random policy is also not worse than teacher or learned behaviours. In other words, the learned policy is better than no policy at all, but it is unclear whether it is better than any other policy.

Finally, SPARC should also be applied to other domains and in combination with more learning algorithms to properly investigates its ability to generalise.

Conclusion This paper demonstrated the potential for SPARC to enable robots to learn from humans. This capability is especially useful in HRI as knowledge of the desired robot behaviour typically comes from domain experts, such as teachers or therapists, rather than roboticists. The standard approach to design robotic controllers requires multiple conversations between the engineers coding the behaviour 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, as 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 e-health, assistive robotics or education.

The implications of this study are two-fold: first, we have demonstrated that, with an appropriate methodology, Interactive Machine Learning 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 behaviours (such as timing between events and actions) can be taught in this way.

These two results are significant. The dynamic and stochastic nature of social interactions makes learning appropriate and contingent social behaviours 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 behaviours 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 if SPARC can be used to teach online a robot to interact in a complex, non-deterministic and real environment. In previous studies (*27, 26*) SPARC was only evaluated in simple environments and not for creating social behaviours. Consequently, this study investigates if SPARC can be applied to HRI to teach a robot to replicate a policy demonstrated by a human. The goal is 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 the 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 is 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. Due to its widespread application to HRI and child tutoring (*30*), we used the NAO robot[1]. Additionally, 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 behaviour and 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 (UK) were recruited to take part in the study. As both schools had an identical OFSTED 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 deal with 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). Due to 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

[1] `https://www.ald.softbankrobotics.com/en/robots/nao`

authorship was not considered and she was not involved in the study design. Consequently, while being 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 controlling 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 was provided before or during the study. As such, 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 together play a game about the food web (cf. Fig. S1A). They completed a quick demographic questionnaire and a first pre-test to evaluate their baseline knowledge (cf. Fig. S1B-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. S1F,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. S1H-K). Afterwards, the child completed a mid-test before playing another two rounds of the game and completing a last post-test to conclude the study. Fig. S1 shows examples of the screen throughout the interaction.

Implementation The robot is controlled using the architecture presented in Fig. 6 with all the nodes communicating together using the Robot Operating System (ROS) (*35*). The teacher interface runs on a separate tablet and is used only for the supervised condition. All the other nodes run on the large touchscreen computer displaying the game interface which is used to guide the child through the study and presents the game rounds and the tests. The default robot behaviour is 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 consists in a mapping between these actions and a representation of the state defined in a 210 dimensions vector with values ranging from 0 to 1 and corresponding features describing the state of the game (animal's energy, 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...).

In the supervised condition, the teacher uses 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*), uses a variation of Nearest Neighbours 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, while the others, not relevant to the current action, are left as 'wild cards'). This slicing is carried out by keeping only the dimensions relevant to a set of features defined by the teacher (i.e. selected on the tablet). This allows 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' (*36*), thus potentially learning quickly complex behaviours. Additionally, each instance in memory possesses a reward value (r) which 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 can propose actions to the teacher that are executed after a short delay if the teacher does not cancel them. Using the interface the teacher can accept (rewarding positively and executing) proposed actions or refuse them (pre-empting the execution of an action and assigning it a negative reward). Additionally, they can select actions for the robot to execute. Fig. 7 shows the flowchart of the action selection process allowing mixed initiative between the teacher and the robot.

The algorithm itself does not take time into account. However, as dimensions of the state are time dependant (using exponential decreases since events), temporal effects can 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 supervised 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 if the robot can replicate the teacher's policy, we first recorded metrics characterising 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 behaviour within the game rounds). These learning outcomes are not critical for the study but serve to characterise the impact of the robot's policy on the children. And finally, in the supervised condition, we recorded the origin of the actions executed by the robot (teacher vs algorithm) and the outcome of the proposed actions (executed vs 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. Due to this high number of actions, the breadth of the state space (210 dimensions) and the complex interdependence between actions and states, precisely characterising a whole policy is non-tractable. 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 characterise the policy executed by the robot in the active conditions (supervised and autonomous). While 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. As 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.

For example, if a child made 5 good connections and 3 bad, their performance would be:

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

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*): $g = \frac{P_{final} - P_{initial}}{P_{max} - P_{initial}}$. This learning gain indicates how much of the missing knowledge the child managed to gain from the game (values above 0 indicate learning).

Additionally, game metrics were also gathered during the rounds of the game to characterise the children's behaviours:

- **Exposure to learning units**: corresponding to the number of unique eating interactions between two items explored by a child in a round ([0,25]).
- **Interaction time**: Duration of game rounds, how long a round lasted until three animals ran out of energy (typical range 0.5 to 3 minutes).

An important metric in education is the engagement with the learning material, i.e. which proportion of the learning domain children explore (*37*). In our case, children explored a food web with 25 correct and 95 incorrect connections. Due to the imbalance between these numbers, more knowledge is acquired by discovering one of these 25 correct connections rather than the 95 incorrect ones. As such, 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’ as 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 analyse 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 $B_{10} < 1/3$ there is no impact, if $B_{10} > 3$ the impact is strong, and if $1/3 < B_{10} < 3$ the results are inconclusive (*38, 39*)). We analysed the results using the JASP software (*40*). 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-Measure 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% Confidence Intervals.

References

1. C. Breazeal, C. D. Kidd, A. L. Thomaz, G. Hoffman, M. Berlin, *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on* (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, *Robotics and Autonomous Systems* **57**, 469 (2009).
5. P. Robinette, A. M. Howard, A. R. Wagner, *IEEE Transactions on Human-Machine Systems* **47**, 425 (2017).
6. Y. Liu, A. Gupta, P. Abbeel, S. Levine, *2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018), pp. 1118–1125.
7. M. Gombolay, R. Jensen, J. Stigile, S.-H. Son, J. Shah (AAAI Press/International Joint Conferences on Artificial Intelligence, 2016).
8. H. Admoni, B. Scassellati, *Proceedings of the 16th international conference on multimodal interaction* (ACM, 2014), pp. 196–199.
9. C.-M. Huang, B. Mutlu, *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, *Journal on Multimodal User Interfaces* **9**, 195 (2015).
11. P. Liu, D. F. Glas, T. Kanda, H. Ishiguro, *IEEE Transactions on Robotics* **32**, 988 (2016).
12. L. Riek, *Journal of Human-Robot Interaction* **1**, 119 (2012).

13. P. Sequeira, *et al.*, *The Eleventh ACM/IEEE International Conference on Human Robot Interation* (IEEE Press, 2016), pp. 197–204.

14. M. Clark-Turner, M. Begum, *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* (2014).

16. W. B. Knox, S. Spaulding, C. Breazeal, *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems* (International Foundation for Autonomous Agents and Multiagent Systems, 2016), pp. 1309–1310.

17. J. A. Fails, D. R. Olsen Jr, *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, *AI Magazine* **35**, 105 (2014).

19. W. B. Knox, P. Stone, *Proceedings of the Fifth International Conference on Knowledge Capture* (ACM, 2009), pp. 9–16.

20. A. L. Thomaz, C. Breazeal, *Artificial Intelligence* **172**, 716 (2008).

21. T. L. Sanders, T. Wixon, K. E. Schafer, J. Y. Chen, P. Hancock, *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2014 IEEE International Inter-Disciplinary Conference on* (IEEE, 2014), pp. 156–159.

22. S. Chernova, M. Veloso, *Journal of Artificial Intelligence Research* **34** (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* (2017).

24. D. Leyzberg, S. Spaulding, B. Scassellati, *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, *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, *Pattern Recognition Letters* **99**, 77 (2017).

27. E. Senft, P. Baxter, J. Kennedy, T. Belpaeme, *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`, Last accessed on 18-02-2019.

30. T. Belpaeme, J. Kennedy, A. Ramachandran, B. Scassellati, F. Tanaka, *Science Robotics* **3**, eaat5954 (2018).

31. P. Dillenbourg, *Computers & Education* **69**, 485 (2013).

32. D. E. Meltzer, *American journal of physics* **70**, 1259 (2002).

33. T. Schodde, K. Bergmann, S. Kopp, *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction* (ACM, 2017), pp. 128–136.

34. P. Baxter, R. Wood, T. Belpaeme, *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction* (2012), pp. 105–106.

35. M. Quigley, *et al.*, *ICRA Workshop on Open Source Software* (Kobe, Japan, 2009), vol. 3, p. 5.

36. R. Bellman, *Dynamic Programming* (Princeton University Press, Princeton, 1957).

37. N. S. Podolefsky, K. K. Perkins, W. K. Adams, *Physical Review Special Topics-Physics Education Research* **6**, 020117 (2010).

38. H. Jeffreys, *The Theory of Probability* (OUP Oxford, 1998).

39. Z. Dienes, *Perspectives on Psychological Science* **6**, 274 (2011).

40. JASP Team, JASP (Version 0.8.6)[Computer Software] (2018).

Supplementary Materials

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.

Acknowledgments

Funding: Supported by the EU FP7 DREAM project (grant 611391), the EU H2020 Marie Sklodowska-Curie Actions project DoRoThy (grant 657227) and the EU H2020 L2TOR project (grant 688014). Author contributions: Emmanuel Senft, Séverin Lemaignan, Paul Baxter and

Tony Belpaeme designed the study, Emmanuel Senft implemented the technical components based on Séverin Lemaignan's work. Emmanuel Senft and Madeleine Bartlett ran the study, Madeleine Bartlett taught the robot. All the authors contributed actively to the writing. Competing interests: The authors have no competing interest. Data and materials availability: All (other) data needed to evaluate the conclusions in the paper are present in the paper or the Supplementary Materials. 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.

Table 1: Example of events during the first minute of the first round of the interaction with the 23rd child in the supervised condition. Lines in blue represent suggestions from the robot and orange the reactions from the teacher. ('mvc' is the abbreviation of the move close action, time is provided in second.)

Time	Event
4.1	childtouch *frog*
4.3	failinteraction *frog wheat-3*
4.9	animaleats *frog fly*
5.8	childrelease *frog*
6.6	robot proposes congrats
7.6	childtouch *fly*
7.6	teacher selects wait
8	animaleats *fly apple-4*
8.3	childrelease *fly*
9.1	teacher selects congrats
9.1	childtouch *frog*
10.3	childrelease *frog*
10.8	childtouch *frog*
11.2	animaleats *frog fly*
12.4	failinteraction *frog apple-2*
12.5	animaleats *frog fly*
13.2	childrelease *frog*
14.2	childtouch *fly*
14.5	animaleats *fly apple-2*
14.6	robot proposes encouragement
15	childrelease *fly*
15.4	animaleats *fly apple-3*
16.9	teacher confirms encouragement
18.2	childtouch *snake*
18.4	failinteraction *snake wheat-3*
18.7	animaleats *snake bird*
19.6	animaleats *snake bird*
20.5	childrelease *snake*
20.6	failinteraction *snake wheat-4*
20.6	robot proposes congrats
20.9	childtouch *eagle*
21.1	animaleats *eagle bird*
22	animaleats *eagle bird*
22.4	childrelease *eagle*
23.3	animaldead *bird*
23.4	teacher selects instead mvc *dragonfly* - *fly*
23.6	robottouch *dragonfly*
26.9	robotrelease *dragonfly*
27.7	childtouch *fly*
28	childrelease *fly*
28.4	childtouch *dragonfly*
28.6	failinteraction *dragonfly apple-1*
29.1	childrelease *dragonfly*
29.4	failinteraction *dragonfly apple-1*
30.3	childtouch *dragonfly*
30.3	failinteraction *dragonfly apple-1*
30.7	robot proposes encouragement
31	failinteraction *dragonfly apple-1*
31.8	teacher selects wait
32.5	childrelease *dragonfly*
34.4	childtouch *wolf*
34.7	robot proposes remind rules
35	animaleats *wolf mouse*
36	teacher selects wait
36	animaleats *wolf mouse*
37.2	childrelease *wolf*
37.7	childtouch *grasshopper*
38.3	robot proposes congrats
42.1	failinteraction *grasshopper apple-1*
42.7	childrelease *grasshopper*
42.7	failinteraction *grasshopper apple-1*
44.4	teacher selects instead mvc *mouse* - *wheat-1*
44.6	robottouch *mouse*
44.7	childtouch *butterfly*
45.1	failinteraction *butterfly wheat-2*
45.6	childrelease *wheat-1*
45.6	robotrelease *mouse*
45.7	robottouch *mouse*
48.9	robotrelease *mouse*
49.3	childtouch *butterfly*
49.3	failinteraction *butterfly wheat-1*
49.6	childrelease *butterfly*
50	childtouch *mouse*
50.3	animaleats *mouse wheat-1*
51	childrelease *mouse*
51.1	animaleats *mouse wheat-2*
51.4	robot proposes congrats
52.3	teacher confirms congrats
52.9	childtouch *snake*
52.9	failinteraction *snake wheat-3*
53.2	childrelease *snake*
53.5	childtouch *mouse*
53.6	animaleats *mouse wheat-3*
54.4	robot proposes congrats
54.5	animaleats *mouse wheat-4*
55	childrelease *mouse*
55.6	childtouch *dragonfly*
56.1	teacher selects wait
56.8	failinteraction *dragonfly apple-1*
57.3	childrelease *dragonfly*
57.5	failinteraction *dragonfly apple-1*
58.6	childtouch *grasshopper*
58.6	failinteraction *grasshopper apple-1*
58.8	childrelease undefined
59.1	childtouch *dragonfly*
59.1	failinteraction *dragonfly apple-1*
59.2	failinteraction *grasshopper apple-1*
59.9	failinteraction *dragonfly apple-1*

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).

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. While the picture depicts an early lab pilot, the main study was conducted on actual school premises.

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 analysed). Each point represents one execution of an action.

Fig. 4. Comparison of children's behaviour 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 minutes, 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 correspond to the beginning of the training; Child number 25 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.

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) analyses 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). Finally, (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 ROS.

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'.