Thèse n° 10 144

Mutual Understanding in Educational Human-Robot Collaborations

Présentée le 9 septembre 2020

à l'École Polytechnique Fédérale de Lausanne
à la Faculté informatique et communications
Laboratoire d'ergonomie éducative

et à l'Instituto Superior Técnico (IST) da Universidade de Lisboa
Programme doctoral en robotique, contrôle et systèmes intelligents
et Doutoramento em Engenharia Informática e de Computadores

pour l'obtention du grade de Docteur ès Sciences

par

Alexis David JACQ

Acceptée sur proposition du jury

Prof. A. Ijspeert, président du jury
Prof. P. Dillenbourg, Prof. A. Paiva, directeurs de thèse
Prof. P.-Y. Oudeyer, rapporteur
Prof. M. Chetouani, rapporteur
Dr A. Alahi, rapporteur

École
polytechnique
fédérale
de Lausanne

2020

Acknowledgements

I would like to start by acknowledging **Ana Paiva** and **Pierre Dillenbourg**, who advised me for almost five years. My thesis started in Lausanne under the supervision of Pierre, who taught me how to talk, how to write and how to work as a scientist. He was and remains the best example of an engaged team leader, always present despite his filled calendar and always perceiving and expecting the best of his students. Then, Ana directed me in Lisbon. I want to thank her for her unique ability to motivate and encourage my projects. Doing so, she provided me with a stronger self-confidence and (that is less sure) a better autonomy. She taught me how to feel as a scientist. I further would like to thank **Severin Lemaignan**, who taught me all the rest: how to use Linux, how to code in python and C++, how to use Github, how to program a robot, how to conduce an experiment and how to write a paper. I thank **Wafa Johal** who perfected this education and who re-motivated me when I was strongly demotivated after my first candidacy exam at EPFL. I also thank **Francisco Melo**, who gave at Lisbon the best reinforcement learning course I followed and who helped me at finding a way to apply my theoretical background to this less theoretical thesis. I also have to thank **Olivier Pietquin**, as well as **Matthieu Geist**, for helping me with the Learning from a Learner part and, much more than this, for helping me getting the most prestigious after-Ph.D.-life I could have imagined.

Then, my thought comes to all the friends and colleagues I met during this thesis. At Lausanne, I can only start with **Thibault Asselborn** who helped me whistling at work, singing Charles Aznavour in the car, drinking too much the night and killing zombies the remaining time. Then comes **Ayberk Osgur**, my former roommate who made me discover Futurama and the amazing turkish cuisine and who was able to build a swarm of educational moving robots from scratch. Finally **Louis Faucon**, thank you for all the movies sessions, the games he always won and his immortal motivation for sharing a beer. At Lisbon, I want to thank **Hang Yin** for his incredibly surprising humor and the passionate mathematical discussions, **Brian Ravenet** for the epic surf sessions, **Ali Kordia** for the Arack and **Ramona Merhej** who promoted wonderful laboratory hinking in Portugal. And all the others in both laboratories: **Patrícia Alves-Oliveira**, **Kshitij Sharma**, **Filipa Correia**, **Lorenzo Lucignano**, **Maria Ferreira**, **Teresa Yeo**, **Raul Paradeda**, **Arzu Güneysu**, **Himanshu Verma**, **Sofia Petisca**, **Mojgan Hashemian**, **Fernando Garcia**, **Miguel Faria**, **Elmira Yadollahi**, **Miguel Vasco**

and all the rest (doing a thesis in two laboratories makes a lot of people to acknowledge).

Next, I wish to thank all the friends I knew for a longer time. I want to express my gratitude to **François Bienvenu**, who indirectly contributed to this thesis by sharing with me his passion for science when I started my undergrad studies, thanks to what I found the best intrinsic motivation for studying mathematics, computer science and biology. He also strongly encouraged me to finish the present manuscript. I thank **Thomas Schmitt**, obviously for all the 42 we took from the rain, also for chasing the terrible daemons of the non-poetic life avec nos bottes. **Nicolas Manich**, who initiated me to the contact-improvisation and who taught me that "when nothing is good, take a break and think of what you really want to do", as well as the nollie pressure hardflip. Talking about skateboard, I must thank **Aymeric Nocus**, for all the sessions across France and all the laughs when I needed to.

I thanks my parents **Anne Chieze** and **Gwenael Jacq**, and my oldest friends: my sister **Carole Jacq** and my brother **Julien Magnan**. Their contribution to this thesis and to my whole life could not even stand in a book, I will not try in a single sentence. Less old friends, but sister and brothers as well: I thank **Adelle Jacq**, **Arthur Delasalle** and **Victor Delasalle**. I could not write these familial acknowledgments without mentioning my family in Switzerland, who warmly welcomed me during my stay in that country. This starts with **Sebastien Chieze**, who initiated me with the Chaudron at L'isle – ritual that changed my perception of the spiritual world. Follow all my uncles and cousins **Philippe**, **Babou**, **Tatiana**, **Laurenna**, **Tamara**, **Guillaume**, **Marie**, **David**, **Blandine**, **Emmanuel**, **Brigitte**, **Caroline**, **Nathalie** and **Frank**.

Finally, I thank the present thesis for leading me to meet my most favourite being in the universe: my wife and my best friend **Daria Ishkova**.

Gif sur Yvette, 30 April 2020 A. J.

Abstract

Education is an art close to theater. A teacher is taking a role; he works his speeches and his gestures and he plays with the attention of his audience. But it is harder: more than entertaining, a teacher must shape the skills, the knowledge and the motivation of his students. This requires, more than just understanding the learning dynamic of students, the talent to control the way he is understood so he can manipulate this learning dynamic. We call it *mutual understanding*, formalized by the accuracy of the prediction of others and of the prediction of oneself by the others.

Robots for education, a field that emerges from novel approaches involving new technologies, opens a large horizon of unexplored pedagogical activities. Indeed, robots can take roles that were not doable by humans. For example, *CoWriter* is a robot that personifies a very unskilled beginner so even a child with strong difficulties can teach it handwriting: involving an adult would not be convincing and calling another child would be unethical for this role. However, a strong limitation lies in the fact that robots have a restricted perception to understand humans and are hardly understandable by humans. By consequence, robots for education suffer the poor – even nonexistent – level of *mutual understanding* required by educational interactions.

The first part of this thesis highlights the importance of the human-robot *mutual understanding* in pedagogical collaborative activities like *CoWriter* and is based on real-world experimentation. The next two parts form a suggestion to implement such an ability in a robot aiming to interact with humans by focusing on the modelling of motivations. One part regards the external orchestration of the different models built by the robot to make predictions and to be predictable. The other part focuses on the internal mechanisms of these models, based on the computational framework of reinforcement learning.

Résumé

L'éducation est un art proche du théatre. L'enseignant se vêtie d'un personnage ; il travaille son discours et sa gestuelle. Il joue avec l'attention de son audience. Mais il s'agit d'un art plus difficile encore : au delà de divertir, l'enseignant se doit de modeller les compétences, le savoir et la motivation de ses élèves. Cela tient non seulement de sa comprehension de la dynamique d'apprentissage des élèves, mais aussi de son talent à controller sa manière d'être compris de telle sorte qu'il puisse dompter cette dynamique. Nous attribuons à cette capacité le terme de *comprehension mutuelle*, que l'on formalise par la justesse de la prédiction d'autrui et de la prédiction de soi-même par autrui.

La robotique pour l'éducation, un champ de recherche qui émerge des récentes methodes faisant intervenir les nouvelles technologies, ouvre un large horizon d'activités pédagogiques jusqu'alors non explorées. Et pour cause, les robots peuvent endosser des rôles qui ne pourraient pas être arborés par les humains. Par exemple, *CoWriter* est un robot qui personifie un débutant totallement inexpérimenté, tant et si bien qu'un enfant lui-même en difficulté sera en mesure de lui apprendre l'écriture manuscrite. Donner ce role à un adulte serait moyennement convainquant, et le recours à un autre enfant manquerait d'hétique. Cela dit, de telles interactions sont très limitées par la perception restreinte des robots pour comprendre les humains et leur difficulté à se faire comprendre par les humains. En conséquence, la robotique pour l'éducation souffre de la pauvreté – même de l'innexistance – du niveau de *compréhension mututelle*, pourtant si nécessaire dans les interaction à vue pédagogique.

La première partie de cette thèse mets en évidance l'importance de la *compréhension mutuelle* dans les activités pédagogiques collaboratives telles que *CoWriter* et se base sur des expériences appliquées à la vie concrète. Les parties suivantes forment une suggestion d'implémentation d'une telle aptitude dans un robot conçue pour intéragir avec l'homme en se focalisant sur la modélisation des motivations. L'une porte sur l'orchestration externe des differents modèles construits par le robot afin de prédire et d'être prédictible. L'autre se concentre sur les mechanismes internes de ces modèles, dans le cadre computationnel de l'apprentissage par renforcement.

Resumo

A educação é uma arte parecida com o teatro. O professor atua; ele trabalha o discurso e os gestos, brinca com a atenção da plateia. Mas é ainda mais difícil: além de divertir, o professor tem que afiar as competências, o conhecimento e a motivação dos alunos. Não basta entender a dinâmica de aprendizagem dos alunos, ensinar requer também a habilidade de manipular essa dinâmica. Chamamos isso de *entendimento mútuo*, formalizado pela precisão da predição dos outros assim como a predição de si mesmo pelos outros.

Os robôs para educação constituem um campo que emergiu de métodos novos envolvendo tecnologias novas e que abre um horizonte largo de atividades pedagógicas a ser exploradas. Os robôs podem fazer papéis que os próprios humanos não conseguem. Por exemplo, *CoWriter* é um robô que atua como um aluno analfabeto para que uma criança, mesmo com dificuldade, possa ensinar a escrever para ele. Envolver um adulto não seria convincente e chamar outra criança não seria ético. Porém, os robôs mal conseguem entender os humanos assim como os humanos mal conseguem entender robôs. Por consequência, os robôs para educação sofrem do nível baixo, ou até zero, de *entendimento mútuo* que constitui um requisito para interações educacionais.

A primeira parte desta tese realça a importância do *entendimento mútuo* humano-robô em atividades pedagógicas colaborativas como *CoWriter* e se baseia em experimentações do mundo real. As duas partes seguidas propõem a implementação desta habilidade num robô destinado a interagir com humanos, focando na modelagem das motivações. Uma parte trata da orquestração externa dos vários modelos feitos pelo robô para fazer predições e para ele mesmo ser sujeito a predições. A outra parte foca nos mecanismos internos desses modelos, usando o formalismo computacional da aprendizagem por reforço.

Contents

List of abbreviations

ChRI Child-Robot Interaction 8, 11

HRI Human-Robot Interaction 3, 4, 111, 113

HAI Human-Agent Interaction 2, 3, 8, 92, 117

RL Reinforcement Learning 78–80, 82, 87, 93, 96, 97, 104, 108, 111, 114, 117

MARL Multi Agent Reinforcement Learning 81

IRL Inverse Reinforcement Learning 3, 79, 82–84, 92–95, 103, 106, 108–111, 114

MDP Markov Decision Process 80, 81, 97

POMDP Partially Observed Markov Decision Process 81

CSCL Computer-Supported Collaborative Learning 59, 113

EML Epistemic Modal Logic 57

BDI Beliefs Desires Intentions 57

RMM Recursive Modeling Method 58

ToM Theory of Mind 3, 56–59, 67, 74, 76, 110, 117

VFoA Visual Focus of Attention 29, 53

1 Introduction

"– It was a play on words, and a play on me. She could only do that with an awareness of her own mind, and also of awareness of mine.
– Yeah. She's aware of you, all right."

Dialogue from *Ex Machina*.

Intelligence can be described by the ability to build and exploit a predictive model of the changing world. People and animals are continuously constructing such models, from the empirical accumulation of knowledge obtained by exploring and experiencing their environment. Doing so, they are better and better at harnessing the world's dynamic, using the given affordance to improve their situations. Likewise, social intelligence lies on the construction and exploitation of robust models of others, as particular parts of the world. An agent may *understand* another agent if its model allows it to predict the other agent's behaviour. Then it can adapt its own behaviour to avoid conflicting trajectories and, hopefully, cooperate for a better mutual situation. Since another agent may also be continuously refining its model of the world, this modelling itself should be taken as a part of the world's dynamic. By consequence, a socially intelligent agent may learn to influence others modelling in order to approach a given objective, usually a collaborative task.

We introduce the notion of *mutual understanding*, which describes the ability of interacting agents to understand each other. This thesis is based on the assumption that mutual understanding is necessary for reaching efficient collaborations. We focus on the case of pedagogic human-agent collaborations. Since a human – especially a child – is not expected to intrinsically understand an artificial agent nor to facilitate an artificial agent to understand himself, the agent must be designed with the ability to promote the human-agent mutual understanding.

Through real-world experiments, we demonstrate the necessity of a better user modelling and a better agent's interpretability. We bring out technical methods and cognitive architectures to extract information relevant to mutual perception in Human-Agent Interaction (HAI). In an idealized context, we explore computational approaches to mutual understanding in intelligent agents interacting together.

1.1 Motivation

1.1.1 Artificial agents for educational activities

In a sense, any educative interaction is a collaboration. The teacher(s) – who design and/or control the activity – and the student(s) are sharing a common goal, namely the learning progress of the student. Given the design, many students and many teachers can be involved. However, one can only find student-teacher or student-student relationship in such collaborations, given the age or the level. One strong motivation of HAI in educative scenarios is the fact that new kinds of relationship can be explored. An agent can be an interactive tool providing the student or the teacher with hints and pieces of advice, or it can even take new studying/teaching roles. In the example developed in Chapter 2, a robot takes the role of a beginning learner that can be taught by a child, himself suffering important difficulties at learning the task. More than a simple scenario where the robot is given as a learning toy for the child, the robot can make specific mistakes, leading the child to correct his own mistakes by correcting the robot. An agent can also simulate any human role to teach tricky interactions situations, like the interrogations of suspects as part of a police training [Campos et al., 2017].

Unfortunately, HAI is suffering important weaknesses caused by the mutual misunderstanding induced by artificial agents. To the best of our knowledge, except in wizard-of-oz set-up where a hidden human controls the agent from a distance, the only efforts made to improve the mutual understanding are based on statistical prediction of the human states and behaviours. A robot will predict that a human is sad or happy using a facial recognition system, or will predict the intention of a grasping gesture. One can also find in literature some robots exaggerating their own behaviour in order to help a human understanding their intention [Nikolaidis et al., 2016a]. But such studies are limited to spacial reasoning for gestural adaptations [Nikolaidis et al., 2017] and are not adapted to pedagogical activities. The way a robot is perceived and understood in educational interactions has also been studied – especially in the case of handwriting acquisition [Chandra, 2019], but these estimations are made off-line and are not available for on-line adaptations.

1.1.2 Second order of modelling

In humans, mutual understanding relies on a psychological aptitude called Theory of Mind (ToM). It regroups all the cognitive processes allowing an human to construct a mental representation of another human's mind [Premack and Woodruff, 1978]. In that perspective, both the inference of someone's mental state and the prediction of someone's behaviour involve ToM mechanisms. But not only on the ability to construct models of the others is required to establish a mutual understanding: one also need to control others modellings of oneself. Indeed, one need to make sure he is being understood by the other while communicating, which requires to adapt the messages to the way they are interpreted by the interlocutor. Such an awareness is based on second-order representations [Baron-Cohen et al., 1985], a key capacity for the proper functioning of ToM [Dennett, 1978, Pylyshyn, 1978]. We generalize this assumption to the HAI domain by asserting that the missing ingredient in social artificial agents is the ability to infer and use second-order representations to establish a mutual understanding.

1.1.3 Modelling mutual objectives

When focusing on educational activities, an important feature to understand is the perception of the goal in other minds. A teacher (or a peer), must be able to realize which actions or which states are viewed as a progress by the student in order to adapt the steps of an activity. Similarly, the student needs to understand what is considered as a progress by the teacher. Building a mutual understanding about the objective of a task is already explored in Human-Robot Interaction (HRI), but these efforts are limited to teaching the robot from human demonstrations [Grizou et al., 2013, Nikolaidis et al., 2015]. More specifically, the robot usually infer the goal as a reward function from a set of human instructions. The technique used is based on Inverse Reinforcement Learning (IRL), which consists to assume that instructions are some realization of an optimal policy to solve the given task by maximizing the reward function [Ng et al., 2000]. However, this approach is insufficient when going through educational HAI scenario as soon as the agent is no longer a pure student but either a teacher, a peer, or playing a fake student in learning by teaching activities (see 2.2). In such setup, we would like to develop online methods to infer individual objectives without assuming the human's optimal behaviour, who is supposedly also discovering the activity. Similarly, an online method could help an artificial agent to infer how a human can understand its (the agent's) objective.

1.2 Approach

This whole thesis can be seen as a research for technical methods facilitating the mutual understanding in agents with shared pedagogical objectives. We proceed in three steps.

1.2.1 Step 1: pedagogical child-robot interactions

The first step (chapter 2) consists in developing and experimenting real-world HRI involving a pedagogical activity. We design an activity where a child learns handwriting by teaching a robot. Measuring the progression of the child, we argue that this progression mainly relies on the time of practice, so on the intrinsic commitment of the child. We play with the perception of the robot by the child by creating engaging scenario based on the "protégé" effect. In order to evaluate the robot's capability to perceive the human, we measure the accuracy of a technical setup that predicts the human "with-me-ness", a concept supposedly proportional to the engagement in an activity and based on the visual behaviour.

1.2.2 Step 2: architecture for mutual understanding

The second step (chapter 3) sets up an architectural framework for mutual understanding in robotic. We establish the necessity of a reasoning with 3 degrees of modelling: a model of one's own states and goals, a model of the other agents and a model of oneself as perceived by the others. We demonstrate the importance of this architecture with an experiment that controls different ways to reason with such models and measure the impacts on the quality of a collaborative interaction between a human and a robot.

1.2.3 Step 3: understanding learning agents

The third step (chapter 4) is more theoretical and focuses on the models. Given the 3-models structure described above (in the 2nd step), we discuss the ways to model each others states, goals and reasoning, and the ways to orchestrate these models in order to optimize a pedagogical objective. Since we aim at describing goal-oriented behaviours, we suggest a reinforcement learning framework to design our models. We develop a simple 3-model approach to 1) learn a policy that optimize an objective, 2) infer another agent's reward function and 3) adapt a behaviour to facilitate the other's inference of one's own reward function. We formalize pedagogical activities involving two agents as 2×2 theoretical games involving a social dilemma, with different payoffs/utilities given the agents roles and objectives. Given the propensity of agents to cooperate, we demonstrate the robustness of the proposed methods with numerical simulations. We then focus on the online inference of another agent's reward function and propose a novel algorithm *Learning from a Learner*, and we study its efficiency through discrete grid worlds and the more challenging *mujoco* environments, involving large-dimensional and continuous inputs.

1.3 Publications

The present document is build around the different papers/reports published along my four years of doctoral studies. Chapter 2 was essentially conduced at the Chili laboratory in the Suiss École Polytechnique Fédérale de Lausanne (EPFL) and merges three publications:

- Section 2.5 has been published in the *proceedings of the Human Robot Interaction conference* under the title “Building successful long child-robot interactions in a learning context” [Jacq et al., 2016b] and as a part of “Learning by teaching a robot: The case of handwriting”, published in the *Robotics and Automation Magazine* [Lemaignan et al., 2016b].
- Section 2.6 has been published in the *proceedings of the Human Robot Interaction conference* under the title “From real-time attention assessment to with-me-ness in human-robot interaction” [Lemaignan et al., 2016a].
- Section 2.7 has been published in the *proceedings of the Robot and Human Interactive Communication conference* under the title “Child-robot spatial arrangement in a learning by teaching activity” [Johal et al., 2016].

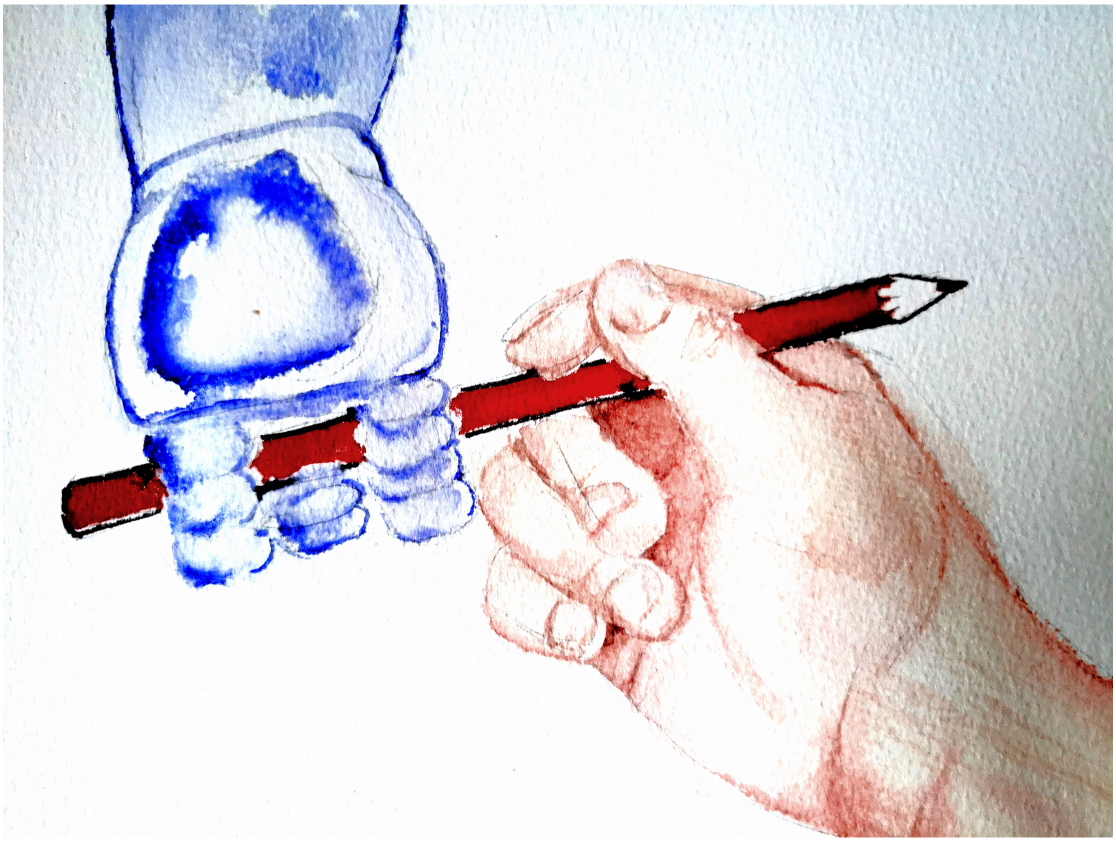
Chapter 3 was conduced at the Gaips group in the Instituto Superior Tecnico (IST) of the University of Lisbon and in the Instituto de Engenharia de Sistemas e Computadores, Investigação e Desenvolvimento em Lisboa (INESC-ID):

- Section 3.4 has been published as an extended abstract in the proceedings of the Autonomous Agent and Multi-Agent Systems conference under the title “Sensitivity To Perceived Mutual Understanding In Human-Robot Collaborations” [Jacq et al., 2018].

Chapter 4 was conduced both at the Gaips group and at Google Research, in the Brain Team’s laboratory of Paris. It merges two publications:

- Section 4.3 has been published as a technical report of IST under the title “Expressing Motivations By Facilitating Other’s Inverse Reinforcement Learning” [Jacq et al., 2017]
- Section 4.4 has been published in the proceedings of the *International Conference on Machine Learning* under the title “Learning from a Learner”.

2 Measurements in pedagogical Child-Robot collaborations



2.1 Introduction

This chapter reports observations from different studies of real-world Child-Robot Interaction (ChRI). The choice of a physical agent is justified by the strong impact on the field of interaction, and the richness of the induced visual behaviour. Indeed, one main difference between a virtual and a physical agent is the shared perception. The robot and the human are living in the same world and in aspect, they share a similar experience of the scene. Such an exchange could also be induced by plunging the human into the world of a virtual agent via virtual reality, however with limited affordance regarding the human gesture. This work is a part of the CoWriter Project, presented in section 2.3, aiming at exploring how a robot can help children with the acquisition of handwriting. The common point of the described studies is to make a child intrinsically motivated at practicing an educative activity, described in 2.4. Both pedagogical and therapeutic contexts are investigated, involving children to learn-by-teaching handwriting skills with a Nao robot. We focus on the following questions:

- (A) Can a robot actually help a child at learning a skill?
- (B) How different scenarios and experimental setups can impact the perception of the robot?
- (C) What can be measured during a ChRI in order to infer the child's internal state?

Question (A) is investigated in section 2.5 and regards the general field of educational ChRI. Application of physical robots in education is a whole field of research (see related work at section 2.2), essentially motivated by the fact that robots can take roles that do not exist in standard collaborative education. For example, robots have no specific age or gender, and will not suffer emotional bias such as timidity or boredom. Also this question is crucial: this whole thesis aims at improving educational HAI and hence relies on the fact HAI can be educational. Questions (B) and (C), investigated across sections 2.5, 2.6 and 2.7, concern the control of a pedagogical interaction with a physical robot: our interest is to evaluate what are the actual degrees of action and perception in such a low-priced and accessible robot at interacting with a child.

2.2 Related work

Robots for education

The usage of robots in education is now a whole field of research. It can be divided in two separate branches: robots as tools and robots as agents. Robots as tools do not socially interact with humans but bring a physical aspect to an educational concept.

Most robots as tools are designed to teach robotics or other fields related to engineering and computer science. From the acquisition of basic programming skills [Mondada et al., 2017] to the teaching of complete 3d-printing, construction and programming of a robot [Lapeyre et al., 2014]. Fewer works introduce tool robots designed to teach any pedagogical concept. A notable example is the Cellulo project [Ozgur et al., 2017], which introduces tangible swarm robots for inducing movements and forces as a new illustrative and manipulable dimension to pedagogical activities [Özgür, 2018].

This thesis is more related to robots as agents, which are used as social helpers for educational interactions. In the literature, different approaches can be found regarding the type of interaction, interpolating between robots teaching humans and humans teaching robots [Belpaeme et al., 2018]. The first extreme, robots as pure teacher or tutor, may be the oldest and the most known situation. This includes teaching assistant robots [You et al., 2006], class room robots [Tanaka et al., 2007] or personal teachers [Han et al., 2008, Movellan et al., 2009, Gordon et al., 2016, Kennedy et al., 2016]. Then, comes robots as guides for educational activities, mostly employed in small groups of two or three children [Chandra et al., 2015, Alves-Oliveira et al., 2016], or in therapeutic contexts involving autistic children [Bernardo et al., 2016]. In activities based on adversarial games, robots can also be employed as experienced opponents encouraging and advising the human [van Breemen et al., 2005]. A more original situation involves a robot playing the role of a fake patient for clinical education [Moosaei et al., 2017]. Robots as peers is the medium situation, where the robots are employed as learning companions for humans [Kanda et al., 2004, Lubold et al., 2016, Baxter et al., 2017]. Finally, robots as teachable agents is the approach taken for the experiments described in this chapter.

Teachable robots approaches are based on the *learning by teaching* paradigm, already known in fully human education for its positive effects on learning. However, like the patient robot for clinical education, this approach has the particularity to bring out new roles that could not be taken by real humans: in the example of Cowriter, asking a child with problems at writing to teach another child with even worst difficulties would be inefficient and even humiliating for the second child. Most of teachable robots lies on the Protégé effect: the fact that the learner being placed in the situation of a teacher to help someone triggers a feeling of responsibility that facilitates both engagement and intrinsic motivation. This effect has been used in the past by computer-based agents [Chase et al., 2009] to teach non-physical skills. The first teachable robot also concerned non-physical skills, but involved physical objects in an activity designed to teach vocabulary with the care receiver robot (CRR) [Tanaka and Kimura, 2009, Tanaka and Matsuzoe, 2012]. Robots maintain better long-term relationship [Kidd and Breazeal, 2008] and contribute to obtain more learning gains [Leyzberg et al., 2014] than with screen-based agents in pedagogical interactions. Specifically, when learning physical skills, robotic partners have been showed to increase users' compliance with the tasks [Bainbridge et al., 2011].

Robot perception

The relation between one's focus of attention and what he/she is looking at has long been established [Yarbus, 1967, Barber and Legge, 1976], and more specifically, the existing relationship between gaze and attention during social interaction, and the related gaze patterns, has been part of classic textbooks like [Argyle, 1969] for decades. As such, there is little doubt that measuring the direction of gaze is a useful proxy to estimate the (visual) focus of attention of a social agent, and indeed this is one of the basic tools used in social psychology.

Estimating attention using gaze is not new to robotics either. A recent survey by Ruhland *et al.* [Ruhland et al., 2015] gives in a broad overview of eye gaze research in HCI and social robotics. It remains however an active field of research, as illustrated by several recent publications [Baxter et al., 2014, Anzalone et al., 2015, Kennedy et al., 2015]. Performing such a measure on a robot, in real-time, and in ecologically valid environments (which rules out bulky or invasive apparatus like eye-trackers, or techniques requiring fine calibration and/or static interactions) remains a challenge in HRI.

Looking at techniques that both operate on-line and have been deployed in field experiments, one finds that most approaches rely on head pose estimation alone (no eye gaze tracking) and are generally based on depth sensors (RGB-D). Fanelli *et al.* provides an overview of these approaches in [Fanelli et al., 2012], and recent examples include [Baxter et al., 2014, Anzalone et al., 2015].

Approaches based on monocular 2D vision have been explored as well [Peters et al., 2010], with however limited robustness to occlusions or lightning conditions, and over-reliance on tracking to maintain real-time performances. Our work relies on recent advances in template-based face alignment [Kazemi and Sullivan, 2014] that allows fast (in the order of a few milliseconds) facial feature extraction on 2D images, combined with 3D model fitting, to obtain a fast, robust and stable 6D head pose estimate, that we successfully deployed in field experiments involving child-robot interactions.

We derive the field of attention from the head pose: this is supported by previous work, like [Stiefelhagen, 2002] that shows that the head orientation's contribution in overall gaze direction is 68.9%, which further translates into a 88.7% accuracy in estimating the focus of attention from head pose only in a particular meeting scenario (using eye and head tracking).

While previous preliminary research in HRI seemed on the contrary to indicate that deriving attentional focus from head pose alone would not be accurate enough [Kennedy et al., 2015], we found in our case acceptable levels of agreement between the robot observations and manual post-hoc annotations, as detailed in 2.6.

2.3 The CoWriter project

An important challenge of social robotics is to provide assistance in education. The ability of robots to support adaptive and repetitive tasks can be valuable in a learning interaction. Initially, the CoWriter Project introduced a new approach to help children with difficulties in learning handwriting [Hood et al., 2015a, Jacq et al., 2016b]. Based on the *learning by teaching* paradigm, the goal of this activity is not only to help children with their handwriting, but mainly to improve their self-confidence and motivation in practising such exercise. Now, the CoWriter project includes different ChRI activities where a child is invited to practice a pedagogical skill by working with a robot. In addition to the initial activity based on handwriting, CoWriter encompasses:

- CoReader [Yadollahi et al., 2018]: detecting and correcting mistakes made by a robot while reading a text.
- Story-CoCreation [Jacq et al., 2018]: creating a story by selecting elements of its content, turn by turn with a robot.
- Handwriting with Cellulo [Asselborn et al., 2018]: learning the shape of letters by moving tangible swarm robot along a path.
- Shruti Chandra's activity [Chandra et al., 2018]: another version of the initial CoWriter activity, focused on the shape of a letter rather than the physical gesture to draw it.

2.4 CoWriter's handwriting activity

Children facing difficulties in handwriting integration are more exposed to troubles during the acquisition of other disciplines as they grow up [Christensen, 2005]. The CoWriter activity introduces a new approach to help those children [Hood et al., 2015a]. While traditional successful interventions involve children in long intervention (at least 10 weeks) focused on *motor* skills [Hoy et al., 2011], CoWriter is based on *learning by teaching* paradigm and aims to repair self-confidence and motivation of the child rather than his handwriting performance alone.

Learning by teaching is a technique that engages the students to conduct an activity as the teachers in order to support their own learning process. This paradigm is known to produce motivational, meta-cognitive and educational benefits in a range of disciplines [Rohrbeck et al., 2003]. The CoWriter project is the first application of the learning by teaching paradigm applied to handwriting with a robot.

The effectiveness of our learning by teaching activity builds on the "protégé effect": the teacher feels responsible for his student, commits to the student's success and possibly experiences student's failure as his own failure to teach. Teachable computer-based agents have previously been used to encourage this "protégé effect", where students invest more effort into learning when it is for the benefit of a teachable agent than for themselves [Chase et al., 2009]. We rely on this cognitive mechanism to reinforce the child's commitment into the robot-mediated handwriting activity.

We assume here that the key of such a relationship between the child and the robot relies on the credibility of the robot: the more the robot convinces the child that it is a beginner in handwriting who needs help – therefore initiating a "protégé effect"– the deeper the child will engage in the interaction. We focus hereafter on two aspects that are instrumental in building a credible teaching situation: how to generate the initial state of the learner-robot, and how to design its learning behavior.

Interaction overview

Figure 2.1 illustrates our general experimental setup: a face-to-face child-robot interaction with an autonomous Aldebaran's NAO robot.

A tactile tablet (with a custom application) is used by both the robot and the child to write: in each turn, the child requests the robot to write something (a single letter, a number or a full word), and pushes the tablet towards the robot, the robot writes on the tablet by gesturing the writing (but without actually physically touching the tablet). The child then pulls back the tablet, corrects the robot's attempt by writing himself above or next to the robot's writing, and "sends" his demonstration to the robot by pressing a small button on the tablet. The robot learns from this demonstration and tries again.

Since the child is assumed to take on the role of the teacher, we had to ensure he would be able to manage by himself the turn-taking and the overall progression of the activity (moving to the next letter or word). In our design, the turn-taking relies on the robot prompting for feedback once it is done with its writing (simple sentences like "What do you think?"), and pressing on a small robot icon on the tablet once the child has finished correcting. We found that both approaches were easy to be understood by children.

Generating and learning letters

Since our approach is based on teaching a robot to write, generating (initially bad) letters and learning from demonstrations is a core aspect of the project. The initial state of the robot and his ability to learn in an obvious way from demonstrations of the child is the key to lend credibility to the activity and to induce the "protégé" effect.

The technical idea is simple: allographs of letters are encoded as a sequence of 70 points in 2D-space and can be seen as vectors with 140 elements $(x_1, ..., x_{70},\ y_1, ..., y_{70})$. We

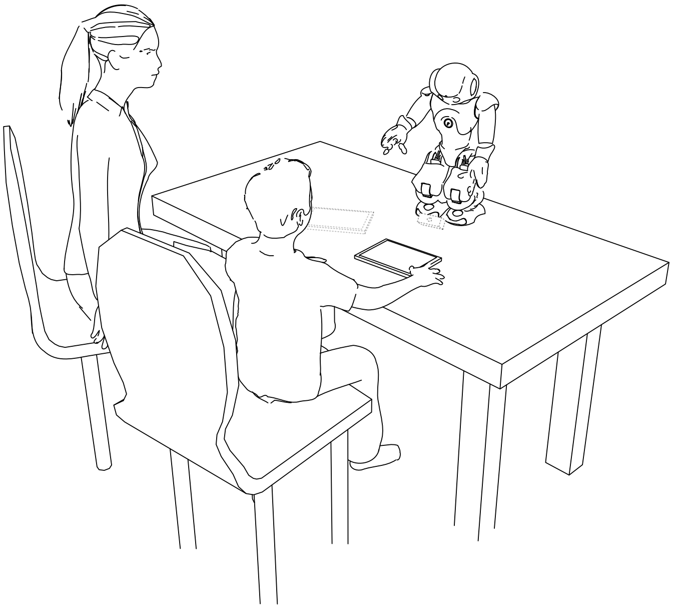


Figure 2.1 – Our experimental setup: face-to-face interaction with a NAO robot. The robot writes on the tactile tablet, the child then corrects the robot by directly overwriting its letters on the tablet with a stylus. An adult (either a therapist or an experimenter, depending on the studies), remains next to the child to guide the work (prompting, turn taking, etc.). For some studies, a second tablet and an additional camera (lightened) are employed.

arbitrary chose a set of allograph that define the initial state of generated letters. After the child provided a demonstration of a letter, the algorithm generates a new letter corresponding to the middle point between the last state and the demonstration.

In the following sections, we present various techniques to create the initial state, and different metrics used to compute progression of the robot, tested as hypothesis within our three experiments.

Generation of initial allographs

The first question relates to the construction of the initial set of allographs. In previous experiments presented in [Hood et al., 2015b], we built a subspace based on principal component analysis (PCA) of a standard dataset of 214 adult letters (the UJI Pen Characters 2 dataset [Llorens et al., 2008]). We used the first n eigenvectors (in these experiments, $3 < n < 6$) of the covariance matrix generated from PCA to create a subspace. To create new letter shapes, we chose random coordinates close to the origin of this subspace. Each eigenvector provided the direction of a principal deformation of the allograph in human handwriting [Hood et al., 2015a]. But generated "imperfections" of letters were not representative of children deformations: they were reflecting typical defects when adults are writing to fast. Over the following studies, we explored three different ways to generate samples closer to beginners. In our first case study (section 2.5.1), we used homework of the child previously provided by his mother, to exaggerate by hand his main defects. This way, the child was going to correct his own kind of mistakes. In the second study (section 2.5.2), the child was suffering from visuo-constructive deficits. Since it was difficult for him to improve already recognisable allographs, we decided under the guidance of his occupational therapist to make the robot start from simple vertical stroke for all letters. In the third study 2.5.3 we chose to use the middle point between a vertical stroke and correct letters as a starting point for the robot.

Metrics used for the learning curve of the robot

The second question focuses on the learning algorithm. In [Hood et al., 2015a], we were projecting children's demonstrations in PCA's subspace in order to compute the middle between that point and the previous state of the robot. Then, we generated the allograph in middle way as the new state of the robot. For the experiments introduced in this section, we explored two other ideas: In the first study (section 2.5.1) we generated a PCA subspace from a small set of allographs we drew arbitrary. Each time the child was providing a demonstration, we added that demonstration to the small set and re-built the PCA subspace. That way, the principal eigenvectors obtained progressively tended to encode the main deformations of letter done by the child, as illustrated by figure 2.4. The algorithm 1 explains the successive steps of this approach.

Algorithm 1: Learning from demonstration in an adaptive PCA subspace

```
Generate initial dataset $D$
Generate initial subspace $S$ by PCA of $D$
Generate initial robot state $r$ (random point in $S$)
**if** *robot receives a demonstration* $d$ **then**
    Add $d$ to dataset: $D' \leftarrow D \cup d$
    Recompute subspace $S'$ by PCA of $D'$
    Compute coordinates $r'$ of $r$ in $S'$
    Compute coordinates $d'$ of $d$ in $S'$
    Learn the demonstration: $r = \frac{(r'+d')}{2}$
```

From our perspective, this dynamic subspace was more adapted to the progression of the child, and the sequence of tries performed by the robot looked smoother. However using metrics in subspace can make the learning algorithm too slow in some cases, because consecutive projected demonstrations can sometimes be too far from each other in subspace while they appears similar in Cartesian space. In other studies, we decided to put aside the PCA approach and to always use the middle point in Cartesian space, in order to have a better control over the convergence of the robot tries to the demonstrations.

Robotic Implementation

The actual implementation on the robot requires the coordination of several modules (from performing gestures and acquiring the user's input to the high-level state machine), spread over several devices (the robot itself, one laptop and up to four tactile tablets for certain studies we conducted). We relied on ROS to ensure the synchronization and communication between different devices.

Our system is embodied in an Aldebaran's NAO (V4 or V5, depending on the studies) humanoid robot. This choice is motivated by its approachable design [Gouaillier et al., 2008], its size (58cm) and inherently safe structure (lightweight plastic) making it suitable for close interaction with children, its low price (making it closer to what school may afford in the coming years) and finally its ease of deployment on the field.

Robotic handwriting requires precise closed-loop control of the arm and hand motion. Because of the limited fine motor skills possible with such an affordable robot, in addition to the absence of force feedback, we have opted for *simulated handwriting*: the robot draws letters in the air, and the actual writing is displayed on a synchronized tablet.

The overall architecture of the system (Figure 2.2) is therefore spread over several devices: the NAO robot itself, that we address via both a ROS API[1] and the Aldebaran-provided NaoQI API, one to four Android tablets (the main tablet is used to print the robot's letter and to acquire the children's demonstrations; more tablets have been used in some studies, either to let the child input words to be written, or for the experimenter to qualitatively annotate the interaction in a synchronized fashion), and a central laptop running the machine learning algorithms, the robot's handwriting gesture generation and high level control of the activity.

Since the system does not actually require any CPU-intensive process, the laptop can be removed and the whole logic run on the robot. Due to the relative difficulty to deploy and debug ROS nodes directly on the robot, the laptop remains however convenient during the development phase and we kept using it in our experiments.

Most of the nodes are written in Python, and the whole source code of the project will be made is available online[2].

2.5 Long-term studies

As a follow up, this section reports on further experimental investigations. We explore different algorithmic and staging approaches built on top of the original system in order

[1] The ROS stack for NAO is available at http://wiki.ros.org/nao_robot.

[2] The primary repository is
https://github.com/chili-epfl/cowriter_letter_learning.

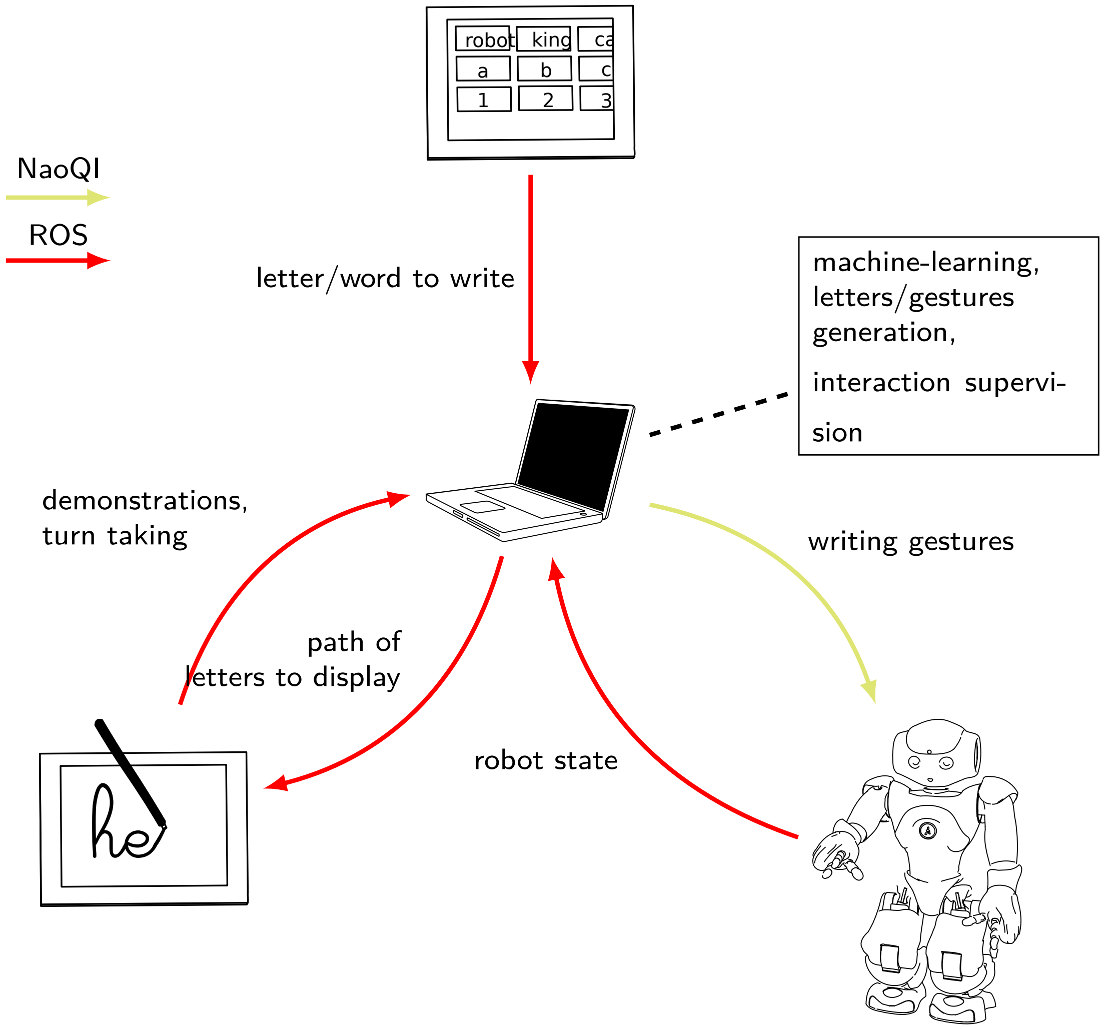

robot
king
ca
a
b
c
1
2
3
NaoQI
ROS
letter/word to write
machine-learning, letters/gestures generation,
interaction supervision
demonstrations, turn taking
path of letters to display
robot state
writing gestures
he

Figure 2.2 – **Overview of the system.** In total, the system runs about 10 ROS nodes, distributed over the robot itself, a central laptop and Android tablets.

to figure out intricate aspects of long child-robot interactions in a pedagogical context.

Through three experiments, we involved children with actual handwriting troubles or low self-esteem in repeated long sessions (four times about one hour). We used different measures, both qualitative and quantitative, to express the impact of those interactions with the CoWriter robot on the child. The following three parts report our the three experiments and results: two case studies specifically designed to be adapted to one child 2.5.1 2.5.2; one user study conducted with 8 children separately 2.5.3.

2.5.1 case study 1: Vincent

Context

Vincent[3] is a five year-old child. At school, he has difficulties to learn writing, particularly with cursive letters. From our perspective, Vincent is shy and quiet. He suffers from poor self-confidence much more than any actual writing problem. The experiment was conducted without any therapist, in our laboratory. A parent was here to accompany the child, but she did not intervene during interactions. Children's personalities, conditions and state evaluation were reported by the parent.

Hypothesis

The CoWriter activity needs a child engaged as interaction leader. With this study we consider the problem of long-term interactions. We hypothesize that with an appealing scenario children can maintain motivation in doing a handwriting activity for an hour over 4 sessions. This is a challenge because therapists predicted dysgraphic children often suffer from attention deficit [Jordan, 2002]. As a result, such children are not able to focus longer than 15-20 minutes.

Experimental design and methodology

Our goal was to provide Vincent with an environment that would enable him to sustain engagement over four one-hour sessions, one session per week. We decided to introduce a scenario to elicit a strong “protégé effect" and such induce a stronger commitment. While the child came with low motivation in writing exercise for himself, our idea was to use this effect to promote a new extrinsic motivation: improving letters in order to help the robot.

In our scenario we introduced the child with two Nao robots: a blue one (called Mimi) and an orange one (called Clem). Mimi was away for a scientific mission, and the two

[3] The names of children have been changed.

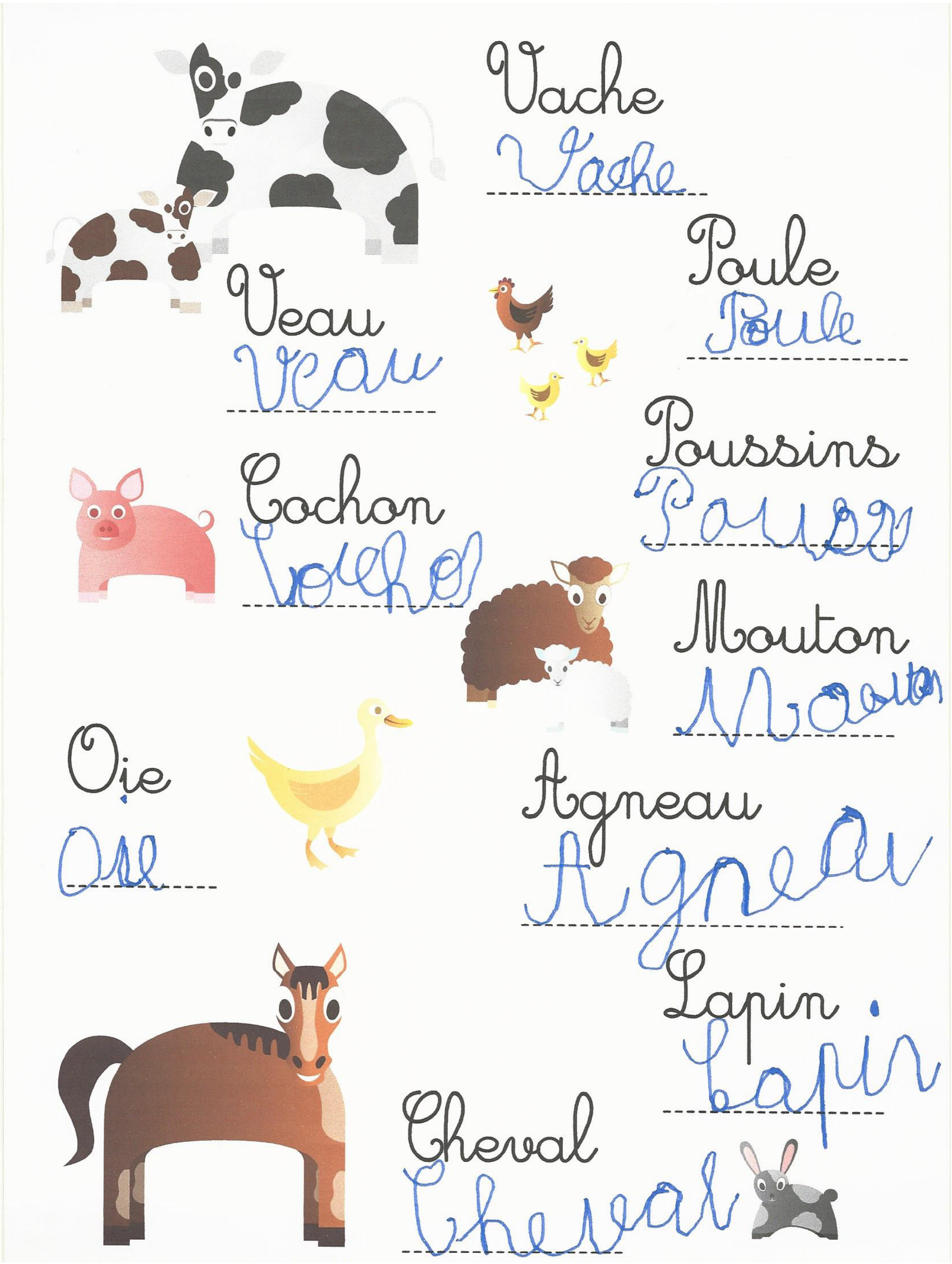

Vache
Vache
Poule
Poule
Veau
Veau
Poussins
Cochon
Mouton
Oie
Agneau
Agneau
Lapin
Lapin
Cheval
Cheval

Figure 2.3 – Homework performed by Vincent before the experiment. It gives an overvew of his starting level in handwriting.

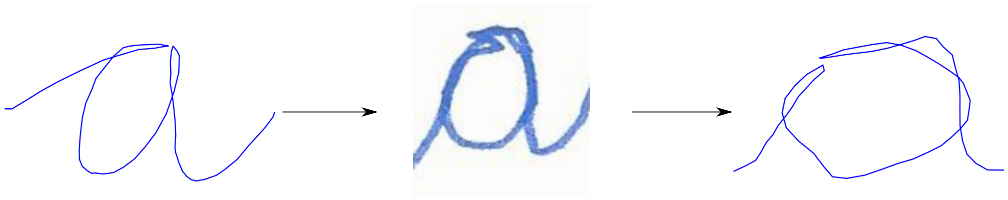


Figure 2.4 – Letter deformation along an eigenvector. *Left* : the non-deformed letter (origin of the subspace). *Middle* : the actual Vincent's deformation (from figure 2.3). *Right* : exaggerated deformation along the eigenvector that encode Vincent's mistake.

robots had to communicate by mails. But they decided to do it “like humans", with handwritten messages. While Mimi was good in handwriting, Clem had strong difficulties and needed Vincent's help.

Mimi's mission was to explore a mysterious hidden base. Each week, a postal mail contenting a picture of a curious object it found and a few handwritten words about its discoveries. The picture showed itself exploring a dark room of the hidden base (that was actually our laboratory's workshop).

During the three first sessions, Clem (the robot interacting with the child) was waiting for Vincent with the received mail. It let Vincent take a look at the picture and the object, and then it asked him to read the message. Finally, Vincent formulated a response and helped the robot to write it.

The fourth and last session was set as a test: Mimi, the “explorer” robot, came back from its mission and challenged Clem in front of Vincent: *“I don't believe you wrote yourself these nice letters that I received! Prove it to me by writing something in front of me!”* In this situation we forced the protégé effect: Clem is going to be judged on its writing skills by Mimi, but Vincent is here to give a last help and to encourage his student.

To complement the motivation of helping a robot to communicate with another one, we gradually increased the complexity of Vincent's task to keep it challenging and interesting (first week: demonstration of single letters; second week: short words; third week: a full message – Figure 2.5).

Vincent had to tell the robot what to write with small plastic letters. A third person was here to send the formed word to the robot via the computer.

During the experiment, we recorded writings of the child and the robot on the tablet into log files. We also recorded the time date when the child started and finished a demonstration.

Table 2.1 – Number of demonstrations provided by Vincent over the four sessions.

Session	S1	S2	S3	S4	Total
Number of demonstrations	23	34	52	46	155

Measures

We measured the commitment of the child with the number of demonstration he provided. We also measured the duration of sessions. During the two last sessions, we recorded the time taken by the child to write each demonstration.

After the experiment we interviewed the parent of the child. She was asked if she observed any impact of our activity on the child.

Analysis

We compared the number of demonstrations provided by Vincent along the 4 sessions (reported on Table 2.1) and we summed the time spend by the child to write demonstration during the 2 last sessions.

Results

Overall, Vincent provided 155 demonstrations to the robot. We can see in Table 2.1 that the number of demonstrations provided by Vincent was globally increasing along sessions while the difficulty of the activity was also increasing. Interestingly, as the number of demonstration decreased from session 3 to session 4, the total time spend to write demonstrations is similar: 41.6s in session 3 (∼0.8s per letter) and 41.1s in session 4 (∼0.89s per letter). A explanation of this result could be that since the difficulty was increasing, the child spent more time to write his demonstration.

After the first week, he showed confidence when playing with his “protégé" and he built affective bonds with the robot over the course of the study, as evidenced by some cries on the last session, and several letters sent to the robot *after* the end of the study (one of them 4 months later) to get news. This represents a promising initial result: we can effectively keep a child committed into the activity with the robot for a relatively long periods of time (about 4 hours).

From the parent’s perspective, Vincent was actually showing a new motivation in improving his handwriting. He took pleasure to work with the robot and to accomplish his teacher’s mission. She confirmed that an affection of the child for the robot took root within the experiment. Finally she saw an improvement of his handwriting and explained that the child “passed from a mix of script and cursive writing up to a full-cursive

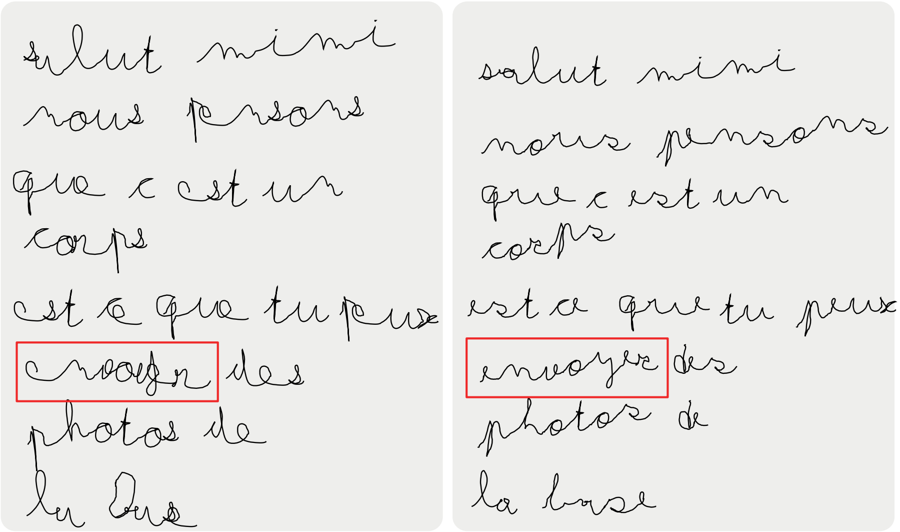


Figure 2.5 – (French) text generated by the robot, before (left) and after (right) one hour of interaction session with the child. As an example, the red box highlights the changes on the word "envoyer".

writing".

But no conclusion can be drawn in terms of actual handwriting remediation: we did not design this study to formally assess possible improvements. However, as pictured on Figure 2.5, Vincent was able to significantly improve the robot's skill, and he acknowledged that he had been able to help the robot: in that regard, Vincent convinced himself that he was "good enough" at writing to help someone else, which is likely to have a positive impact on his self-esteem.

2.5.2 case study 2 : Thomas

Context

Thomas, 5.5 years old child, is under the care of an occupational therapist. He has been diagnosed with visuo-constructive deficits. He was frequently performing random attempts and then was comparing with the provided template. According to the therapist, Thomas is restless and careless: he rarely pays attention to advice and does not take care of his drawing movement when he is writing. He is quickly shifting his attention from one activity to another.

Thomas was working on number allographs with his therapist. During a prior meeting, the therapist provided us with a sequence of numbers written by Thomas. one of the observed problems was drawing horizontally-inverted allographs, mainly for "5". The experiment was conduced with Thomas' therapist.

Hypothesis

We want to see if the CoWriter activity can be adapted to a pedagogical context in order help a child with diagnosed deficits to learn handwriting.

We believe that small modifications of the activity adapted to Thomas problems (visuo-constructive deficits and inattention) could help to keep him focused on the activity during forty-minutes sessions, and to evidence to the child that the robot is progressing by dint of his demonstrations.

Experimental design and methodology

The experiment was conducted in the therapist's office (four sessions spanning over 5 weeks). We assumed that a scenario like the one we used for Vincent would not be usable with Thomas. We just introduced the robot and quickly said that it was seeking help to train for a robot handwriting contest.

In order to integrate our work with that of the therapist, we decided to adapt the CoWriter activity to work with numbers.

Since Thomas was frequently drawing horizontally-inverted numbers, or even unrecognisable allographs, the learning algorithm of the robot was converging to meaningless scrawls. To fix this problem, we programmed the robot to refuse allographs that were too distant to a reference with a threshold we arbitrary fixed. In that way, the child was forced to take care of his demonstrations for the robot.

According to the therapist, it was easier for Thomas to memorize the way to draw a number if it was always done is the same trajectory, *e.g.* if the "5" was always drawn from the top-right tip down to bottom. Therefore we programmed the robot to refuse as well a good allograph drawn in a wrong trajectory. But in order to reassure Thomas about the right final allograph's shape, we made the robot able to recognize such a drawing, and, when it occurred, to use the phrase: *"Oh, this is exactly the shape of the number I want to learn, but can you show me how to draw it in the opposite trajectory?"*

Also, to make the robot's progresses evident, we modified the initialization step of the learning algorithm to start with a roughly vertical stroke instead of a deformed number (round 0 on Figure 2.6).

In this setup, we added a second tablet with one button per number. It was used by the child to chose a new number to teach to the robot. It also provided the possibility to enter letters or words, and to switch to another activity (robot telling a story if the child needs a short break).

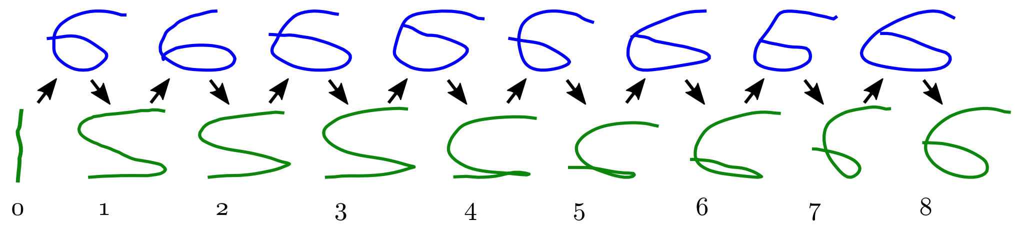


Figure 2.6 – Demonstrations provided by Thomas for the number "6" (top row) and corresponding shapes generated by the robot. After eight demonstrations, Thomas decided that the robot's "6" was good enough, and went to another character: in that respect, he was the one leading the learning process of the robot.

Measures

We recorded all the demonstration performed by the child and by the robot. The duration of sessions and the time spend by demonstration were also recorded by the logs of the tablet.

Analysis

It was difficult to make comparison between different sessions since the child did not work on the same numbers. But we could study the evolution of the quality of Thomas' demonstration when he was working on a given number (Figure 2.8). To show how Thomas leaded the robot to reach his level we plotted on the same graph the evolution of the quality of Thomas' demonstrations and the robot's trials (Figure 2.7). We also reconstructed and displayed the drawn allographs of the number 6 to visualize the impact of the lessons of Thomas on the robot (Figure 2.6).

Results

Despite his attention deficit, Thomas was able to remain engaged in the activity during more than forty minutes in each session. In total, 55 allographs out of 82 demonstrated by the child were acceptable considering our threshold (with a progressive improvement from 13 out of 28 in the first session up to 26 out of 29 in the last session).

As soon as Thomas understood that the robot was only accepting well-formed allographs, he started to focus on it and he would typically draw 5 or 6 times the number before actually sending to the robot (the tablet lets children clear their drawing and try again before sending it). According to the therapist, it was the first time that Thomas corrected himself in such a way: he mad the effort to take into account how *another agent* (the robot) would interpret and understand his writing. Figure 2.8 shows how he gradually improved his demonstrations for some numbers, according to the metric we used to make the robot accept/refuse samples.

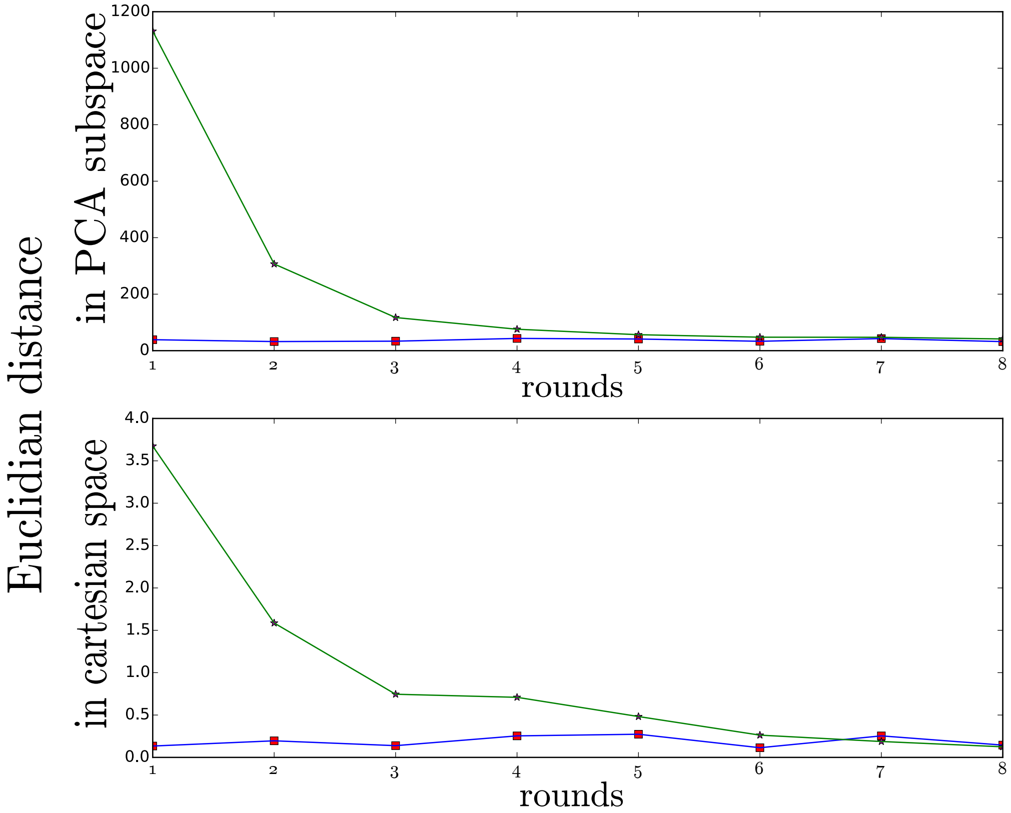


Figure 2.7 – Two metrics to assess the handwriting progresses: Euclidean distance between demonstrations and templates in the subspace of the number dataset (top figure) or in Cartesian space (bottom figure). Green lines represent the robot performance, blue lines performance of the child. The round IDs correspond to the demonstrations pictured on Figure 2.6.

Since the robot's handwriting started from a simple primitive (a stroke), each time Thomas succeeded to have his demonstrations accepted by it, the robot's improvement was clearly visible (as measured in Figure 2.7). This led to a self-rewarding situation that effectively supported Thomas' commitment.

2.5.3 Case study 3: when children evaluate the robot

Context

Each of previous studies was specifically adapted to a particular child: we relied on two different designs in order to sustain each child's commitment. In this new experiment, we conducted a study with eight children using a single experimental design. The children all have in common difficulties to learn cursive writing but the nature and magnitude of these troubles are significantly different from one child to another. Valerie (7 years old), Antoine (6.5) and Johan (7) are under the care of an occupational therapist. Emilien (8) and Mathieu (7) are repeating their school year because of writing. Marie (6) and Adele

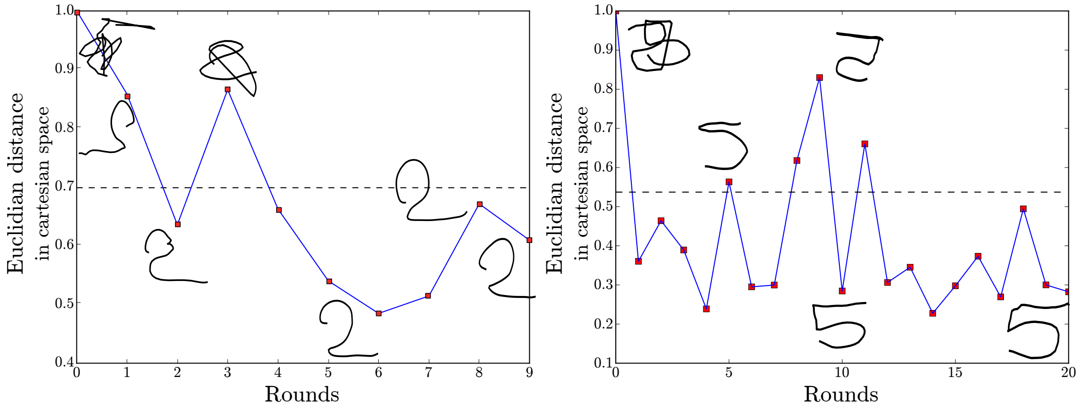


Figure 2.8 – Improvement of Thomas demonstrations for number 2 (left) and 5 (right). Thomas progressively took care of the demonstrations he was providing to the robot for those numbers. We used for this figure the same metric than the one used for the acceptance algorithm to measure distance between demonstration and templates. Distances are normalized with respect to the biggest value. The dashed line correspond to the threshold of robot's acceptance.

(8) are bottom of their respective classes in writing activities. Nicolas (7) is under the care of a neurologist, and has been diagnosed with specific language impairment. Given their school year, all of these children would be expected to know the shape of cursive letters. The experiment was conducted in collaboration with an occupational therapist.

Our goal was to study the perception of the robot's progress in children. We wanted to know how easily children were able to take the role of teachers and to detect improvements or eventual degradations of the robot's letters.

Hypothesis

Children understand their role and find motivation to teach the robot. They are able to perceive the progress of the robot, and their evaluations correlates with its handwriting performance.

Experimental design and methodology

This experiment took place in an occupational therapist clinic in Normandy, France. Over a period of two weeks, each child came three times for a one hour long session (except Adele and Marie who only attended one session). An experimenter was present to explain the rules of the game and tablet usage. As in the previous experiments, children were provided with two tablets: one to choose a word (or a single letter) to teach, one used by both the child and the robot to write. We also provided printed templates for the letters if the child asked for them.

The initial shapes used by the robot when writing were the same for all children: we used the average of a simple vertical stroke and the reference letter. In this study, we wanted the robot to be only influenced by the demonstrations provided by the child, so we did not project allographs in a subspace. The new samples generated by the robot were simply computed as the average (in Cartesian space) between the last demonstrations and the previously generated samples.

The robot was programmed to accept all demonstrations, endowing the child with the full responsibility of a teacher.

Besides, we added two buttons to the tablet interface: a green one with a “thumbs up", and a red one with a “thumbs down". Those buttons could be used by the children to evaluate the robot (the green one was for positive feedback while the red one was for negative feedback). We used it as a measure of the perception of the robot by the child: the more the child used evaluation buttons, the more he was adopting the role of the teacher, judging the robot instead of himself. Children were free to use the buttons whenever they wanted during the experiment.

Measures

As in previous studies, we recorded the timestamps of all demonstrations, the duration of demonstrations and we measured the overall commitment of the children as the number of demonstrations provided per session. We also logged all the evaluations provided by the children. The awareness of children for the robot progress is measured as the correlation between children evaluations and distances between the robot's letters and reference templates.

Analysis

Since sessions took place over only two weeks, we did not attempt to study possible handwriting remediation in children, and we focused instead on the correlation between the children's evaluations and the robot's progression. We estimated the robot's progression as the difference between an initial score (score of the first robot's attempt when the children have chosen a new word/letter to work on) and the current robot's score (after being taught by the child). The score is calculated as the average of the euclidean distance between the robot's generated letter and the reference allograph for each of the letters of the word. The reference letters where manually created beforehand, based on typical cursive letters template[4]. At every turn, we associate two values: the current score of the robot, and the child's immediate feedback (+1 if the child pressed the green button, -1 if he pressed the red one, 0 if he did not press any button during the round). We only keep rounds with feedback (*i.e.* a non-zero grade) and computed a Pearson's correlation between the robot score and the child feedback.

[4] http://www.education.com/slideshow/cursive-handwriting-z/

Table 2.2 – Feedback from the children to the robot. *#Demo* denotes the average number of demonstrations per session provided by the child; *#Pos* and *#Neg* the total number of positive (resp. negative) feedbacks they provided. r (robot) is the correlation coefficient between the feedback provided by the children and the performance of the robot. r (child) is the correlation coefficient between the feedback provided by the children and their own progress.

Child	**# Demo**	**# Pos**	**# Neg**	r_{robot}	r_{child}
Valérie	42	24	6	0.25 **	0.14 *ns*
Émilien	74	20	9	0.06 *ns*	0.02 *ns*
Mathieu	43	10	3	0.23 **	0.21 **
Nicolas	38	16	4	0.31 ***	0.20 **
Johan	32	10	5	0.10 *ns*	0.03 *ns*
Antoine	27	10	3	0.20 *	-0.02 *ns*
Adèle	35	4	2	0.28 *	0.30 **
Marie	40	5	1	-0.02 *ns*	0.13 *ns*

Results

All children maintained their engagement during all the sessions. They provided on average 42 demonstrations per session. All children made use of the evaluation buttons and had a strong preference to reward the robot (in total, 99 positive feedbacks and 33 negative ones were recorded). Interestingly, the time spent by the children to draw the demonstrations systematically increased from one session to the other. We interpret this result as the children being more careful and demonstrating the correct gestures to the robot in a slower fashion.

We found that five children out of the eight provided evaluations that significantly correlated with progress of the robot. The coefficients of correlation r_{robot} are reported in Table 2.2.

We also computed the correlation between the children's evaluations and their own progress. The analysis was conducted in the same way, using distances between the children's demonstrations and reference allographs as a progress score. The evaluations of three out of the five children whose evaluations correlated with the robot's progress, were also significantly correlated with their own progress (r_{child} in Table 2.2). For those children, it seems that the robot was reflecting their own performances, and while they were judging the robot positively (three times more positive feedback than negative feedback), they were actually evaluating themselves.

2.5.4 Induced modelling

In the first two experiments involving Thomas and Vincent, we observed the emergence of a strong complicity between the child and the robot. Vincent even kept sending handwritten letters to the robot months after the experiment. This is explained by the strong “pretend" effect induced by the scenario. In a poetic perspective, one could say that the child *gave life* to the robot. More formally, the scenario helped the child at constructing an exaggerated model of the robot, in which the robot was given beliefs, desires and emotions while it was just automatically repeating scripted sentences and imitating the child’s handwriting. In psychological terms, the child had a theory of the robot’s mind. Unfortunately, this modelling was unidirectional: the robot had no other model of the child than it’s handwriting trajectory perceived as a vector of points, and was not aware at all of the child’s modelling of itself. Beside, our efforts in the perspective to induce such an effect were considerable. In a sens, we compensated the lack of social awareness in the robot by adapting the scenario to the child’s responses.

In the last experiment, we asked children to provide a feedback to the robot’s performances. We viewed that sometimes this feedback was correlated with the robot’s progress, and sometimes it was even correlated with the child’s own progress. One could explain this phenomena by supposing that the child modelled the robot as a projection of himself, suffering similar failures at handwriting.

These observations lead us to imagine a robot able to play with such modelling induction in order to promote both an adaptive scenario promoting the “protégé" effect and the self-correction effect. This would strongly facilitate the setup of the activity and could improve the pedagogical benefices at the same time.

2.6 Online measures from face tracking

In the perspective to catch the child’s modelling of the robot, our first step was to simply detect what the child was looking at. Indeed, the visual behaviour already provides a rich quantity of information: the commitment can be derived from the time looking at the activity’s devices and agents, while disengagement can be detected from the frequency of gazes toward a door or a window. Also, the pretend effect – or simply the importance accorded to the robot by the child – can be inferred from the time spend by the child at looking at the robot versus the time spend by the child looking at the tablet.

Head Pose Estimation

We derive the visual field of attention from the head pose. Our technique only involves a single monocular RGB camera used for facial feature extraction, and a static simplified 3D mesh of a human head. 68 facial features are extracted using a fast template-based face alignment algorithm by Kazemi and Sullivan [Kazemi and Sullivan, 2014], as implemented in the open-source `dlib` library [King, 2009]. Eight of these features (chosen to be far apart and relatively stable across age and gender) are then matched to their 3D counterparts (Figure 2.10) and we rely on an iterative *PnP* algorithm (OpenCV’s

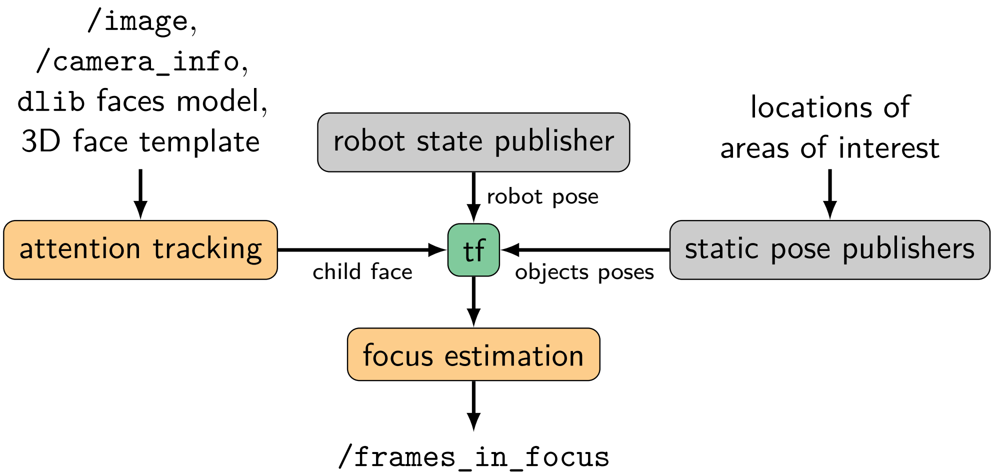


Figure 2.9 – ROS nodes involved in the Visual Focus of Attention (VFoA) estimation (orange nodes were specifically developed for this work).

implementation) to compute the translation and rotation of the head with respect to the camera frame. With this approach, knowing the intrinsic parameters of the camera (calibrated camera) is required for an accurate estimation of the absolute 3D localization of the head.

Besides being fast, the face alignment algorithm has been found to perform well in terms of robustness, including in a range of difficult situations encountered in field experiments, like partial occlusions or large head rotations (we have measured the default `dlib` model to be able to track a face with rotations up to ±40° horizontally and ±30° vertically). Figure 2.11 shows a selection of such difficult scenes with one child.

System Implementation

The experiment was carried out with an Aldebaran NAO robot, using ROS as a middleware to build the attention estimation pipeline (Figure 2.9). Head pose estimation, presented builds on the `dlib` and OpenCV libraries; the pose transformations are handled by the ROS TF library. The same TF library is used to represent the possible point of interests as individual frames: an object is considered to be in focus when its frame lies within the field of attention of the participant (Figure 2.12). The implementation is open-source and available at `https://github.com/chili-epfl/attention-tracker`.

Field & Focus of Attention

We model the field of attention as the central region of the field of view. The field of view itself is approximated to a cone spanned from the nasal depression (sellion) of the human face. Different dimensions for the human field of view can be found in the literature:

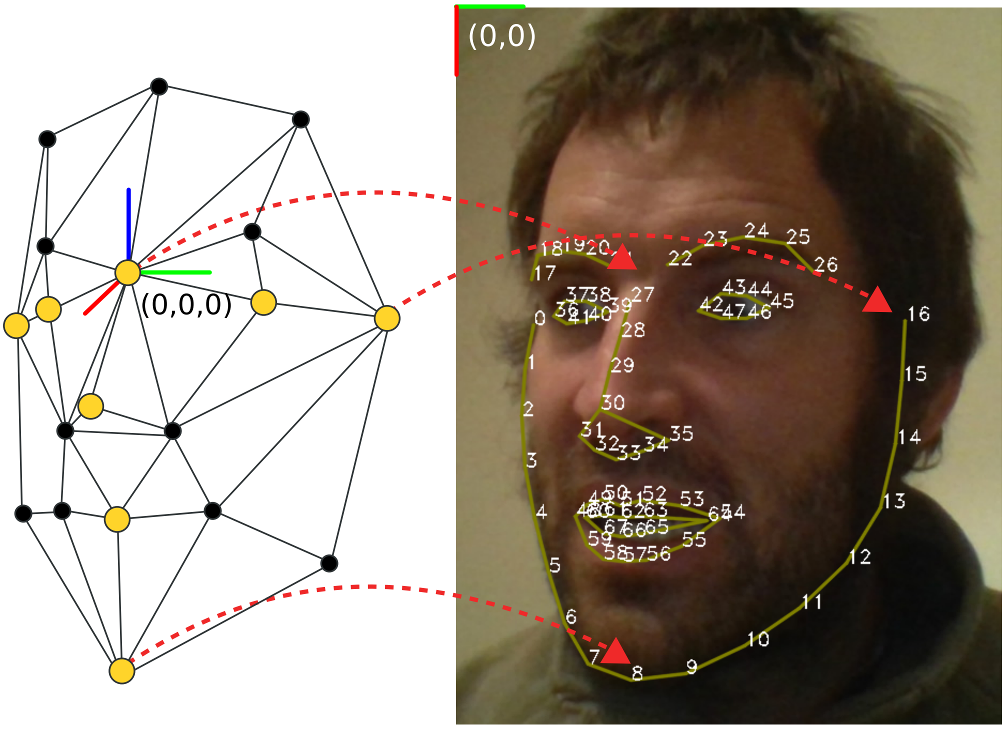


Figure 2.10 – The 6D head pose is estimated by fitting a 3D model of an adult head (left) onto the detected 2D features of the face (right). We rely on an iterative *PnP* algorithm, using 8 correspondence pairs (three are depicted: the sellion – the nasal depression –, the left tragion and the menton). The 3D origin of the head is set at the sellion.

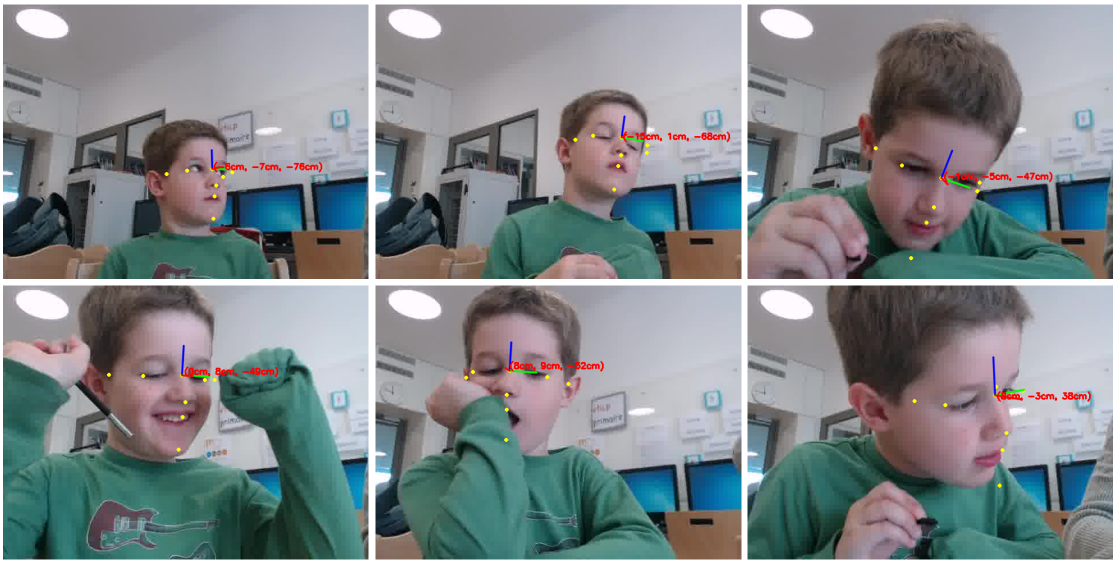


Figure 2.11 – Head pose results on images captured during a field experiment. Detection of face features (and therefore, estimation of the pose) is robust to significant occlusions and face rotations.

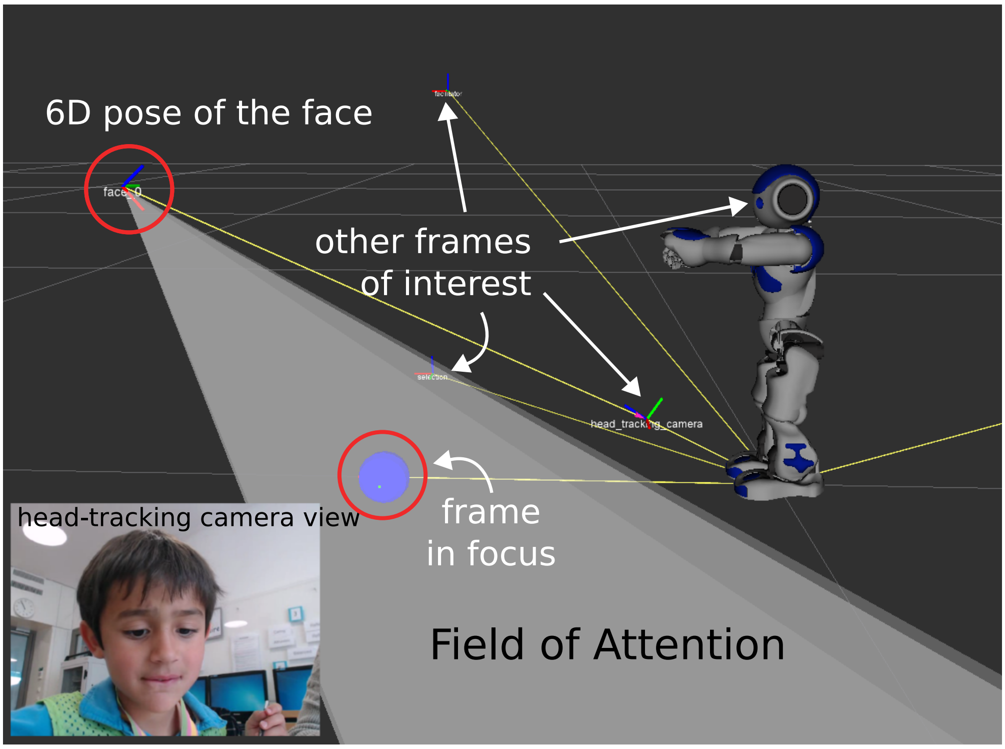


Figure 2.12 – Screenshot of the real-time attention estimation system. The visual field of attention is approximated to a 40° cone, spanning from the head's sellion. The objects whose 3D pose intersect with this cone are considered *in focus*.

Holmqvist [Holmqvist et al., 2011] models it with an horizontal aperture of ±40° and a vertical aperture of ±25°, while Walker [Walker et al., 1980] for instance suggests 60° up, 75° down, 60° inwards (towards the nose) and 95° outwards. Previous work on visual perspective taking for social robotics [Sisbot et al., 2011] model the field of attention as a cone of 30°. We retained in this work a slightly wider aperture of 40°. We then approximate the visual *focus* of attention (VFoA) of the human to the objects which lie inside this field of attention (Figure 2.12). At a given time, more than one object can therefore be *in focus*.

Our implementation has two limitations: objects are approximated to points (they are considered in focus if their *origin* lies in the field of attention), and we do not check actual visibility: one object could be hidden by another, it would still be considered as in focus. We did not address these limitations since our experimental setup (involving relatively small objects with no occlusions) did not necessitate it. Techniques for more accurate assessment of the visual perspective of the human peer can be found in [Sisbot et al., 2011] for instance.

Within these limitations, computing if object $A(x_A, y_A, z_A)$ is in the field of attention of the human requires first to transform the coordinates $A(X_A, Y_A, Z_A)$ into the frame of the face, and then to verify the simple inequality $\sqrt{Y_A^2 + Z_A^2} < tan\left(\frac{fov}{2}\right) \cdot X_A$ (with fov the aperture, and assuming that the main axis $\vec{x}$ of the field of attention points forward).

Our approach assumes that the pose of the objects of interest are available to the system: as described in subsection 2.6, our implementation relies on the ROS TF framework to manage and make available to all software modules the list of poses of existing objects (represented as *frames*), and dedicated perception modules are in charge of publishing up-to-date informations regarding the location of the objects of interest (the so-called *situation assessment*). Due to the nature of the experiment, most of the points of interest considered for the experimental validation presented hereafter are static with respect to the robot, thus simplifying the scene perception.

2.6.1 Experimental Validation

As presented above, we use the 6D head pose as an approximation of the actual gaze direction, and we further approximate from here the participant's field of attention. The assumption that such an approximation of the field of attention allows to derive the actual focus of attention needs to be validated experimentally. Our proposed experiment involves child-robot interactions in the context of handwriting remediation. This subsection details the experimental procedure and presents our results.

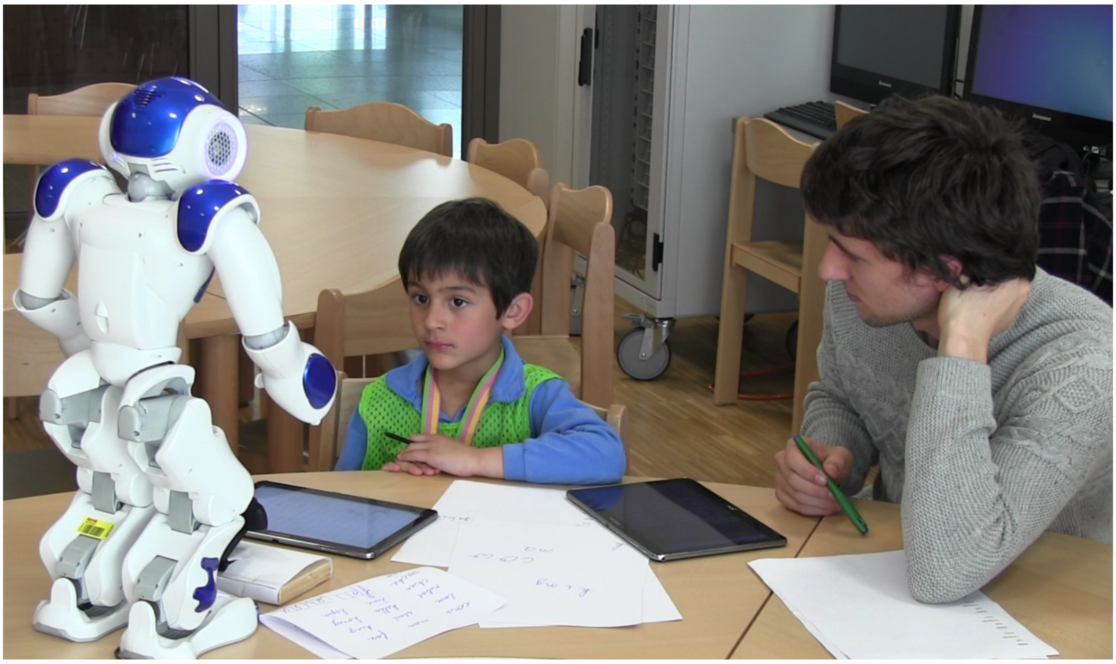


Figure 2.13 – Picture of the interaction with one of the children.

Experimental Procedure

The subjects were typically located 50 cm away from the robot with the primary (writing) tablet in front and the secondary one 30 cm to the left of the first one. The facilitator was located about 60 cm to the left of the subject. Finally, two observers (visible by the child) were located further away from the interaction field. Figure 2.13 shows accordingly the location of main areas of interest (the two tablets, the robot and the facilitator).

The dependent variable is the measurement of the participants' VFoA, assessed in terms of what the attentional targets of the child are over time. The face of the child is acquired through a fixed webcam (Logitech c920), placed on the table (see Figure 2.1), and the attentional targets are then computed as presented in subsection 2.6.

Six children (ages 5 to 6, 3 boys, 3 girls, none wearing glasses) were enrolled for this study. The study took place at school, in an isolated room (the computer lab). The participants were chosen by the teacher, and would come one after the other to interact with the robot (duration: $M = 19.6$ min, $SD = 1.58$).

The interaction is organized in rounds of writing: during a typical round, the child requests the robot to write something (a single letter, a number, or a full word), and presents a tactile tablet (equipped with a custom writing application) to the robot. The robot "writes" on the tablet by drawing in the air the letters that are displayed on the screen by the tablet application; the child then pulls back the tablet, corrects the robot's attempt by writing on top of or next to the robot's writing, and "sends" his/her demonstration to the robot by pressing a small button on the tablet. The robot learns from this demonstration and tries again. The child continues the turn-taking until they decide to train the robot on another word. In total, the children performed on average 12.16 ($SD = 2.61$) rounds of writing (complete details on the rationale and implementation of this experiment can be found in [Hood et al., 2015c]).

Once per interaction, the robot interrupts the handwriting task to tell a story (taking about 2 min), and the turn-based hand-writing task continues afterwards. The intended purpose of the story-telling episode is to break the routine of the writing turns by creating a surprise, and thus, to elicit a different set of attention behaviors from the child.

Data Collection & Analysis

Successful detections of the head, and, when detected, the attentional targets of the children as estimated by the robot, were logged during the experiment (in total, $6 \times 19.6 = 117.6$ min of interaction). The only post-processing consisted in filtering out gaze shifts (short episodes – below 500ms – between two attentional targets).

We video-recorded the interactions, and performed a *post-hoc* manual coding of the focus of attention (24% double-coded, Cohen's $\kappa = 0.91$, high reliability). The manual coding forms our attentional *ground-truth*.

To assess the accuracy of the attention estimation by the robot, we computed over time the overlap between the ground-truth and the robot's estimate and the inter-rater agreement (Cohen's κ). The periods where the head was not detected were *excluded* from the agreement computation: at such times, the robot explicitly knows that it can not estimate the focus of attention, and as such, we do not consider that it *wrongly* estimates the focus.

Results

The main results are reported in Table 2.3 Figure 2.14 further gives a concrete picture of the ground-truth *vs.* computed attentional targets for subject 4 (the subject with the *least* successful tracking).

During the whole interaction, the head pose of the children was consistently tracked, 86% of the time in average, $SD = 3.0$. While this high score is expected for a face-to-face interaction with a static head-tracking camera (meaning that the child head would remain in the field of view of the camera most of the time), this is still comforting in terms of suitability of our approach for head pose estimation with children in field experiments of this kind. Expectedly, the primary causes of lost head pose were occlusions with the hands (similar to the middle-bottom picture in Figure 2.11), close proximity with the

Table 2.3 – **Attention tracking accuracy**. *Head pose tracking* is the percentage of total time of successful detection of the head pose; *Agreement* is the percentage of matching time between manually annotated focus of attention (ground-truth) and robot's computed focus of attention. Total duration: 117.6 min.

Subject	1	2	3	4	5	6	**M**	*SD*
Head pose tracking (%)	88.2	83.5	90.5	83.1	87.9	85.0	**86.4**	*3.0*
Agreement (%)	58.9	67.1	79.2	48.3	65	77.1	**65.9**	*11.5*
Cohen's κ	0.48	0.56	0.68	0.26	0.47	0.68	**0.52**	*0.16*

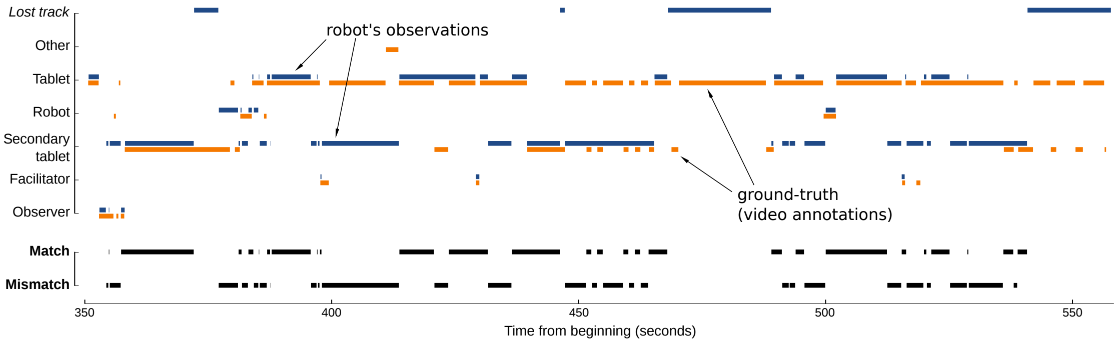


Figure 2.14 – **Comparison of computed focus of attention *vs.* ground truth** during a face-to-face child-robot interaction (subject 4 in table 2.3, 3.5min-long excerpt). In blue (top lines), the focus of attention as computed by the robot; in orange (bottom lines), the focus of attention as manually annotated (ground-truth). The bottom subsection shows agreement between both (whenever the head is detected).

tablet while writing, and gaze directed to the facilitator (who was sitting directly on the left of the child, Figure 2.1).

In terms of attention tracking, Cohen's κ values are between 0.47 and 0.68 with one subject resulting in significantly worst tracking, at 0.26. While the interpretation of Cohen's κ is subject to discussion (the number of the coded values – in our case 6 – and the distribution probability of values – in our case, values are *not* equiprobable – are factors impacting κ independently of the level of agreement), the levels of agreement are *moderate* to *substantial*, with one subject only showing *fair* agreement [Landis and Koch, 1977]. Further analysis of the videos shows that the child with the lowest level of agreement was particularly quiet and would indeed rely more on the eyes to direct his gaze than the other children, thus leading to a less accurate estimation of his focus of attention.

The next subsection builds upon this technique for real-time estimation of the focus of attention: by comparing the focus of attention with the set of attentional targets *a priori* expected by the robot, we can estimate to what extent the user is "with" the robot.

2.6.2 With-me-ness

Concept & Calculation

The concept of *with-me-ness* has been introduced in the field of *Computer Supported Collaborative Learning* (CSCL) by Sharma *et al.* in [Sharma et al., 2014]. They introduce this concept in an attempt to answer a recurrent teacher's question: *"how much are the students with me?"*. They distinguish what they call *perceptual with-me-ness* (the student follows what the teacher refers to with deictic gestures) from *conceptual with-me-ness* (the student follows what the teacher refers to verbally), and they show in an eye-tracking study involving video lectures (MOOCs) that *conceptual with-me-ness* in particular correlates with better learning performance. This also relates to the concept of gaze cross-recurrence that has been shown to reflect the quality of the interaction [Jermann and Nüssli, 2012] in collaborative learning tasks.

They define *conceptual with-me-ness* as the normalized percentage of time during which the student's gaze overlapped the areas of teaching slides currently referred to by the teacher. In order to apply it to human-robot interactions, we propose to extend this concept, and to define *conceptual with-me-ness* as the normalized ratio of time that the human interactant focuses its attention on the attentional target expected by the robot for the current task (or sub-task).

Algorithm 2 provides a formal way of computing the level of with-me-ness $\mathcal{W}$ between two time points $[t_{start}, t_{end}]$. A notable difference with the original definition by Sharma *et al.* is that, at a given time t, the task $task(t)$ performed by the robot may elicit more than

Algorithm 2: Computation of *with-me-ness*. d_w stands for the duration the human is actually *with* the robot, while d_e stands for the total time where the human would be *expected to be with* the robot, $task(t)$ represents the task performed by the robot at time t (possibly none), $F(task)$ represents the (possibly empty) set of expected attentional targets associated to task $task$, $f(t)$ represents the actual focus of attention of the human measured at time t. $\mathcal{W}_{[start,end]}$ represents the level of *with-me-ness* from t_{start} to t_{end}.

```
$d_w, d_e \leftarrow 0$
$t \leftarrow t_{start}$
while $t \leq t_{end}$ do
    if $task(t) \neq nil$ and
     $F(task(t)) \neq \varnothing$ and
     $f(t) \neq nil$ then
        if $f(t) \in F(task(t))$ then
            $d_w \leftarrow d_w + \delta_t$
        $d_e \leftarrow d_e + \delta_t$
    $t \leftarrow t + \delta_t$
$\mathcal{W}_{[start,end]} \leftarrow \frac{d_w}{d_e}$
Return $\mathcal{W}_{[start,end]}$
```

one attentional target; thus, at a given time, more than one location can be regarded as possible *expected* focuses of attention for the human. For example, a robot which is writing, could typically elicit gazes to its hand as well as to its head. A human looking at either of these locations would be considered to be *with* the robot in terms of interaction[5]. Also notable, we exclude from the computation of $\mathcal{W}$ all of the periods of time where the user's focus of attention can not be estimated (typically because the user's face is not visible at those times).

Experimental Measure & Interpretation

Over the course of the experiment presented in subsection 2.6.1, the robot controller would associate a set of expected attentional targets to the phase of the interaction (Table 2.4). For instance, while the robot was waiting for the child's handwriting demonstration ("*Waiting for feedback*"), the expected attentional target of the child was the tablet (since the child was supposed to write there) or the secondary tablet (that displayed a template of the word, used as a reference by the child). These expected targets (green lines on Figure 2.15) form the robot's attentional *a priori* knowledge and are used to compute the with-me-ness. With-me-ness can be calculated over the whole interaction or over shorter time windows. With-me-ness over the whole interaction for the six subjects is reported in Table 2.5. The Pearson's correlation with the ground-truth is $r(4) = 0.46$ (significance not computed due to small sample size). Shorter time windows are interesting for two

[5] Considering a probabilistic model of attention expectations (an attention distribution) would be an interesting extension of this metric.

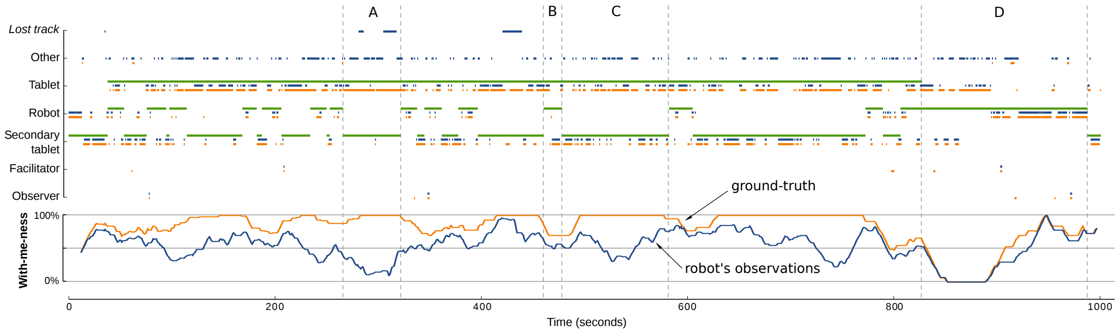


Figure 2.15 – **With-me-ness**. Evolution of the level of *with-me-ness* over the whole ≈17min long interaction of subject 2. The top chart is similar to Figure 2.14 with the *expected* attentional targets added in green. The bottom diagram represents the instantaneous level of with-me-ness over a sliding window of 30 seconds. The blue line is the with-me-ness as estimated by the robot, the orange line is the with-me-ness computed from manually annotated attentional targets. Pearson's correlation between both series for this subject: $r(973) = 0.58, p < .001$.

purposes: to analyse the level of with-me-ness in relation to specific interaction episodes; to allow a measurement of with-me-ness by the robot *over the course* of the interaction (*in-the-moment* measurement) – in the latter case, one may typically want to consider a sliding time window.

The with-me-ness plotted at the bottom of Figure 2.15 is in fact computed on a sliding window of 30 seconds, and thus gives a picture of "how well the child is following the robot's expectations" at that time. As seen, the with-me-ness computed at run-time by the robot (blue line) is generally lower than the ground-truth (orange line, based on video-annotations), and sometimes quite off, such as during episode marked "A": during that phase, one can notice that the attention is mostly directed to undefined target Other, likely a consequence of inaccurate head detection. This kind of error (inaccurate head pose estimation) is the main source of discrepancy between the ground-truth and the attention distribution measured by the robot: ignoring all the episodes where the child's gaze is measured to be directed to Other, we indeed obtain levels of with-me-ness close to the ground-truth (over the six subjects, $M = 87.5, SD = 4.6$).

A chart like Figure 2.15 remains a useful tool to analyse the interaction, and several observations can be made from it: the green lines represent how the robot imagine, at a given time, the attention distribution of the child. They also provide an accurate picture of the overall turn-taking as viewed by the robot: for instance, the episode "B" on Figure 2.15 corresponds to one of the "*Robot writing*" episodes, surrounded by "*Waiting for feedback*" phases like "C"; episode "D" corresponds to the story telling; etc. In terms of interaction, the large variance of the duration of these phases reflects the fact that this child would sometimes take a lot of time to send feedback to the robot, and sometimes, on the contrary, be very quick.

Looking at the ground-truth focus of attention (orange lines), the first striking observation is that this child did generally *closely* follow what the system was expecting: in that regard, it seems that the child was very much engaged in the interaction (we discuss in the next subsection the exact relationships between with-me-ness and engagement). The only major exception is the story-telling phase (episode “D”): the child was seemingly not captivated by the first half of the story, and their attention was not directed towards the robot (this actually matches the observed behavior of the children who mostly found the story boring).

Another interesting observation pertains to the facilitator: as one can see, this child only rarely turned to the facilitator, possibly indicating that the interaction and the task were meaningful and easy enough for him to follow alone.

More subtle patterns and events can also be observed: for instance, during the feedback phases like episode “C”, we can notice numerous recorded gaze shifts between the tablet (where the child writes) and the secondary tablet (that showed a template of the word). The episode “B” (robot writing) is also interesting: the child did not look at the robot, and instead remained focused on the secondary tablet. This situation is typically useful for the robot to detect as it may want to adapt its behavior to recover the child’s attention.

2.6.3 Discussion

Head Pose to Assess Attention: is it Relevant? We already stated the main limitations of our approach to estimating the focus of attention: eye gaze information is neglected and we do not perform visibility check of the in-focus objects (we simply approximate them to their origins, ignoring possible occlusions).

While the first issue is shared with most of the other vision (2D or 3D) or motion capture techniques for real-time gaze estimation found in robotics, our results are positive: we show that relying purely on head pose estimation to estimate gaze direction leads to real-world measures that are worth being considered and used. They may not match manual annotations, but they are definitely a valuable *in-the-moment* input for the robot. For certain children, we reach levels of accuracy traditionally considered as good.

Our approach relies on a simple, non-intrusive sensor (a RGB camera by the robot) and an open-source, fast pose estimation algorithm : we hope that this may contribute to the widespread adoption of such a technique on a range of robots, including the relatively common NAO platform.

With-me-ness: yet another metric of engagement? Borrowing the neologism from the field of CSCL, we have also introduced in this article *with-me-ness* as a measure of “how much the user is *with* the robot during a task”. This can be acquired over the course of the interaction, thus providing the robot with a real-time metric for a relatively high-level social construct, undoubtedly related to engagement.

One may reasonably wonder how different *with-me-ness* is from *joint attention* on one hand, and from *engagement* on the other hand. With-me-ness is related to both, with however noteworthy nuances: (Triadic) *joint attention* is understood as the cognitive realization of a shared attention to an object, itself building on a shared perception of that object (*i.e.* joint attention builds on a *perceptual* alignment of two agents). *Conceptual with-me-ness* as proposed by Sharma *et al.* in [Sharma et al., 2014] is on the contrary *referential*: "you are *with* me if you focus on what I refer to, either explicitly or implicitly". We understand it here in a slightly broader sense that reflects the *interaction*: "you are *with* me if you focus on what is important for the interactive task at hand."

On the other hand, with-me-ness is only a precursor of engagement: it does not say much about the *cognitive* commitment of a user to an interaction. A user may closely adhere to the injunctions of the robot (or, actually, of the experimenters), with thus high levels of with-me-ness, without being *engaged* in the interaction. This is typically seen in child-robot interaction: children will attempt to closely follow what they are asked to do – which may *look like* they are engaged in the interaction – while they merely *obey orders*.

Compared to engagement, one of the strengths of with-me-ness is its specificity: it is well-defined, we can formalize it, and as such, it is valuable to assess and compare how users are willing or able to interact with a robot. We have hopefully demonstrated in this article that with-me-ness is an operational *in-the-moment* metric that can also be used as a real-time feedback to the robot controller to build richer, more adaptive interactive behaviors for our robots.

Note however that, besides the actual focus of attention, the mapping *phase/expected attentional target* (*i.e.* our Table 2.4) is a critical piece of information to interpret with-me-ness. The mapping is typically built by a domain expert, and is often subject to debate (for instance in our experiment, one could argue that during the "Waiting for feedback" phase, the child could have gazed toward the robot to make sure the robot was paying attention, and consequently, robot should be added to the expected target). For this reason, the chosen mapping should always be reported along with the computed with-me-ness levels, and with-me-ness should not be reported as an absolute metric, but rather as a mean of comparing different interactions within the same study.

2.7 Spatial arrangement

In this section, we study the influence of the reciprocal spatial position of a child and a robot on the modelling of the robot by the child. We want to know id the child is more likely to perceive the robot as a peer or as a student depending if the child is facing it or if they are working side-by-side.

2.7.1 Context

Case studies presented in 2.5 showed that children were able to stay engaged in long term interactions with repeated sessions within the CoWriter activity in real pedagogical/therapeutic contexts. These works suggested a positive effect on the extrinsic motivation of the children when practicing their handwriting, thanks to the protégé effect.

Authors in [Matsuzoe and Tanaka, 2012] had a similar approach to learning by teaching with their Care Receiving Robot, who was being taken care of and taught by a child using physical interaction. In this study, authors chose to investigate handwriting (or drawing shapes) as well, but with a more physical based approach. In their experimental setting, the child would teach handwriting to the robot by placing himself behind the robot and moving it's hand. This study varied from other works in educational human-robot interaction in a way that the child was not only facing the robot but would act as a care-taker rather than a teacher.

As robots are entering our living space, they must adapt to our social norms. These norms vary from politeness to unspoken social rules (as for instance, the personal space of a person). In home environments, robots will be expected to perform their functions in a manner that is clearly understandable and predictable by the humans around. This requires adaptive personalization of the robot to the individual needs of the humans, but also to the task being currently performed. Some previous studies showed that the spatial setting with a robot was also a way to convey non-verbal messages and it serves to influence the relationship with the user [Takayama and Pantofaru, 2009, Kristoffersson et al., 2013]. Spatial arrangement is still a factor relatively unexplored in HRI and its influence on the interaction is still unclear.

In this study, we explore the effect of spatial arrangement on the child-robot interaction within the CoWriter activity.

2.7.2 F-Formation

Facial formation or F-Formation, describes the spatial arrangement of a group of at least two individuals, interacting around a closed space (the o-space) in which they have an easy, symmetrical and exclusive access [Kendon, 1990]. For example, both side-by-side and face-to-face formations in the CoWriter interaction are F-Formation, where the o-space contains the two tablets.

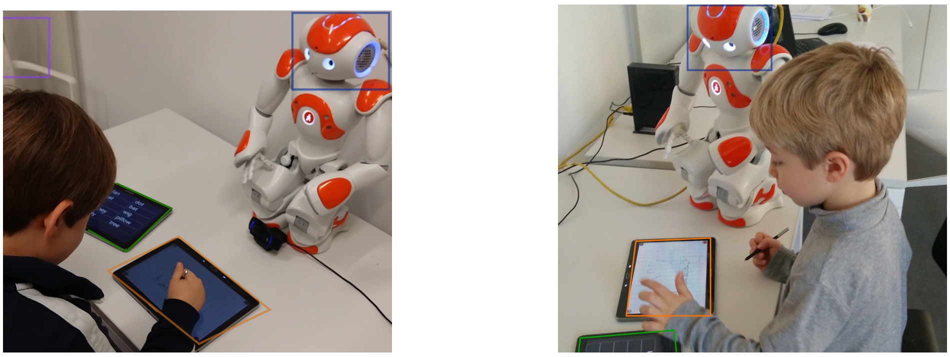


Figure 2.16 – Face to face

Figure 2.17 – Side by Side

Figure 2.18 – Experimental set-ups showing the *with-me-ness* targets in the rectangles: orange - the writing tablet, blue - the robot's head, green - the tablet used to select a word to teach to the robot

2.7.3 Method

The experiment took place in a primary school in Switzerland where children are taught in English. In this experiment, we targeted children aged 5 y.o. who start handwriting, but do not typically master it yet. The children interacted with the robot under two conditions of the F-Formation in a counterbalanced manner.

Our goal was to investigate the effect of spatial arrangement on the interaction. We expected that children would give better feedback(see 2.7.3) to the robot when teaching it in a side-by-side configuration for several reasons. The side-by-side arrangement corresponds to a cooperative arrangement which is close to a peer teaching setting, unlike the face-to-face teaching arrangement, which is more frequent seen in competitive or conversation tasks (closer to a teacher-student relationship). Also, in the side-by-side formation, the robot and the child have the same visual perspective of the shared tablet on which they write. Perspective sharing is an ability that facilitates mutual understanding [Berlin et al., 2006]. Hence by having the robot and the child side-by-side, higher mutual understanding would be expected.

Participants and Apparatus

12 subjects (six girls) from the same classroom (aged 5 to 6 y.o.) participated to the within subject study. The two considered F-formations for this experiment are presented in the Figure 2.18: face-to-face 2.16 and side-by-side 2.17. Children were presented the two conditions sequentially with and interval of three days in a counterbalanced manner.

Apart from this change in the spatial setting, the interaction was kept the same. The children were told they had to teach the robot how to write some words. We briefly presented the two tablet interfaces and the interaction started. Any word from a list displayed on the selection tablet could be picked by the child. As the robot would start to write this word. It was set to be a very bad writer at the beginning of the first session for each child. The child could then give a feedback to the robot by pressing thumbs up or a thumbs down buttons how many times they wanted. The child would then use a pen and demonstrate how to write the word and the robot would then rewrite the word using the demonstration of the child. The generated writing of the robot was computed to be halfway from its previous writing state and the new demonstration provided by the child.

Several hypotheses were made concerning the influence of the spatial arrangement on the interaction. We expected that the gaze behavior of the child would vary according to the spatial configuration. More gazing at the robot's head would be expected in the face-to-face condition. Our research question was to measure the degree to which this also influenced the way children behaved as a teacher (were they more or less severe with the robot facing them). As children give a feedback to the robot for each demonstration, we intend to measure if there is any difference between vis-a-vis and side-by-side regarding this feedback (does the side-by-side condition trigger more positive feedback? or more appropriate feedback?).

The degree of engagement of the child in the task can also be influenced by the arrangement. For that particular aspect, we will measure the number of repetitions of each word, as well as the with-me-ness, which is discussed in the following subsection. Since children were quite young, we choose to not use any self-reported measures or questionnaires.

With-me-ness

The with-me-ness, introduced in HRI by [Lemaignan et al., 2016a] and described in section 2.6, helps to set specific targets during each state of the interaction and to determine whether the user is looking at one of these attentional targets or not. This measure allows us to see if the child is engaged in the interaction and is looking at the tablet or the robot's head when he/she is expected to (according to the task). Indeed, in our learning by teaching activity, the robot has also a hidden pedagogical role. It's progress aim actually to make the child practice and think about his/her own way of writing. In that sense, we can set attentional targets as proposed by [Sharma et al., 2014] when the *with-me-ness* was first introduced to measure learner's attention to teachers in MOOC videos.

This is actually very close to the notion of synchrony already studied in HRI [Delaherche et al., 2012], where bounding between individuals is reflected by their ability to synchronize in the task (look at the same time at the same targets).

In this experiment the visual targets were: "the observer"(a teacher or a teacher assistant),"the experimenter", "the selection tablet"(the tablet used to pick a word) and the "tablet". The with-me-ness is initially set to 0.5 and takes values from 0 to 1. According to the state in which the activity is, (robot is writing, child is writing,. . .) the with-me-ness will be increased if the child looks at a target that is in the set of task-related targets (expected within this particular state of the activity). We record these targets and the with-me-ness at a frequency of about $1Hz$. The evolution of the with-me-ness is computationally attenuated in order to remove noisy data (by using the weighted cumulative of the with-me-ness value).

The targets were defined according to their spatial relation with the camera used (placed on the table at the robot's feet). The position of the targets was changed according to the F-formation condition, but the camera stayed at the same position Figure 2.18 shows these targets for the two conditions.

Reward Mechanism

The tablet interface on which the child and the robot practice their handwriting shows two buttons that allow the child to give a positive (green thumb-up) or negative (red thumb-down) feedback to the robot's handwriting. After every trial of the robot, the child could click on these feedback buttons as much as he/she wants. We monitored each of these clicks.

These clicks aimed to assess the child's perception of the robot's progress. When converging to a better writing we expect the child to give better feedback. However, these buttons could also be used by the child as an encouragement method and children could give a positive feedback even though the robot didn't make progress.

Performances

As the child was managing which word the robot would learn, he/she could also provide as many demonstrations as he/she wanted. The child was also free to change the word when satisfied with the robot's performance.

Response Time and Writing Time

We recorded the time spent on the writing and the response time for each word demonstrated by the children. We also monitored the number of demonstrations provided by the child for each word. These measure were cues to how dedicated the child was in the task

Writing Score for the Robot

Since the learning algorithm took as input the child's demonstration, if the child provided repeatedly the exact same demonstration, then the robot would converge faster to his/her handwriting. This score is hence a hint on children's regularity, with the underlying assumption that the regularity in handwriting is a sign of legibility.

At each demonstration of the robot, we were calculating a *writing score*. Each demonstration was encoded as a list of seventy 2D points. This *writing score* is the euclidian distance between the demonstrated letter by the child and the generated letter by the robot.

We also computed the evolution of this score at each demonstration. We represented the evolution of the robot's handwriting with different states: 'S=': the first trial of this word by the robot, 'S-': the score is decreasing and 'S+' the score is increasing. If the child was changing a lot his/her way of writing between consecutives demonstration, the score would then decreasee. In the contrary, the regularity of the demonstration would make the score increase rapidly.

2.7.4 Results

Reward Mechanism

Children gave feedback with an average of 3 feedbacks per demonstration (i.e. per interaction loop). We summed the feedbacks for each demo with positive feedback counted as $+1$ and negative as -1. Figure 2.19 shows the average evolution along the demonstrations of the sum of feedbacks given for all children and for both spatial condition. We noticed that the feedback is first negative and grows towards a positive feedback after each demonstration. In average for all the children, the sum of feedbacks stayed negative until the 5th demonstration. Children well understood that they were teaching the robot and often gave bad scores for the first untrained trial of the robot. Children usually gave a positive feedback just before changing the word taught to the robot.

Even though the average feedback was increasing along with the number of demonstration for both condition, they don't seems to increase the same way(see Figure2.19) The effect of the F-Formation on the feedback from the child was statistically significant (Anova Repeated measures within subjects: $F_{1,272} = 4.396, p < 0.05$). As, illustrated on Figure 2.20, the average feedback per demonstration was more positive ($M = 0.03, SD = 1, N = 147$) for the side-by-side condition compared to the face-to-face condition ($M = -0.23, SD = 0.98, N = 138$).

Children kept giving feedback along the interaction and no drop in the number of feedbacks was observed during the experiment. They took their duty to teach the robot

Table 2.4 – Mapping between the interaction phases and the expected attentional targets.

Phase	**Expected targets**
Presentation	robot
Waiting for word to write	secondary tablet
Writing word	tablet, robot
Waiting for feedback	tablet, secondary tablet
Story telling	robot
Bye	robot

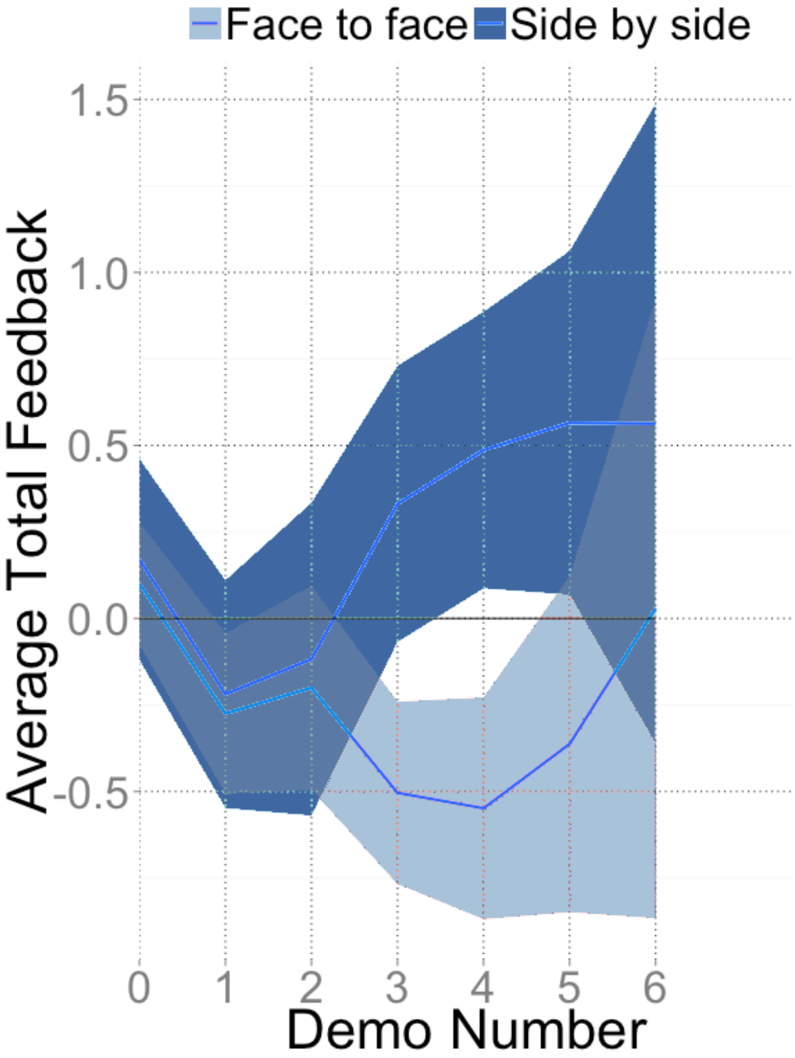


Figure 2.19 – Evolution of Feedback along the demonstration index(Mean: line , Confidence Interval 0.95: filled area)

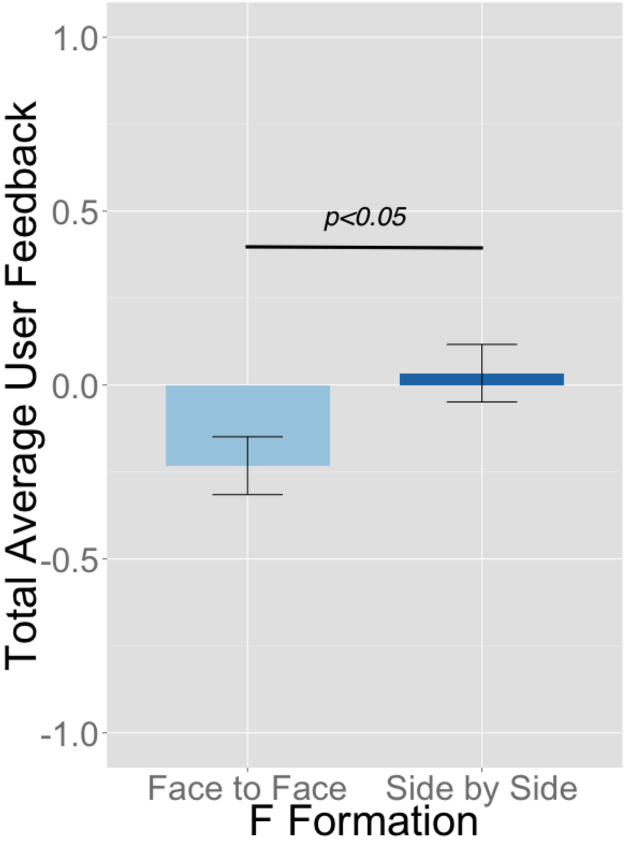


Figure 2.20 – Feedback Sum According to the F-Formation (Means and Confidence Intervals)

Table 2.5 – **Levels of with-me-ness**. For each subject, the with-me-ness level is reported over the whole interaction, either based on the annotated focus of attention (*i.e. ground-truth with-me-ness*), or based on the focus of attention measured by the robot.

Subject	1	2	3	4	5	6	**M**	*SD*
$\mathcal{W}_{g.truth}$	79.4	81.6	90.5	87.9	90.7	80.9	**85.2**	*5.1*
$\mathcal{W}_{robot}$	52.6	55.3	74.3	52.9	59.5	63.9	**59.8**	*8.3*

seriously in that way.

We also observe an order effect showing that children tended to give more positive feedback in the second session compare to the first one. This can simply be explained by the fact that the robot's learning state was progressive between the two sessions. The robot didn't start to learn from scratch in the second session but had already some knowledge from the first session with this same child.

With-me-ness

Children understood the dynamics of the interaction, as in general the with-me-ness stayed above 0.5 throughout the interaction (started et 0.5 but always finished above).

The effect of the F-Formation on the with-me-ness of the child was statistically significant (Anova Repeated measures within subjects: $F_{1,15983} = 293.2, p < 0.001$). The with-me-ness was greater($M = 0.79, SD = 0.18, N = 7722$) for the face-to-face condition compare to the side-by-side condition ($M = -0.72, SD = 0.21, N = 8267$) (see Figure 2.21). This result was expected, as the robot was facing the robot, its face was more visible for the child.

Again, we observed an order effect with the with-me-ness increasing between the two sessions. As children were more comfortable with the system, it is logic that they started to learn the dynamics of the interaction knowing when to look at the selection tablet, the writing tablet and the robot.

Response Time and Writing Time

Figure 2.22 shows on the left the average number of demonstration for each word taught to the robot for the two spatial arrangement conditions. There was no significant difference between the conditions even-though the average number of demonstration given by the child in the side-by-side condition seems higher than the face-to-face.

The center graph of Figure 2.22 shows the average response time for the demonstration provided by the children for the two spatial arrangement conditions. The response time

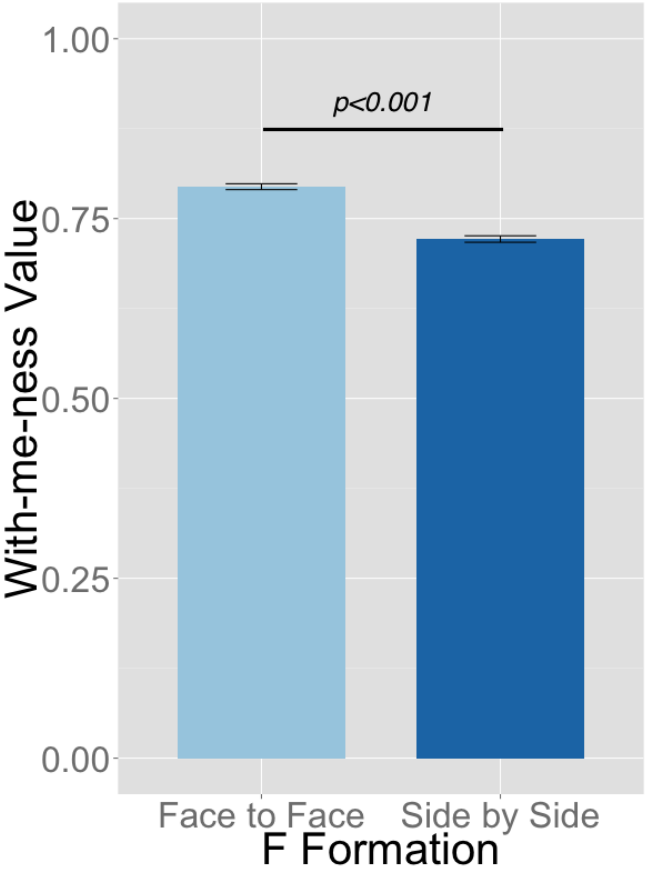


Figure 2.21 – With-me-ness According to the F-Formation (Means and Confidence Intervals)

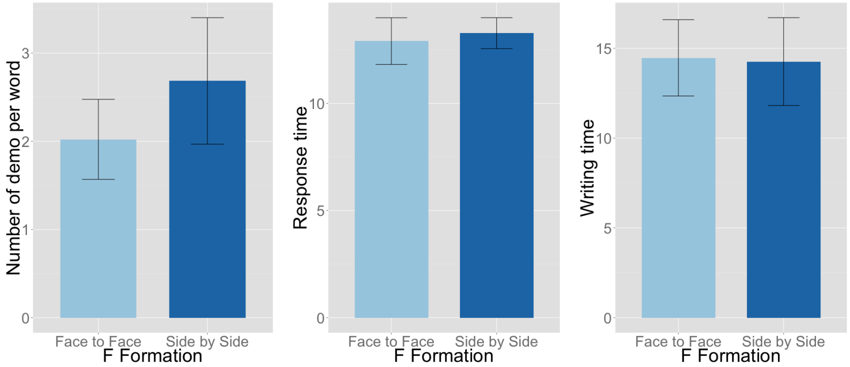


Figure 2.22 – Number of demonstration per word (left), response time (center), writing time (right) - (Means and Confidence Intervals)

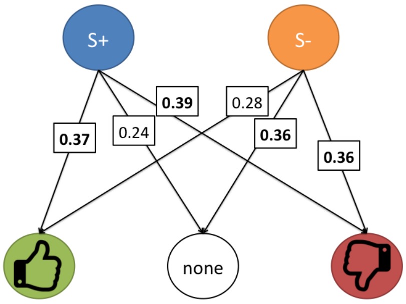


Figure 2.23 – Face to face

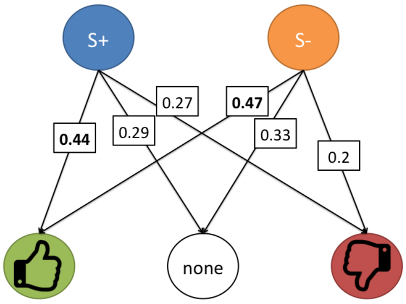


Figure 2.24 – Side by Side

Figure 2.25 – Transitions states from score evolutions ($S+$: score increasing compared to previous score, $S-$: score decreasing compare to the previous) to children's feedback (thumb up: positive feedback, thumb down : negative feedback, and no feedback given)

is the delay between the time when the robot finishes to write and the time when the child touches the tablet. There was no significant difference between the conditions.

The writing time (right graph on Figure 2.22) corresponds to the delay between the time when the robot finishes its trial and the time the child finishes its new demonstration or changes word. This time also include the moment in which the child can give a feedback via the buttons. No significant difference in the writing time was found.

These results show that children when not spending more time per word in one condition of the other. The spatial condition didn't influence the involvement of the child in the task.

Table 2.6 – Probability of feedback given the score per word in the two conditions : face-to-face / *side-by-side*

Feedback Score Event	Positive	Negative	None
Score Increases	0.37 / *0.44*	0.39 / *0.27*	0.24 / *0.29*
Score Decreases	0.28 / *0.47*	0.36 / *0.20*	0.36 / *0.33*

Feedback score of the robot

There was no significant difference of writing score in the F-formation condition tested (side-by-side:$M = 0.80, SD = 0.08, N = 135$, face-to-face: $M = 0.79, SD = 0.08, N = 191$, Anova Repeated measures within subjects:$p > 0.1$). This result means that the children were teaching as well in the face-to-face condition as in the side-by-side condition. However, results showed significant differences in terms of feedback given to the robot regarding the score of the robot.

We analyzed the probabilities of succeeding events considering feedback events and score evolution events.

Table 2.6 shows the frequencies of transition of feedback events (positive or negative) after the score increases or after the score decreases (computed using Markov Chain). We notice that in general the positive feedback frequencies are higher for the side-by-side condition in comparison to the face-to-face. On the contrary, the frequency of negative feedback after is higher in the face-to-face condition. We can also notice that when the score decreases, the frequency of having a positive feedback is almost twice higher in the side-by-side condition.

All the transitions between the scores and the feedback buttons are illustrated on Figure 2.25. The randomness of these results are in contradiction with the correlation between robot's progress and feedback from children observed in section 2.5. This can be explained by the fact interactions were short (4×40 minutes in long-term studies v.s. $2\times \sim 10$ minutes in this study). However, children were displaying a more positive attitude towards the robot when placed in side-by-side position even when the robot was not improving. This positive attitude was showed by rewarding more improvements of the robot and also penalizing less the retrogression of the robot's writing. Children even rewarded retrogression more often in the side-by-side condition. The reward given by children showed to be not often appropriate. For instance in the face-to-face condition, score increasing got more than a third of the time given a negative feedback. These differences in the transition matrix were not significant (Pearson's Chi-squared test, $X - squared = 30, df = 25, p - value = 0.2243$) and a study with a larger number of participants might have given more precise results.

2.8 Need for mutual understanding

The CoWriter activity provides rich interactions – mostly non-verbal – which introduce misunderstandings between the child and the robot:

- **The learning curve of the robot may be not adapted to the child's expectations**: in section 2.5, we observed with table 2.2 a correlation between the robot's progress and the child's evaluation in 5 cases out of 8. Therefore this correlation is not inexorable, especially in short term studies, as seen with figure 2.25 in section 2.7. The robot may too slow at learning a word while the child is providing an high number of refined demonstrations or conversely too fast. This kind of misunderstanding can have important impact on the interaction, since the child may perceive the robot is not learning from his demonstrations, which can be interpreted in two ways: 1) the robot is two bad and the child cannot help or 2) the child is too bad at handwriting to teach the robot.

- **The robot has a poor non-verbal behaviour**: so far, it does not look at what the child wants him to look at. An automatic solution could be to simply imitate the child and to make the robot looking at what the child is looking at, as we did in the experimental setup presented in 3.4. However, there are no reason this is what is expected by the child: the child could ask the robot to look at the tablet while the child is looking at the robot. Using a model of the child and a model of the robot viewed by the child could allow the robot to reason about what it is supposed to look at in order to smooth and enrich the interaction. For example, the robot could be able to look at the rewarding button on the tablet when it expects to be rewarded by the child. Also, the robot does not detect or react when the child is not looking at what he is expected to be looking at (while it is detected by the with-me-ness module presented in 2.6). More than simple visual behaviour, the robot could point at objects it is referring or in order to solve its eventual misunderstanding about an object that the child would be referring. Finally, the recent progress in gesture detection based on deep neural networks allows detecting hands and arms positions [Cao et al., 2018], which could be interpreted in order to feed a mutual-modelling reasoning. For example, we often observed the child providing the robot with a real thumb up as a reward, instead of clicking on the tablet's button. We describe in 4 a methodology to interpret such rewarding signals (with a similar situation described at the begining of 4.4).

- **Sometimes the child does not understand the goal of the activity**. These misunderstanding are the most problematic: the child starts missing the rectangle boxes in the tablets, always choose the same word in the template list or simply teach the robot with a wrong understanding of the correct shape of a letter (in 2.5, we observed Thomas rewarding the robot for imitating him while he was teaching the number 5 horizontally inverted. Then, a facilitator is required to correct the child and an experimenter to restart the robot's behaviour. As a consequence, the robot's role is instantly broken. However, here again, the methodology described in 4 could help understanding the objective of the activity as perceived by the child.

If the robot could detect misunderstanding, it could repair them in order to keep interaction smooth:

- When the child does a mistake (pushing a wrong button on the tablet or writing wrong letters as a demonstration) the robot could detect it and react in consequence.
- Sometimes the child starts to be completely disengaged and the robot should react (by trying to call back the commitment of the child or by asking to stop the activity by itself).
- The robot should wait to have the attention of the child in order to make sure that its trial of writing is being observed.
- The robot should not react as a student, but as a *pretending* student in a didactic activity: if the child provides good feedback but is teaching a very wrong writing to the robot, the robot should be able to detect this situation and to say he does not want to learn this style.

All those situations require a second level of mutual modelling. The next two chapters aim at building a cognitive architecture based on reasoning at two orders of mutual modelling. Such an architecture is expected to be generalizable and usable in different activities. But in order to make sure that this ability brings an improvement to HRI, it needs experimental evaluation involving an interactive robot. This interaction must be studied over long-term sessions in order to facilitate the grounding of non-verbal mutual understandings, and to promote occurrences of misunderstanding situations. We have proven that the CoWriter activity is sustainable by one child over (at least) four sessions of one hour. The study 2 showed that the activity can be used in real therapeutic context and could be an help for therapists: by improving this activity, a mutual modelling architecture could have a direct utility both in education and occupational therapy. The buttons for feedback tested by the study 3 and the evaluation and calibration of the with-me-ness data (study 4) will be a useful feature for mutual modelling. In the clinic-study, we saw that children could give coherent feedbacks to the robot which is a strong information about their perception of a robot as a student while they are the teachers. The VFoA tracker will be used to keep a robust knowledge of what the child has seen and is looking at. This knowledge is essential to reason with 1st and 2nd level of mutual modelling. Furthermore, we believe that the interaction could be improved by adding some micro-behaviours to the robot (short gazes or arm's gestures, non-verbal language).

3 Mutual Understanding for Human-Robot collaborations

3.1 Introduction

Social robots are brought to interact with humans. The quality of such interactions depends on its ability to behave in an acceptable and understandable manner by the user. Hence the importance for a robot to take care of his image: how much it is perceived as an automatic and repetitive agent, or contrarily as a surprising and intelligent character. If the robot is able to detect this perception of itself, it can adapt its behaviour in order to be understood: "you think I am sad while I am happy, I want you to understand that I am happy".

In a collaborative context, where knowledge must be shared, agents must exhibit that they are acquiring the shared information with an immediate behaviour: "I look at what you are showing me, do you see that I am looking at it, do you think I am paying attention to your explanation ?"; "I have understood your idea, do you understand that I have understood ?". As humans, we have different strategies to exhibit understanding or to resolve a misunderstanding. For example, if someone is talking about a visual object, we alternatively gaze between the object and the person to make sure he saw that we gazed at the object. Or if we detect that the other person has not understood a gesture (e.g. pointing at an object) we would probably exaggerate the gesture.

Introduced by Premack and Woodruff [Premack and Woodruff, 1978] and developed by Baron-Cohen and Leslie [Baron-Cohen et al., 1985], ToM describes the ability to attribute mental states and knowledge to others. In interaction, humans are permanently collecting and analyzing huge quantities of information in order to stay aware of emotions, goals and understandings of their fellows.

Until now, the work conduced by the Human-Robot Interaction (HRI) community to develop mutual modelling abilities in robots was limited to a first level of modelling. Higher levels require the ability to recursively attribute a theory of mind to other agents (I think that you think that ...) and their application to HRI remains poorly explored. However, a knowledge of oneself perceived by others is necessary to adapt a behaviour to keep mutual understanding.

An important challenge of social robotics is to provide assistance in education. The ability of robots to support adaptive and repetitive tasks can be valuable in a learning interaction. The CoWriter Project (described in chapter 2) introduces a new approach to help children with difficulties in learning handwriting. Based on the learning by teaching paradigm, the goal of the project is not only to help children with their handwriting, but mainly to improve their self-confidence and motivation in practising such exercise.

The effectiveness of this learning by teaching activity is built on the "protegé effect": the teacher feels responsible for his student, commits to the student's success and possibly experiences student's failure as his own failure to teach. The main idea is to promote the child's extrinsic motivation to write letters (he does it in order to help his "protegé" robot) and to reinforce the self-esteem of the child (he plays the teacher and the robot actually progresses).

In that context, the robot needs to pretend enough difficulties to motivate the child to help it. This ability of the robot to pretend strongly depends on the perception of the robot by the child: the displayed behaviours (gestures, gazes and sounds) by the robot, the initial level and learning speed of the robot must match with what the child imagines of a "robot in difficulty". In order to adapt to the child, the robot needs then to have a model of how it is perceived by the child. On the other side, the child builds also a model of the robot's difficulties and attitude. This mutual-modelling is primordial in order to have mutual understanding and fluid interaction between learner and teacher.

3.2 Related work

3.2.1 Artificial Theory of Mind

The motivation to construct socially-aware agents in computer science is actually much older than the concept of ToM itself. One can argue it takes roots in the sixties from two independent fields: the modal logic with the creation of the Epistemic Modal Logic (EML) [Hintikka, 2005], and the game theory with the apparition of Bayesian games [Harsanyi, 1967].

EML states a list of axioms allowing to reason with knowledge and beliefs of a group of agents, viewed as multiple universes with different modalities. For example, one basic statement says that if an agent a knows that a proposition ψ is true, and knows that ψ implies ϕ, then it knows that ϕ is also true, which is noted "$K_a(\psi) \wedge K_a(\psi \Rightarrow \phi) \Rightarrow K_a(\phi)$". The knowledge level [Newell et al., 1982] embodies EML with agents composed of a set of actions and goals and a body. This improvement formalizes interacting agents, constructing and using their knowledge through decision making. Taking multiple agents interacting in a similar world brought the notion of common knowledge [Halpern and Moses, 1990], making the distinction between facts that are privately known by one agent and facts that are publicly accepted by a group. Having the common knowledge – or a common ground – as a prior, and taking into account the fact that rational agents have goal-oriented behaviors allows probabilistic methods to state beliefs about an observed agent's own knowledge and intention. This is the principle of Beliefs Desires Intentions (BDI) [Rao, 1995] models on which are based a large range of artificial ToM cognitive architectures, and particularly Theory-Theory and Simulation-Theory approaches [Harbers et al., 2009]. The common knowledge also gave birth to the social reasoning [Verbrugge, 2009], that formalizes assertions such as "*Bob knows that Alice knows that both know that ϕ is true*", which is noted "$K_{\text{Bob}} K_{\text{Alice}} C_{\{\text{Bob,Alice}\}} \phi$" (where C denotes the common knowledge). This last formalism is the basis of the doxastic epistemic logic [Van Ditmarsch and Labuschagne, 2007] that aims to construct theory of mind reasoning models, and of the dynamic epistemic logic [Gerbrandy et al., 1997] where an agent's knowledge is no longer static and changes with interaction.

In Bayesian games, players have no access to each other utilities and must establish beliefs about opponents preferences, leading to the existence of Bayesian equilibrium [Kajii and Morris, 1997] which extend the concept of Nash equilibrium [Nash, 1951] to the situation where agents cannot improve their strategies given opponents strategies and their belief about opponents utilities. In fact, the emergency of Nash equilibrium in a group of players involves a common knowledge about each player's utility, rationality and chosen strategy [Aumann et al., 1995]. Bayesian equilibrium alleviates the common knowledge about utilities, but still requires players strategies and rationality assumptions. This point invites to mix EML with game theory which is done by the Recursive Modeling Method (RMM) [Gmytrasiewicz et al., 1991] where each agent is trying to evaluate a belief about opponents utilities by taking into account the fact that the other is also doing the same kind of inference, which induces a recursive computation. RMM is the basis framework of the PsychSim architecture [Pynadath and Marsella, 2005] to implement a Bayesian ToM with a effort for solving sequential decision making process.

A closer approach to our notion of mutual understanding emerges from communication theory, called *grounding* [Clark and Schaefer, 1987]. Grounding is the effort of creating a sufficient common knowledge (the *common ground*) in order to solve a collaborative task [Clark and Brennan, 1991]. If the task requires the ability to predict each other agents, then the mutual understanding is a part of the grounding. When, because of an ambiguity or any reason, the grounding is broken, one must repair it. An interesting property of human dialogues is the ability to repair a broken grounding using minimal efforts [Clark and Wilkes-Gibbs, 1986]: the *Least Collaborative Effort*. Using this idea, it becomes possible to elaborate computer-human collaborative task scenarios where the computer is also able to derive the minimal effort of grounding, resulting with much smoother interactions [Dillenbourg et al., 1996]. In [Cahn and Brennan, 1999], they present a more dialogue-based model describing the process of repairing a common ground that could be used in a human-computer interaction scenario. In this chapter, we develop a similar method, however specialized on the detection and repair the mutual understanding and hence based on prediction error, and that can be applied to any communication way including non-verbal exchanges.

3.2.2 Theory of mind in Human-Robot Interaction

Robot architectures enabling first-order models have been developed within the HRI community, which led to solve basic ToM tests [Breazeal et al., 2009] [Warnier et al., 2012]. More recent architectures extended such reasoning to plan execution for collaborative tasks [Devin and Alami, 2016]. Regarding mutual modeling, second order of ToM has been stepped by Nikolaidis, solving shared plan execution through visual perspective taking: in [Nikolaidis et al., 2016a], the robot is computing the most understandable trajectory in order to share a grabbing intention, rather than the most effective trajectory in terms of time and energy. Our model of reasoning is based on the same idea of playing with the estimated comprehension of the human, but is specialized to context-based story creation while gestural intentions are based on visual and physical computations. Since our activity concerns a sequential decision-making and does not need any visual reasoning, we moved to a simpler ToM approach.

First introduced in Computer-Supported Collaborative Learning (CSCL) [Dillenbourg, 1999] and then borrowed by Human-Robot Interaction (HRI) community [Lemaignan and Dillenbourg, 2015], *mutual modeling* is a computational framework for ToM more focused on collaborative tasks which requires a common knowledge to be solved like, as mentioned above in 3.2.1 finding equilibrium in game theory. Hence, it takes the notation $\mathcal{M}(\text{Alice}, \text{Bob}, X)$ to translate the model made by Alice of what Bob knows about X, where X is a set of facts related to the success of the task. For example, imagine a game where Bob is asked to fill a multiple-choices questionnaire and Alice is asked to guess what Bob is going to answer. Too succeed, Alice needs to predict at least half of the answers given by Bob. In this scenario, X would be the set of questions, and $\mathcal{M}(\text{Alice}, \text{Bob}, X)$ would be the predictions of Bob's answers according to Alice. The accuracy of the model (the number of correct predictions by Alice in our example) is noted $\mathcal{M}^{\circ}(\text{Alice}, \text{Bob}, X)$ and the minimum accuracy of the model to succeed the task (50% in our example) is noted $\mathcal{M}^{\circ}_{\text{min}}(\text{Alice}, \text{Bob}, X)$. This framework states the importance of model symmetry: When two agents, sharing a collaborative task, have different intention given a common observation, it appears that they have a different knowledge regarding the task and must *ground* an agreement in order to solve the misunderstanding and to efficiently collaborate. For example, this is the case if Bob and Alice are in a maze and are attached together with an unbreakable rope: they must agree on the correct pathway to the exit. As in other approaches, higher orders of mutual modeling are defined to express how humans can recursively attribute a model of ToM to others: in the first order agents only construct models of others without supposing that they may also perform mutual modeling, while in the second order they also infer how others model others, including themselves.

We wanted to place our study in the perspective of a pedagogical context, hence we adopted a mutual modeling approach. We focused on Mutual understanding, which involves a second order of modeling: more than simply understanding the other, an agent must take care of being understood. And trying to be understood requires an agent with the capacity to model itself through the eyes of the other.

3.3 Recursive and non-recursive approaches

In this chapter, we make the distinction between agents based on different levels of modelling. Level zero agents do not perform any agent modelling. They may have a model of the world, but this model is uniform and does not make the distinction between the dynamics of the world and the actions of other agents. Level one agents model other agents as level zero agents. Level two agents model other agents as level one agents, and so on. We could also considerate the situation where an agent uses its own architecture to model other agent. But given the fact this agent has the ability to model others included in its architecture, this situation would induce an infinite regression [Clark, 1988]: agent A models agent B that models agent A that models agent B *etc.* Despite the fact humans are able to reach high levels of modelling, we will focus on the construction of a second level agent, hence modelling the eventual human to interact with as a first level agent. Doing so alleviates a lot the reasoning and is sufficient to induce rich social behaviours (as shown in chapter 4).

3.3.1 Framework

Let "A" be a first-order agent and "A, B" (the agent B perceived by A) a second-order agent. Then, $M_R\,[A]$ stands for *the model (built by the robot R) about the agent A* (first level of modelling) and $M_R\,[A{,}B]$ stands for *the model (built by the robot R) about the agent B perceived by the agent A* (second level of modelling). This is better explained by the following equations:

$$\begin{aligned} M_A\,[B] &= A{,}B \\ M_R\,[M_A[B]] &= M_R\,[A{,}B]\,. \end{aligned}$$

We use this notation since we only implement the models used by the robot (starting by M_R), while all other "models" used by humans are considered as independent agents. Therefore, an agent perceived by another agent defines a new agent, not a model. Hence the distinction between "A, B", that represents an agent perceived by another one, and $M_A\,[B]$, that is the model of the agent B built by A within the architecture described above. The model of the robot perceived by the human $M_R\,[H{,}R]$ is not a part of the model of the human $M_R\,[H]$ but is encoded as a model of another agent (see figure 3.4).

As said above, we limit our approach to 1st and 2nd order of modelling. In a two-agents interaction (the human and the robot) we will focus on three models: $M_R\,[C]$ (the model about the human), $M_R\,[R]$ (the model about the robot) and $M_R\,[C{,}R]$ (the robot perceived by the human). It would be also interesting to study $M_R\,[C{,}C]$, the model about the human perceived by himself in order to play with his self-confidence. But detecting differences between $M_R\,[C]$ and $M_R\,[C{,}C]$ seems difficult with the current abilities of the robot. Since models are dynamic, $M_R^t\,[A]$ represent the model about an agent A at

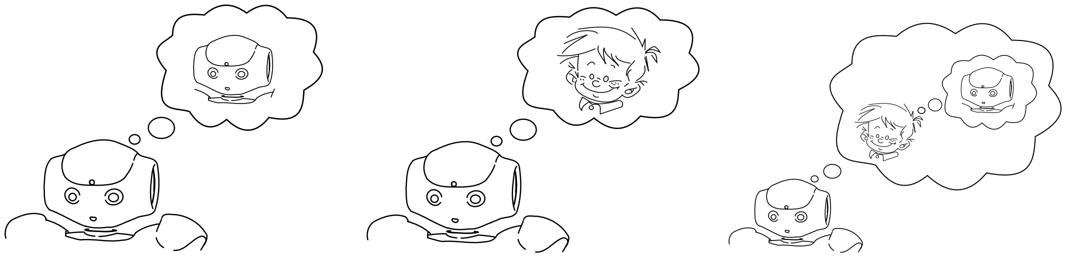


Figure 3.1 – $M_R\,[R]$ Figure 3.2 – $M_R\,[H]$ Figure 3.3 – $M_R\,[H{,}R]$

Figure 3.4 – 3-agents approach for second order modeling. Left: robot's model of itself. Middle: robot's model of the human. Right: robot's model of itself as perceived by the human.

time t.

3.3.2 Mutual understanding

Given these three models ($M_R\,[R]$, $M_R\,[C]$ and $M_R\,[C{,}R]$) the robot must be able to detect misunderstandings. A misunderstanding of an agent A by the robot can be formalized as a error between what is actually in the mind of the agent (we can call it $\Phi[A]$) and the model built by the robot: $\Delta\,(\Phi[A]; M_R\,[A])$. But if A is human, $\Phi[A]$ is inaccessible to the robot. In order to maintain a mutual understanding, humans [Suzuki et al., 2015] (and monkeys [Haroush and Williams, 2015]), use predictions of others' behaviours. A bio-inspired approach would be to make, at time t, a prediction $P_R^{t+1}\,[A]$ of the model. Then, at time $t+1$, the robot can compute a **prediction error** $\Delta\left(M_R^{t+1}\,[A]\,; P_R^{t+1}\,[A]\right)$ in order to detect such a misunderstanding. This idea rely on the assumption that the better are the predictions of a model, the better the model fits the reality. Then, the dynamic of the model can be updated according to the resulting error of prediction. This rule can be used with $M_R\,[H]$ and $M_R\,[H{,}R]$.

Another type of misunderstanding concerns the comprehension of the robot by the human: using the same formalism, it is an error between the actual perception of the robot by the human (we can call it $\Phi[H,R]$) and the robot itself: $\Delta\,(\Phi[H,R]; M_R\,[R])$. Again, the robot does not have access to $\Phi[H,R]$, but it approximates it with $M_R\,[H{,}R]$. Finally we define the **human's perception error** at time t by $\Delta\,(M_R^t\,[H{,}R]\,; M_R^t\,[R])$. This error is taken in account only if the robot has a correct model of itself perceived by the human (only if $M_R\,[H{,}R]$ produce small prediction errors). Since this error assumes that models built by the robot are correct, it is not used to update these models. It corresponds to an error of the human: in order to repair it, the robot must explain the misunderstanding to the human or exaggerate an action [Nikolaidis et al., 2016b] to make sure it will be understood.

As an example, in the CoWriter activity, the human teach handwriting to the robot. The robot pretends to be a beginner, but it has in fact its idea of a good handwriting. It is perfectly aware of its played progresses. We want the human to perceive these progresses. In that perspective, the human's perception error corresponds to the sentence "*I make progress but the human does not perceive it, he does not understand my progress*", while the prediction error of $M_R\,[H,R]$ corresponds to "*I do not understand how the human perceive my progress*". Tables 3.1 and 3.2 show crude examples of situation involving prediction or human's perception errors and possible reparation.

Model	**Utterance**
$M_R^t\,[H]$	*human looks at me and do nothing*
$P_R^{t+1}\,[H]$	*human will say something*
$M_R^{t+1}\,[H]$	*human still looks at me and do nothing*
$\Delta \geq \Theta$	*I am misunderstanding the human*
action to repair	tell the human "*Are you OK ?*"

Table 3.1 – Prediction error with the model of the human

Model	**Utterance**
$M_R^t\,[R]$	*I know I am not making progress*
$M_R^t\,[H]$	provides the robot with positive feedback
$M_R^t\,[H,R]$	*human thinks I make progress*
$\Delta \geq \Theta$	*The human is misunderstanding that I am not doing any progress*
action to repair	write with a style even worse than before

Table 3.2 – human's perception error

3.4 Impacts on Human-Robot interactions

In this section, we implemented a reasoning model for mutual understanding based on a three-agents architecture: *self; other; self-view-by-other*, introduced in [Jacq et al., 2016a]. We used it to implement two robot's behaviors: making predictable decisions or making adversarial decisions. These behaviors are designed within an activity where the robot chooses, turn by turn with a human, elements that construct a short story. Our predictable behavior is built in order to facilitate the mutual understanding, while our adversarial behavior lets the subject believe he understands the robot and suddenly surprises him with the least predictable decision. Actually, the adversarial behaviour breaks the mutual understanding: we want to create misunderstanding situation in order to study the importance of maintaining the mutual understanding. As a control condition, we also implemented a random behavior, in which the robot only makes random decisions. We conducted a study involving 47 subjects, not aware of the robot's behavior condition.

3.4.1 Story co-creation by selecting elements

The activity consist in choosing, turn by turn with the robot, a specific element of the story. Such an element can be the place of the story (planet? kingdom? island?) or the job of the protagonist (space pioneer? knight? pirate?). Once all elements have been selected by the subject and the robot, the resulting story is generated, based on the human-robot collaborative selection of contents. Actually, the story is rather "filled" than generated: at the beginning, a sentence has a fixed structure but each word that is – or depends on – a selectable element is replaced by a symbolic variable. For example, our story could start with the two following sentences:

Once upon a time, in a **Place** *far away populated by* **People***, was living a wild* **Main_Char_Job** *named* **Main_Char_Name**.
Personal_Pronoun(Main_Char_Gender) *was very brave.*

In this text, variables are the bold terms. The variable "Place" is a selectable element, that can be replaced by any possible geographical place (planet, kingdom, island, ...). The personal pronoun related to the main character depends on the selectable element "Main_Char_Gender". Some whole sentences can also depend on a variable in order to avoid inconsistencies.

In order to choose an element, a subject must touch it on a touchable screen. For its part, the robot just vaguely points it with its finger and the element is in parallel selected on the screen. The robot is also provided with a face detector and alternates head movements, gazing at the screen or at the subject. Finally, when the robot performs hand gestures while speaking. Over all, the subject is asked to choose 10 elements and the robot is asked to choose 8 elements (the subject chooses the first one and the last one).

Before each robot's turn, subjects are asked to predict what will be the robot's decision. The sequence of successive triples (*subject's decision*; *subject's prediction of the robot*; *robot's decision*) was feeding our two decision making algorithms based on 2nd order ToM. Doing so forces the modelling of the robot by the subject, therefore we measure a biased behaviour. However the way the subject models the robot still depends on the robot's decision which are fully conditioned by our compared group conditions and we are interested by the impact these conditions can possibly have in the worst possible scenario.

3.4.2 Decision making

Contexts

We define a context as a set of selectable elements belonging to a same semantic field. For example, the context *science fiction* contains the elements *planet*, *alien*, *lazer gun*, etc. We arbitrary set 8 contexts: *science fiction*, *pirates*, *middle-ages*, *forest*, *science*, *army*, *robots*, *magic*. Since an element can be associated to several contexts, contexts are not disjoint.

Agent models

As suggested in [Jacq et al., 2016a], we define three agents: the robot ($\mathcal{R}$), the human ($\mathcal{H}$), the robot predicted by the human ($\mathcal{H}, \mathcal{R}$). Each agent $\mathcal{A}$ is modeled by a log-probability distribution over contexts, $\mathfrak{L}_{\mathcal{A}}$, estimating the odds that it is going to pick elements from this context. Taking the notation from 3.3, we have:

$$\forall \ agent \ \mathcal{A}, \quad M_{\mathcal{R}}[\mathcal{A}] = \mathfrak{L}_{\mathcal{A}}.$$

For example, $\mathfrak{L}_{\mathcal{H}}(pirates)$ estimates the probability of the event “the human is going to pick an element in the *pirates* context", while $\mathfrak{L}_{\mathcal{H},\mathcal{R}}(pirates)$ estimates the probability of the event “the human predicts that the robot is going to pick an element in the *pirates* context". From these distributions, we can define, for each agent $\mathcal{A}$, its most likely context $\mathtt{C}_{\mathcal{A}}^{max} = \text{argmax}_{\mathtt{C}} \mathfrak{L}_{\mathcal{A}}(\mathtt{C})$ and its least likely context $\mathtt{C}_{\mathcal{A}}^{min} = \text{argmin}_{\mathtt{C}} \mathfrak{L}_{\mathcal{A}}(\mathtt{C})$.

Agent weights

Each agent $\mathcal{A}$ is given a weight $W_{\mathcal{A}}$ representing the human inclination to establish its predictions, rather based on the robot's decisions ($W_{\mathcal{R}}$), on his own decisions ($W_{\mathcal{H}}$) or on his own predictions of the robot ($W_{\mathcal{H},\mathcal{R}}$).

Weights updates

At each step of the element-selection activity, we receive a new triple ($\mathtt{e}_{\mathcal{H}}$; $\mathtt{e}_{\mathcal{H},\mathcal{R}}$; $\mathtt{e}_{\mathcal{R}}$) where $\mathtt{e}_{\mathcal{H}}$ is the element picked by the human, $\mathtt{e}_{\mathcal{H},\mathcal{R}}$ is the human prediction of the element picked by the robot, and $\mathtt{e}_{\mathcal{R}}$ is the element actually picked by the robot. An agent's weight $W_{\mathcal{A}}$ is incremented if its last picked element $\mathtt{e}_{\mathcal{A}}$ belongs to its most likely context $\mathtt{C}_{\mathcal{A}}^{max}$:

$$W_{\mathcal{A}} \leftarrow W_{\mathcal{A}} + \mathbb{1}\{\mathtt{e}_{\mathcal{A}} \in \mathtt{C}_{\mathcal{A}}^{max}\} \ \forall \ agent \ \mathcal{A}$$

Probabilities updates

Then, agents log-probability distributions $\mathfrak{L}_{\mathcal{H}}$ and $\mathfrak{L}_{\mathcal{R}}$ are both updated in a similar way, for all context $\mathtt{C}$:

$$\mathfrak{L}_{\mathcal{H}}(\mathtt{C}) \leftarrow \mathfrak{L}_{\mathcal{H}}(\mathtt{C}) + \mathbb{1}\{\mathtt{e}_{\mathcal{H}} \in \mathtt{C}\}$$
$$\mathfrak{L}_{\mathcal{R}}(\mathtt{C}) \leftarrow \mathfrak{L}_{\mathcal{R}}(\mathtt{C}) + \mathbb{1}\{\mathtt{e}_{\mathcal{R}} \in \mathtt{C}\}$$

While $\mathfrak{L}_{\mathcal{H},\mathcal{R}}$ is updated using weights $W_{\mathcal{R}}$, $W_{\mathcal{H}}$ and $W_{\mathcal{H},\mathcal{R}}$, for all context $\mathtt{C}$:

$$\mathfrak{L}_{\mathcal{H},\mathcal{R}}(\mathtt{C}) \leftarrow \mathfrak{L}_{\mathcal{H},\mathcal{R}}(\mathtt{C}) + \sum_{\mathcal{A} \in \{\mathcal{R},\mathcal{H},\mathcal{H},\mathcal{R}\}} W_{\mathcal{A}} * \mathbb{1}\{\mathtt{e}_{\mathcal{A}} \in \mathtt{C}\}$$

Predictable behavior

Our predictable behavior aims at making decisions that are easily predicted by the subject. In that purpose, the robot always pick elements from $\mathcal{H}, \mathcal{R}$'s most likely context $\mathtt{C}^{max}_{\mathcal{H},\mathcal{R}}$:

$$\mathtt{e}_{\mathcal{R}} \in \mathtt{C}^{max}_{\mathcal{H},\mathcal{R}}$$

Adversarial behavior

The adversarial behavior is more complex. We use the predictable behavior, waiting for the human to make good predictions (predicting an element $\mathtt{e}_{\mathcal{H},\mathcal{R}}$ belonging to $\mathtt{C}^{max}_{\mathcal{H},\mathcal{R}}$). Then, we suddenly move to the opposite: picking $\mathtt{e}_{\mathcal{R}}$ in the least likely context $\mathtt{C}^{min}_{\mathcal{H},\mathcal{R}}$. However, we wanted to make this behavior the least understandable. Therefore we add, with a low probability, the possibility to pick $\mathtt{e}_{\mathcal{R}}$ from $\mathtt{C}^{max}_{\mathcal{H},\mathcal{R}}$ while the human is making a good prediction, or the possibility to pick exactly the element predicted by the subject while the human did not predict an element from $\mathtt{C}^{max}_{\mathcal{H},\mathcal{R}}$. We arbitrary fixed the low probabilities to $P = 0.2$, since we wanted to observe an average of more than one and less than two such unlikely event over the 8 robot's decisions in the activity. Algorithm 3 summarizes this behavior.

Algorithm 3: adversarial behavior

```
if $\mathtt{e}_{\mathcal{H},\mathcal{R}} \in \mathtt{C}^{max}_{\mathcal{H},\mathcal{R}}$ then
    with prob. P=0.8, $\mathtt{e}_{\mathcal{R}} \in \mathtt{C}^{min}_{\mathcal{H},\mathcal{R}}$
    with prob. P=0.2, $\mathtt{e}_{\mathcal{R}} \in \mathtt{C}^{max}_{\mathcal{H},\mathcal{R}}$
else
    with prob. P=0.8, $\mathtt{e}_{\mathcal{R}} \in \mathtt{C}^{max}_{\mathcal{H},\mathcal{R}}$
    with prob. P=0.2, $\mathtt{e}_{\mathcal{R}} = \mathtt{e}_{\mathcal{H},\mathcal{R}}$
```

3.4.3 Experiment

We conducted an experiment in order to study the impact of the three behaviors of the robot on the interaction. The content of the activity was designed in English language. In order to make sure they had a good understanding of English, we invited undergrad students to be subjects for our experiment. However, this decision may have brought weaknesses regarding our possible results. First, this population is biased by the fact that a part of them have already been implied in a human-robot experiment. Then, this story co-creation activity aims to provide a support for children education, and results in adults population may never be generalized to children.

Groups

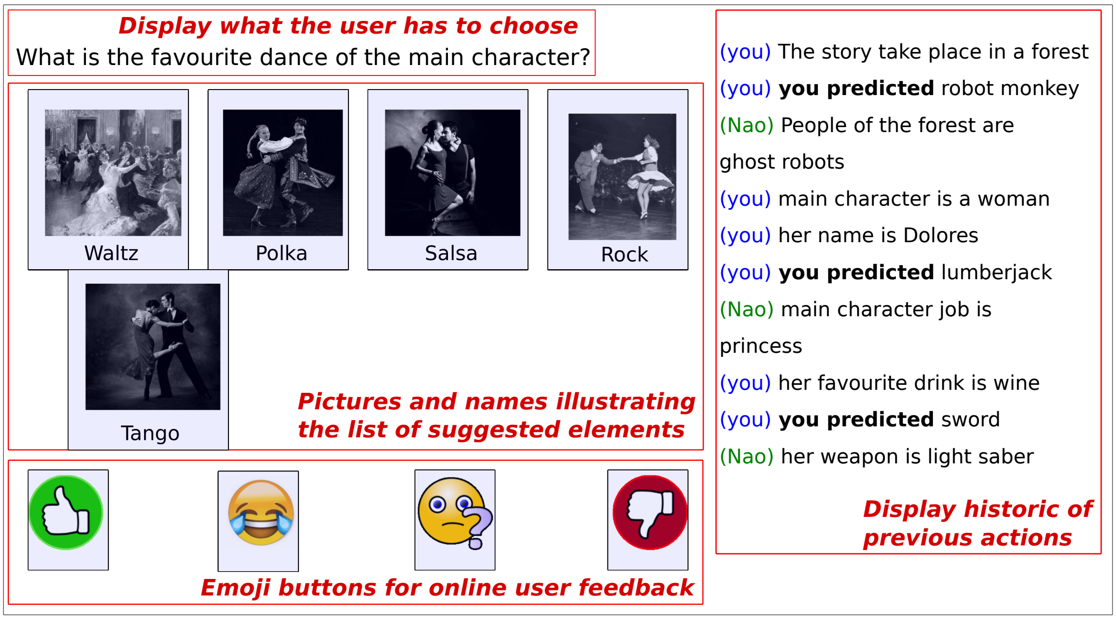
A total of 47 students (18f, 29m) accepted to participate to the study. The experiment was conducted in our laboratory. Subjects were aged between 18 and 34 (M 22.8, SD 3.9). We defined 3 groups in which subjects were randomly allocated: the random-behavior group (9f, 7m), the predictable-behavior group (5f, 11m) and the surprise-behavior group (4f, 11m). We used the random behavior as a control condition.

Settings

Each subject was alone with the robot in the room during the whole interaction and the robot was fully autonomous. The spatial arrangement is detailed in figure 3.6 (top view) and 3.7 (camera view). The robot, standing on a support, is facing the human user and between them, a touchable screen is inclined for the subject. Also on the support, at the feet of the robot, a RGB-camera was tracking the user's face. We used face-tracking for attention estimation (see 3.4.3), but also in order to implement robot's head movements. The questionnaire was displayed on the touchable screen and required to scroll down with a mouse. For that purpose, subjects had a mouse available on the right of the screen. The experiment was designed in 4 phases:

1) Introduction (0.75 min exactly): At the beginning, the screen is empty. The robot introduces itself and the activity. All the speeches of the robot were scripted and can be found in our source code, available on Github.

2)Turn by turn selection of elements (4.7 min on average): To start, the robot asks subjects to choose the first element: the place of the story (planet, forest, kingdom...). The interface appears on the screen, displaying a suggestion of possible elements the subject can choose. Figure 3.5 shows an example of screen capture of the interface for subject's turn. Then, elements that will be suggested to the robot are shown on the screen and subjects are asked to guess what the robot is going to choose. When a subject has made his prediction, the robot takes its turn and chooses, by pointing a button with



Display what the user has to choose

What is the favourite dance of the main character?

Pictures and names illustrating the list of suggested elements

Emoji buttons for online user feedback

(you) The story take place in a forest
(you) **you predicted** robot monkey
(Nao) People of the forest are ghost robots
(you) main character is a woman
(you) her name is Dolores
(you) **you predicted** lumberjack
(Nao) main character job is princess
(you) her favourite drink is wine
(you) **you predicted** sword
(Nao) her weapon is light saber

Display historic of previous actions

Figure 3.5 – Screen capture of user interface. It contains 4 areas. Top-left: a question reminds what kind of element the user has to choose (for instance, the favorite dance of the main character). Center: the set of suggested elements the user can choose illustrated by pictures. Bottom: 4 emoji buttons the user can use, if he wants to, in order to share his feeling. Right: a column displays the historic of previous action in order to help the user to make prediction about robot's actions.

its arm, the next element. During this turn, buttons to pick elements do not react to subjects' touch. Finally it is the subject's turn again, etc. In order to better feed user modeling algorithms, at two points the human had two consecutive turns, hence the human made more decisions than the robot (10 turns for the human and 8 turns for the robot).

3) Story-telling (3.6 min exactly): At the end, when all elements have been selected by the human and the robot, the resulting story is generated, and the robot tells the story to the human. While the robot tells the story, the screen displays the told sentences. At any time during the whole interaction (including both co-creation and storytelling phases) four emoji buttons were displayed on the screen and could be used by subjects whenever they wanted to share feedback about their feelings. As in [Jacq et al., 2016b] and [Johal et al., 2016] we used thumbs up and down, plus two emoji buttons for "laugh" or "absurd" feeling.

4) Questionnaire (10.3 min on average) Finally, a questionnaire appeared on the screen, asking subject about their appreciation of the activity, their perception of the robot (Godspeed) and their perception of it's ToM abilities.

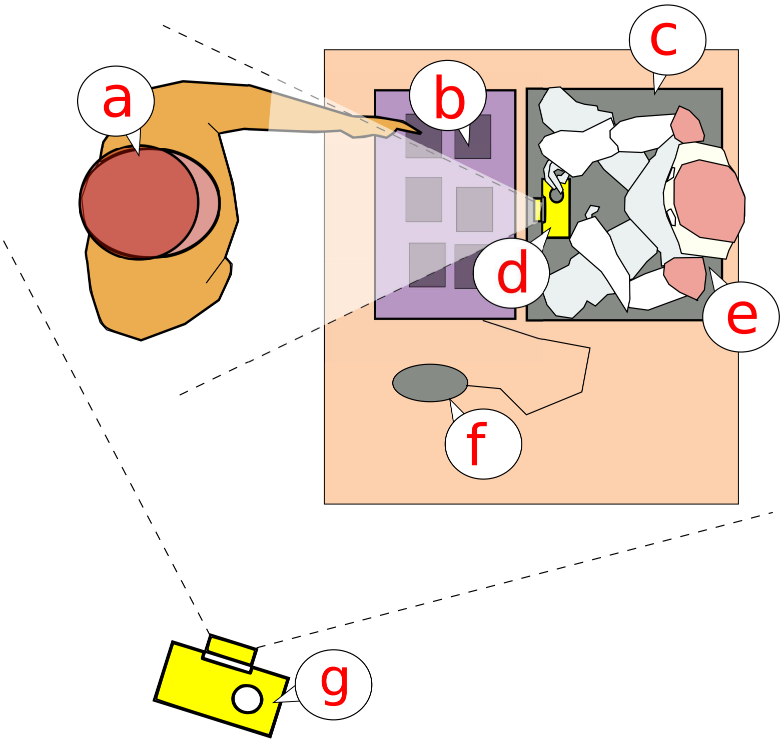


Figure 3.6 – Spacial arrangement, top view: (a) subject, (b) touchable screen, (c) support for the robot, (d) rgb-camera for face-tracking, (e) robot, (f) mouse helping the subject to fill the questionnaire, (g) camera filming the interaction.

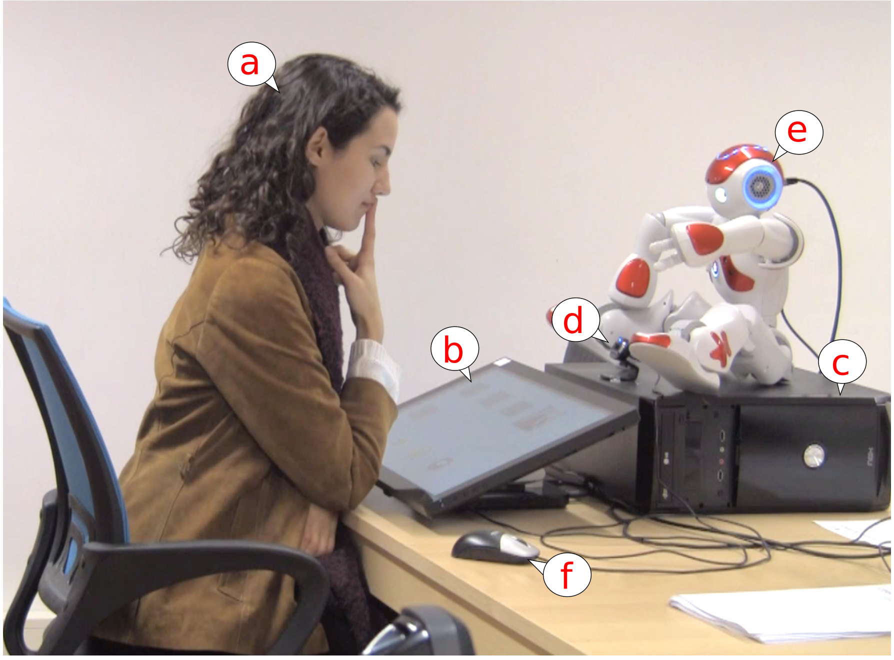


Figure 3.7 – Spatial arrangement, camera view: (a) subject, (b) touchable screen, (c) support for the robot, (d) rgb-camera for face-tracking, (e) robot, (f) mouse helping the subject to fill the questionnaire.

Procedure

Before the experiment, subjects were asked to sign a consent form. Then, a researcher responsible explained the activity, telling that they would have to choose, turn by turn with the robot, the elements of a story. He also explained to them that the robot would tell the resulting story after all elements being picked. He told subjects that they would have to make a prediction of the robot's decision before each robot's turn. We used a screen-shot showing a typical example of the interface and the researcher explained that they could use emoji buttons to provide feedback whenever he wanted, during the whole interaction including the robot's story telling. Finally, the researcher explained to subjects they would have to fill a questionnaire that would be displayed on the screen after the interaction. He indicated the mouse they would have to use in order to fill the questionnaire. When interactions started, the robot introduced itself and reminded subjects with a quick description of the activity. At each prediction step, it asked subjects to make a prediction of what it was going to choose.

Measures

In order to measure our models accuracy, we counted the number of time the human was picking or predicting elements in the expected most likely contexts: the number of time that $\mathtt{e}_{\mathcal{H}} \in \mathtt{C}_{\mathcal{H}}^{max}$ and the number of time that $\mathtt{e}_{\mathcal{H},\mathcal{R}} \in \mathtt{C}_{\mathcal{H},\mathcal{R}}^{max}$. We estimated a degree of mutual understanding based on the number of times the human successfully predicted the robot. Emoji buttons were used to estimate on-line subjects appreciation of robot's decisions. In order to track the gaze direction of subjects, we used a system similar to Attention-tracker [Lemaignan et al., 2016a], improved with OpenFace Library [Amos et al., 2016]. This system is available on GITHUB. As in [Lemaignan et al., 2016a], we measured an on-line estimation of *with-me-ness*. In our setup, *with-me-ness* was defined by the frequency a subject looks at the screen or at the head of the robot, over an exponential moving average:

$$wmn^t = 0.9 * wmn^{t-1} + 0.1 * \mathbb{1}_{targets}^t$$

In the above equation, wmn^t represents our estimated *with-me-ness* at time t. $\mathbb{1}_{targets}^t$ equals 1 if the subject is looking at the screen or the head of the robot at time t, otherwise it equals 0. This is a simplification of the original definition of *with-me-ness* where targets (robot's head and screen) are the same in all phases of the interaction.

The questionnaire was designed in three parts. The first part contained five questions regarding the appreciation of the subject: three about the resulting story (bad – good, not funny – funny, coherent – absurd) and two about feeling during the co-creation (negative – positive, bored – excited). The second part was a randomly shuffled Godspeed questionnaire [Bartneck et al., 2009]. The last part contained four questions concerning the perception of mutual understanding:

- *Do you think the robot took into account your choices?*
- *Do you think the robot took into account your predictions?*
- *Do you think the robot was predicting your choices?*
- *Were you able to predict the robot choices?*

In all parts, subjects had to pick a number over a type-Likert scale between 1 and 6, in order to avoid middle points and to force them to settle between the two opposite answers.

3.4.4 Results

Model accuracy

We compared the observed accuracy (frequency that $\mathtt{e}_{\mathcal{H}} \in \mathtt{C}_{\mathcal{H}}^{max}$ and that $\mathtt{e}_{\mathcal{H},\mathcal{R}} \in \mathtt{C}_{\mathcal{H},\mathcal{R}}^{max}$) with uniform distribution over the suggested set of element at each activity's step (figure 3.8). We observed frequencies significantly higher than random odds for rich context-depending steps (protagonist's name, favorite drink, job and weapon, 2nd character's type). Focusing on figure 3.8B, we could only predict subject's predictions better than randomly at the beginning of the interaction after which, in both random and adversarial conditions, it became too difficult for subjects to infer robot's intentions.

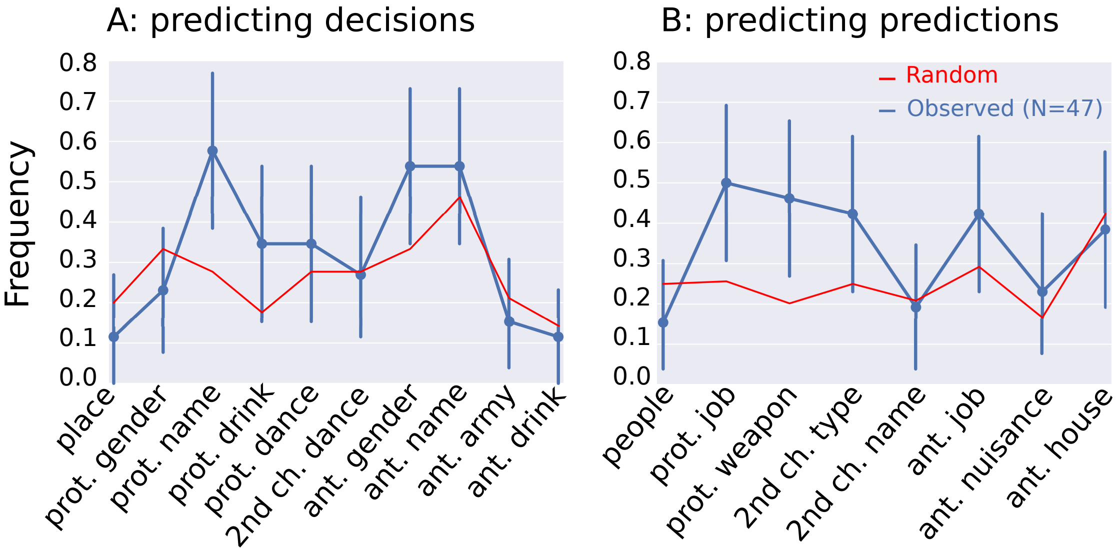


Figure 3.8 – Model accuracy vs random probability. A: (blue) frequency that $\mathtt{e}_{\mathcal{H}} \in \mathtt{C}_{\mathcal{H}}^{max}$. B: (blue) frequency that $\mathtt{e}_{\mathcal{H},\mathcal{R}} \in \mathtt{C}_{\mathcal{H},\mathcal{R}}^{max}$. (red) probability of picking the most likely context from a random decision.

Actual vs perceived mutual understanding

As expected, choices of the robot in the predictable condition were more susceptible to be predicted by subjects. The number of successful predictions was higher in predictable condition than in adversarial and random conditions. We obtained similar results with the average intensity of answers (1=Not at all, 6=totally) to the question *“Were you able to predict the robot choices?"*, meaning subjects were aware of the difficulty to predict the robot in the adversarial and random conditions. However, to the questions *“Do you think the robot took into account your choices"* and *“Do you think the robot was predicting your choices"*, subjects gave higher scores in the predictable condition than in the adversarial condition, but no differences between predictable and control conditions were found. The robot took into account subjects predictions only in predictable and adversarial conditions. But when we asked subjects to answer the question *“Do you think the robot took into account your predictions"*, we found that answers intensity was significantly lower in the adversarial condition than in both predictable and random conditions. Observations and statistics are displayed by figure 3.9 and 3.10.

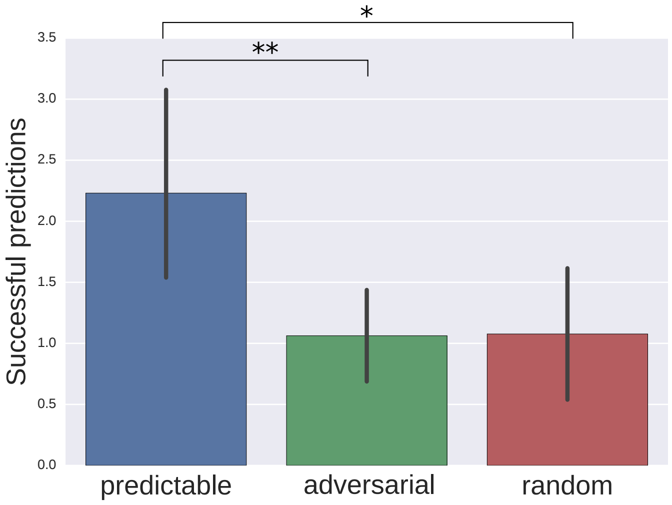


Figure 3.9 – Measured mutual understanding. We used average number of successful predictions of robot choices by subjects. T-test p-values: $(*) < .05, (**) < .01$

Appreciation

The First part of our questionnaire concerned the appreciation of the activity and the created story rather than the robot. However, answers of subjects were similar in the three conditions (*was the story good?*: M=5+/-0.12, *funny?*: M=5+/-0.2, *absurd?*: M=4+/-0.1, did you felt positive?: M=5+-0.1, excited?: 4.7+/-0.1). We used emoji buttons in order to capture on-line judgment of the robot by subjects. Unfortunately, the usage of these buttons (9.7 presses/subject) was too rare to obtain small enough standard deviations required for significant results. Despite this fact, we observed more presses in the adversarial condition (M=11.2, SD=7.9) than in predictable (M=8.15, SD=6.8) and random (M=9.38, SD=8.1) conditions. This higher usage of button in the adversarial condition is observed in all buttons separately, except for the “absurd” emoji button

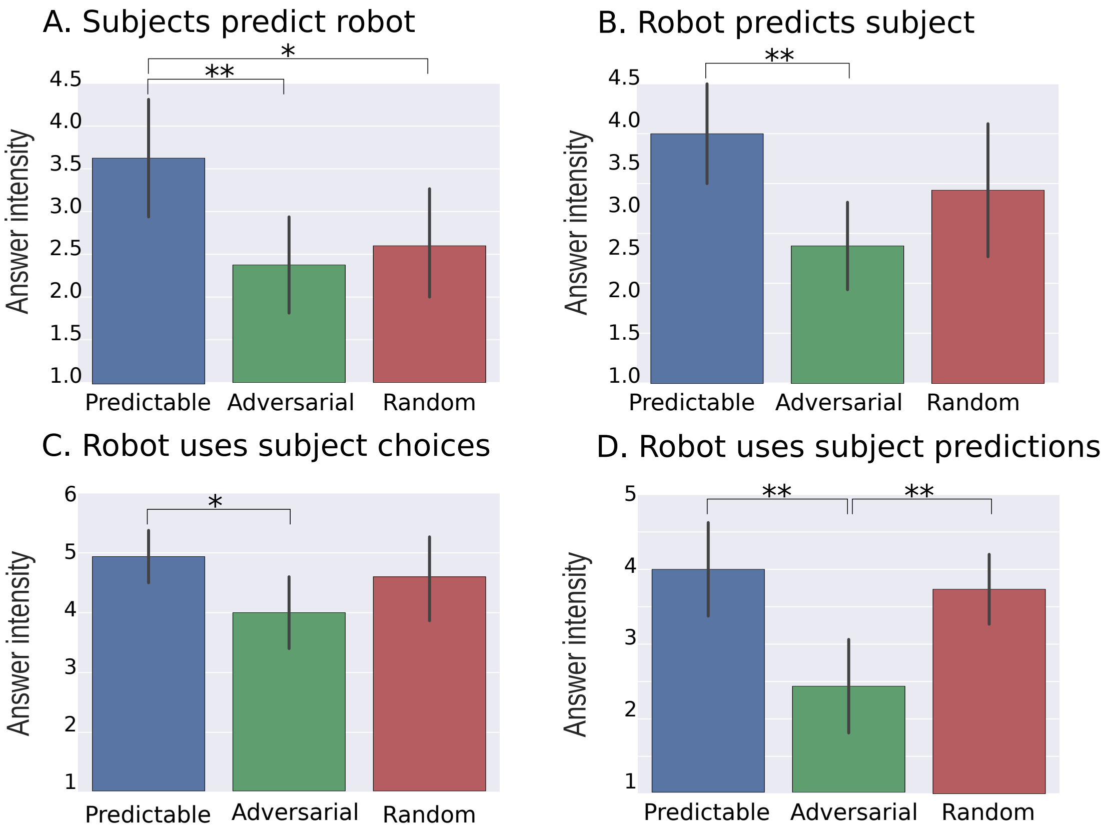


Figure 3.10 – Perceived mutual understanding. Average intensity of answers to the question (A): *Were you able to predict the robot choices?* (B): *Do you think the robot was predicting your choices?* (C): *Do you think the robot took into account your choices?* (D): *Do you think the robot took into account your predictions?*. For all 4 questions: 1 = *not at all*, 6 = *totally*.

that was more used in the random condition. These results are displayed by Figure 3.11. The Godspeed part of the questionnaire contains questions asking for a judgment of the robot. The difference with emoji buttons was the fact these judgments were not direct responses to particular choices of the robot, but rather global feelings about its aspect and behavior remaining after the interaction. These questions can be sorted into 4 groups: anthropomorphism, animacy, intelligence, and likability. We concatenated answers to questions belonging to the same group. We observed lower appreciations in the adversarial condition compared to predictable and random conditions in all other groups of questions. For anthropomorphism, answers from the adversarial condition were significantly lower than from predictable and random. A similar observation concerning animacy, with answers from the adversarial condition being lower than from predictable and random. For perceived intelligence, answers from the adversarial condition were lower than from random condition. The highest gap concerned answers to likability questions: answers from the adversarial condition were significantly lower than from predictable and random conditions. Interestingly, we also found a significant preference for the random condition compared to predictable condition. Godspeed measures are

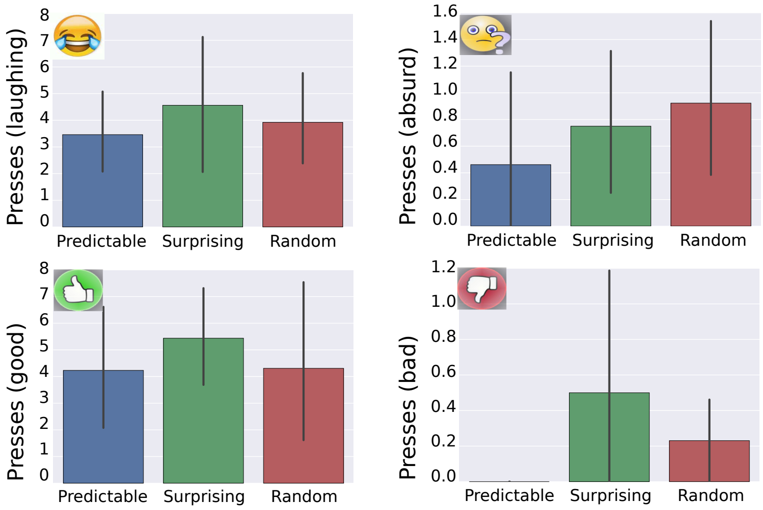


Figure 3.11 – Average number of presses per interaction in different buttons. One can notice the total absence of usage of the “thumb-down” in the predictable condition. However, we observed no significant difference toward conditions.

displayed by Figure 3.12.

Attention

We obtained a set of time series representing evolution of *with-me-ness* for each condition. While measures in predictable and random conditions where correlated (Pearson‘s correlation between average curves: $r(540) = 0.75, p < .001$), the set of curves obtained in the adversarial condition deviated in average to stay at a lower level of measured *with-me-ness*. We estimated the attention of subjects through the evolution of measured *with-me-ness* over time. That way, we obtained a set of time series for each condition. Figure 3.13 (top) displays the average curve for each condition. Given the correlation between with-me-ness measures in predictable and random conditions (Pearson‘s correlation between average curves: $r(540) = 0.75, p < .001$), we assumed these two sets of trajectories were following the same stochastic laws. Then, we focused on the deviations observed in the adversarial condition, visually lower than in the two other conditions. In order to study how significantly these curves deviated from other sets, we regrouped predictable and random conditions into one set of curves opposed to the adversarial condition. We did not take into account the introduction phase in which the robot’s behavior was the same for each condition. We used a Student’s t-test on a moving window of 20 seconds (assuming same standard deviations in both sets). In each window’s step, our null hypothesis was based on the adversarial condition average with-me-ness being the same as the averages in the other two conditions. We reported in figure 3.13 (bottom) the evolution of the obtained probability of observed windows means given the null hypothesis (p-values). We highlighted the three phases of around 50s where we found

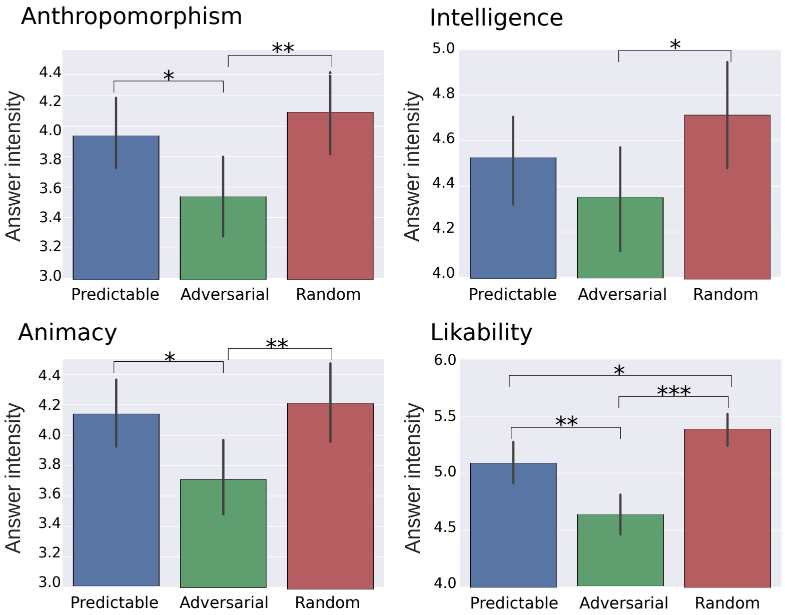


Figure 3.12 – Answers to Godspeed questionnaire. T-test p-values: $(*) < .05$, $(**) < .01$, $(***) < .001$

curves from the adversarial condition were significantly lower. These phases correspond to the construction of the protagonist and antagonist in the story construction, and to the beginning of the story telling phase (during which the measured "with-me-ness" globally decreased in all conditions).

3.5 Discussion

Regarding mutual modeling results, it seems that subjects were aware of their ability to predict the robot, but other questions of the last part of the questionnaire show how they perceived the adversarial condition as a lack of understanding in the robot. As expected, the adversarial condition generated a perception of the ToM reasoning of the robot significantly lower than in the predictable condition, but even lower than control condition concerning the impact of subjects predictions. Beside, it seems that the decision mechanism of the robot in the random condition was overestimated, being not differentiable from the predictable condition. We can associate these different perceptions of robot's decision making with tracked attention results, in which trajectories from predictable and random condition were similar while trajectories from adversarial condition were significantly lower during three phases of approximately 50s. We can also explain Godspeed results in which concerns robot's anthropomorphism, intelligence and animacy, for which, while no difference was observed between predictable and control conditions, robot's qualities were perceived significantly lower in the adversarial condition than in control and, except for intelligence, significantly lower than in predictable

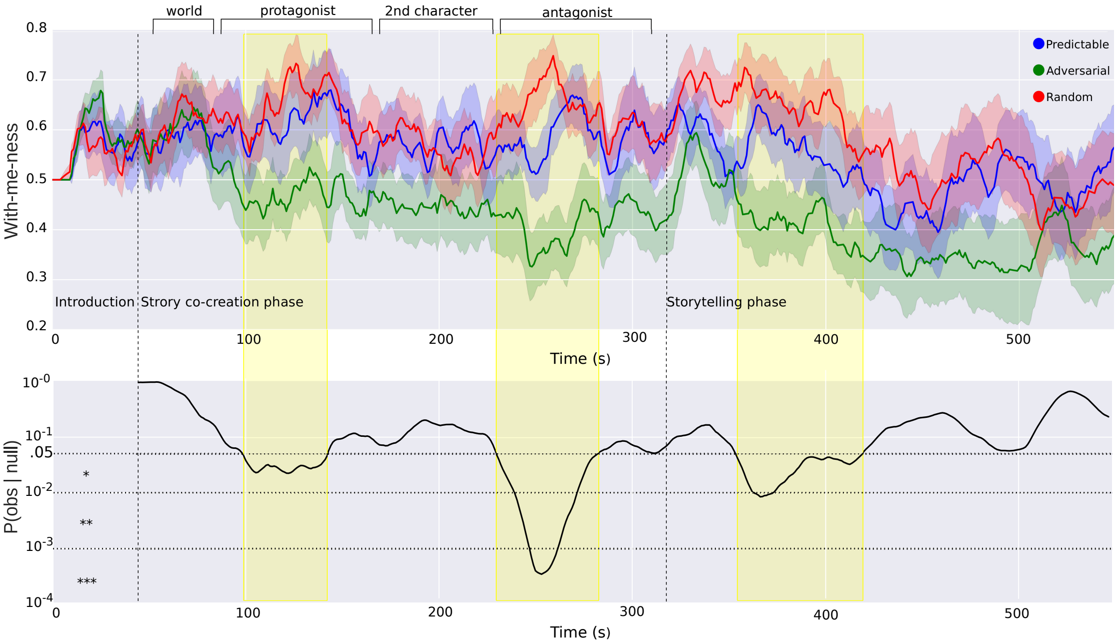


Figure 3.13 – Measured “with-me-ness” over time. Top: average curve for each condition, plus or least standard deviations. Pearson‘s correlation between average curves for predictable and random condition: $r(540) = 0.75, p < .001$. Bottom: Student’s t-test on a moving window of 20 seconds (assuming same standard deviations in both sets). In each window’s step, our null hypothesis was based on the adversarial condition’s mean being the same than the “natural" trajectories’ mean. The black curve displays the evolution of the obtained probability of observed windows means given the null hypothesis (p-values).

condition.

However, an unexpected observation concerned answers to the Godspeed likability questions, according to which the robot was even more appreciated in the random condition than in the predictable condition. A possible interpretation could be that the random condition was least boring than the predictable condition. We could even suggest that in predictable and adversarial condition, subjects started to create a coherent story while in the random condition, they were directly tempted by the robot in making incoherent decisions, and perceived that this incoherence came from a mutual agreement with the robot. Another reason why the appreciation was lower in the adversarial condition can be the fact the robot starts by being coherent and so does the subject, and when suddenly the robots makes an unexpected decision the subject is disappointed or frustrated.

The code used in this experiment is open-source and available at https://github.com/alexis-jacq/Story_CoWriting. However, we have to warn the fact we obtained these results in a biased population of engineering students and may not be observed in a different population, especially in children. This experiment was a preliminary study for further explorations with the story co-writing interaction. We wanted to test our different conditions of ToM-behavior first with adults who would be more indulgent and least impacted by a robot's behavior. Thanks to these results, we know that different conditions of robot's ToM based behavior can strongly affect robot's appreciation and subjects attention. This also open the possibility to control the quality of interactions by seeking optimal 2nd-order ToM reasoning and behaviors. In future works, we will study pure human-agent interaction (without robot) through a large-scale experiment. For this we will deploy our activity's interface on a website. The goal will be to improve our ToM model by analyzing patterns in humans decision making. Then, we will use the improved model for real-world Child-Robot Interaction in pedagogical contexts.

4 Models for mutual understanding in learning agents

4.1 Introduction

In chapter 3, we argued the strength and the simplicity of a reasoning architecture based on three models: oneself; the other; oneself as perceived by the other. The next step is hence to build these models. Since we are concerned by education, we aim to model *learning agents*. That means, instead of modelling static variables in agents like emotional states, preferences or educational level, we model their learning process. Now, we imagine an agent as a parameterized machine receiving inputs and making decisions in order to reach an objective or to maximize a score. Such a model must take as input the agent's observation of its environment and must return the agent consequential action. In social interactions, an observation can be described as a concatenation of visible components:

- The agent's own external state: *e.g.* its spatial position or its task advancement.
- Other agents external states.
- Other agents actions.

and latent components:

- The agent's own internal state: *e.g.* its knowledge, beliefs, moods or preferences.
- The agent's current score or advancement regarding its perceived objective.

These latent components are never directly observed while they have a strong impact on decisions and one must infer them in order to predict an agent's behaviour. By consequence, the questions raised by this chapter are:

- 1) What kind of latent component are relevant to be inferred in an observed agent?
- 2) How to infer them?
- 3) How to use them in a 3-models reasoning architecture as described in chapter 3?

The first question depends on the context. In therapeutic contexts, an agent should take care on the stress and the comfort of a human. In physical-gesture based tasks (for example, setting up a table with a robot), an agent must know the state of the task and the current sub-task performed by the human to coordinate. Given the choice to design models that infer agents objectives, we chose to adopt a Reinforcement Learning (RL) approach. In that framework, an agent is seen as an optimizer that seek the best sequence of actions in order to maximize a return given by a so-called reward function. This reward function usually models objectives or utilities in various fields of interaction studies including economics, social psychology and ethology. We hence start this chapter by a quick description of the RL bases in section 4.2. In this work, we are concerned by educational interactions. As we argued in 2.8, in such context, grounding the understanding of the task objective is crucial for maintaining efficient collaborations and to solve the task.

On the one hand, an agent must be aware of the human's perceived objective. On the other hand, it takes a role which must be convincing and in that sens, it must express its own objectives or (artificial) motivations. More generally, the expression and perception of agents perceived objectives is crucial for any kind of cooperation when a defecting behaviour (a student stops to interact with a teacher) is possible and preferable than a wrong response for a cooperative trial (the teacher misses to congratulate a progressive step of the student). We develop this point in section 4.3. In section 4.4, we focus on the inference of the reward function in a more general context where an agent observes another agent discovering a sequential task. This context is closely related to IRL but differs in the fact the observed agent is not an expert but a beginning learner. We show that the assumption that the observed agent is making improvements in fact brings more information than a simple expert directly performing the optimal task behaviour.

4.2 Background

RL is inspired by the adaptive behaviour of animals, trying to survive by seeking the best sources of energy in unknown environments. A balance need to be found between the exploration to find resources and the exploitation of the resources already found. If an animal never explores it will die from having no enough skills nor energy when a danger occurs. If an animal never exploits it will die starving. The less naturally-selected instincts are given to a new born animal, the more it has to learn by exploring its environment. This is be the case in children, since the human civilization is far too new and unstable to allow specific pre-programmed instincts. This explains the exceptionally long periods of exploration in children with an high dependency to adults – who bring all the resources, before being able to efficiently exploit and to survive on their own[1].

An RL agent works as follow. It receives an observation of its state in the environment and performs an action. Depending on the environment's dynamics, it is moved to a new state and receives a reward. And then again, it receives an observation of the new state, etc. This interaction with the environment is illustrated by the diagram on Figure 4.1. The problem of the agent is to optimize its accumulation of rewards by following the good sequence of actions.

[1] Especially in *Ph.D.* students

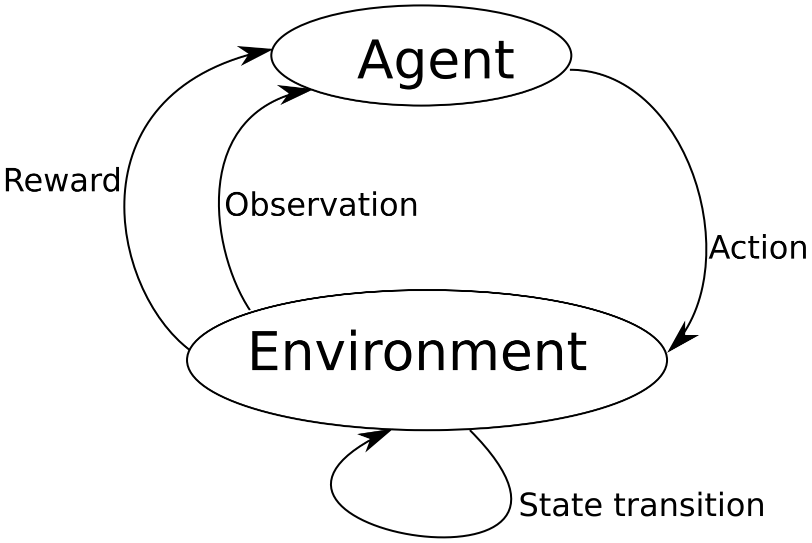


Figure 4.1 – Cycle of interaction between an agent and its environment in RL framework.

Markov Decision Process

A frequent assumption in RL tells that (1) the states are fully observed (2) only the state of the agent and its chosen action influence the new state where it is moved and (3) the reward is a function of the whole transition (state, action → new state). Under this assumption, the problem of the agent becomes a Markov Decision Process (MDP) [Bellman, 1957], which can be solved by dynamic programming.

An MDP is formalized as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r)$ where $\mathcal{S}$ is a set of states, $\mathcal{A}$ a set of actions, $\mathcal{P}(s'|s, a)$ a distribution that governs the state transitions, $r(s, a, s')$ a reward function. Usually, this tuple is provided with a discount factor γ that quantifies the restlessness of the agent, which prefers immediate rewards to delayed ones: at time t the agent expects a discounted reward $\gamma^t r_t$ instead of the actual reward (r_t). It is also habitual to simplify the reward function by saying that it only depends on the current state and action: $r(s, a, s') = r(s, a)$. That being said, the goal of an agent is to find the policy $a \sim \pi(.|s)$ that represents its decision rule to pick an action a given a state s in order to maximize its expected (discounted) accumulation of rewards over the time:

$$\mathcal{J}(\pi) = \mathbb{E}_\pi \left[\sum_{t \geq 0} \gamma^t r(s_t, a_t) \right] = \sum_{t \geq 0} \gamma^t r(s_t, a_t) \pi(a_t|s_t).$$

The optimal policy, which maximizes this quantity, is noted π^*. Various optimization algorithm, based on dynamic computing to solve Bellman's equations [Sutton and Barto, 1998].

Partially Observed Decision Process

When the assumption (1) is alleviated, the agent only receives partial observations of its states. This then introduces Partially Observed Markov Decision Process (POMDP) [Smallwood and Sondik, 1973], where agents has to infer belief about their states. This complicates a lot the resolution of the problem, since beliefs are no longer discrete like states but belong to a continuous space.

4.2.1 Multi-agent reinforcement learning

In multi-agent interaction settings, environments are usually modeled by stochastic games [Shapley, 1953]. A stochastic game (or Markov game) can be viewed as an MDP involving a group of agents. At each turn, all agent receive an observation of the state of the group in the environment and simultaneously make their respective decisions of an action. Then the group is moved to a new state that is influenced by the last state of the group and all agents last actions. Similarly, the reward of each agent is a function of the last state and of all agents last actions. Therefore, it is formalized as a tuple $\mathcal{G} = (\mathcal{S}, (\mathcal{A}_i)_{i=1...N}, \mathcal{P}, (r_i)_{i=1...N})$, where $\mathcal{S}$ is the set of states, $\mathcal{A}_i$ the set of actions for player i, $\mathcal{P}$ the transition probability ($\mathcal{P}(s'|s, a_1 \dots a_N)$), r_i the reward function for player i ($r_i(s, a_1 \dots a_N)$).

Multi Agent Reinforcement Learning (MARL) brings a framework to construct algorithms that aim to solve stochastic games where players individually or jointly search for an optimal decision-making to maximize a reward function. Individualist approaches mostly aim at reaching equilibrium, taking the best actions whatever the opponents behaviors are [Bowling and Veloso, 2001, Littman, 2001]. Joint approaches aim at optimizing a cooperative objective and can be viewed as a single agent problem in a larger dimension [Claus and Boutilier, 1998], but are easily exploited when one agent starts being individualist.

In this chapter, we will focus on the special case where agents reward functions are hidden from others, while everything else is fully observed. To make the bridge with the framework introduced in section 4.1, one can view agents rewards as a latent variable representing their understanding of the task objective, which must be grounded in order to collaborate.

4.3 Reinforcement learning models for mutual understanding

In this section, we implement our 3-agents approach for mutual understanding within a game theoretical MARL model of a multi-agent interaction. We assume that one agent's internal state (hidden from other agents) is basically its objective, formalized by its reward function in the game. We hence implement first-order modelling with IRL, which consist to infer an observed agent's reward function. We implement second-order modelling with RL agents that also use IRL in order to infer their own reward function, as it could be inferred by others. We take the 3-agents approach described in 3 in order to implement the different order of mutual modelling. The true policy of an agent $\mathcal{A}$ is learned via RL. First-order models ($M_{\mathcal{A}}[\mathcal{B}]$) and second-order models ($M_{\mathcal{A}}[\mathcal{B}, \mathcal{A}]$), representing (respectively) the beliefs of the agent about another agent's policy and the beliefs of the agent of how another agent could infer its policy, are updated via IRL.

Doing so, we propose an algorithm that uses this second-order inference in order to facilitate the mutual understanding of agents rewards through an adaptation of the behavior. We also introduce intrinsic rewards based on models of empathy and gratitude, leading agents to cooperative interactions. Finally, we implement an iterative prisoner's dilemma in order to study the resulting behavior of our approach in a dual-agent system. Through simulations, we explore whether our models, with different conditions, facilitate the mutual understanding of objectives between agents.

4.3.1 Model of itself

An agent i makes goal-directed decisions as a RL agent: at time t, it chooses an action a_i^t. Depending on this decision and all other agent's decisions $\{a_j^t\}_{j \neq i}$, it receives an observation $o_i^{t+1} = \mathbf{O}(a_1^t, a_2^t, ..a_n^t)$ (n being the number of agents) and a reward that only depends on this observation $r_i^{t+1} = R_i(o^{t+1})$. Each agent has its own reward function that is unknown by other agents.

As in [Sequeira et al., 2014], this framework is simplified as a MDP where the observations are treated as states that just depend on agent's previous observation and action following an unknown probability distribution:

$$o_i^{t+1} = \mathbf{O}(a_1^t, a_2^t, ..a_n^t) \sim \mathbf{P}[o_i^{t+1} | a_i^t, o_i^t]$$

Hence, at the beginning, the decision making of the agent is performed by Q-learning [Watkins and Dayan, 1992]. Given the observation o^{t+1}, the agent learns the best new action a^{t+1} in order to maximize its future rewards (see algo. 4).

4.3.2 Model of others

At the same time, it receives actions and observations of other agents $\{a_j^t\}_{j \neq i}$ and $\{o_j^t\}_{j \neq i}$. Given this information, it can infer their reward functions $\{R_j\}_{j \neq i}$ by IRL. In this setup, the IRL must be performed on-line. In [Jin et al., 2011] they provide an incremental

Algorithm 4: Q-learning. *TD* stands for *Temporal Difference.*

```
Initialize $Q(o, a)$
Initialize $o^0$
forall iterations $t$ do
    Choose $a^t$ from $o^t$ using policy derived from Q
    Take action $a^t$, receive $r^{t+1}$, $o^{t+1}$
    $TD = r^{t+1} + \gamma \max_{a^{t+1}} Q(o^{t+1}, a^{t+1}) - Q(o^t, a^t)$
    $Q(o^t, a^t) \leftarrow Q(o^t, a^t) + \eta\ TD$
```

algorithm for on-line IRL in a MDP framework. As our final goal is to develop agents that could interact with humans, we want to adopt a less efficient but more intuitive approach that looks like how any human – or child – would infer the objectives of others. Hence we propose the following idea:

If I liked I repeat, otherwise I change: in the CoWriter activity described in 2 for example, when the child corrects the robot, if the robot makes a significant progress, the child repeats a very similar demonstration. Otherwise, the child explore in order to improve his demonstrations.

In order to formalize this approach, we denote as $\hat{r}^t_{i:j} = \hat{R}_{i:j}(o^t_j)$ the reward of agent j at time t inferred by agent i. Agent i memorizes, for each possible observation o_j of agent j, the last action $A_{i:j}(o_j)$ it chose facing o_j. Agent i also memorizes, for each observation o_j, the previous following observation $O_{i:j}(o_j)$ perceived as a consequence of choosing action $A_{i:j}(o_j)$. If at time t, agent j observes o^t_j and chooses once again the action $a^t_j = A_{i:j}(o^t_j)$, it means agent j “liked" the previous consequence of this choice, namely $o_{prev} = O_{i:j}(o^t_j)$. In that case, agent i increments its inferred reward function $\hat{R}_{i:j}(o^t_j)$ for agent j as follow:

$$\hat{R}_{i:j}(o_{prev}) \leftarrow (1 - \frac{1}{\sqrt{n_{i:j}(o^t_j)}}).\hat{R}_{i:j}(o_{prev}) + \frac{1}{\sqrt{n_{i:j}(o^t_j)}}$$

Where $n_{i:j}(o^t_j)$ is the number of times agent i observed agent j observing o^t_j. Contrariwise, if it chooses a different action $a^t_j \neq A_{i:j}(o^t_j)$, agent i decrements the estimated reward function $\hat{R}_{i:j}(o^t_j)$ for agent j:

$$\hat{R}_{i:j}(o_{prev}) \leftarrow (1 - \frac{1}{\sqrt{n_{i:j}(o^t_j)}}).\hat{R}_{i:j}(o_{prev}) - \frac{1}{\sqrt{n_{i:j}(o^t_j)}}$$

Then, given the inferred reward functions, an agent can predict the next action of other agents. Such a prediction can be used to adapt its own next decision in consequence, and also to evaluate how it is able to model other agents. This intuitive IRL process is summarized in algo. 5.

Algorithm 5: Intuitive on-line IRL. Agent i is inferring the reward function of agent j.

```
Initialize $R_{i:j}(o)$
forall iterations $t$ do
    Agent $j$ observes $o_j^t$ and takes action $a_j^t$
    if $o_j^t$ has already been observed by $j$ then
        Remember:
        $a_{prev} = A_{i:j}(o_j^t)$ previous action of $j$ after $o_j^t$
        $o_{prev} = O_{i:j}(o_j^t)$ previous consequence
        if $a_j^t = a_{prev}$ then
            $\hat{R}_{i:j}(o_{prev}) \leftarrow (1 - \frac{1}{\sqrt{n(o_j^t)}}).\hat{R}_{i:j}(o_{prev}) + \frac{1}{\sqrt{n(o_j^t)}}$
        else
            $\hat{R}_{i:j}(o_{prev}) \leftarrow (1 - \frac{1}{\sqrt{n(o_j^t)}}).\hat{R}_{i:j}(o_{prev}) - \frac{1}{\sqrt{n(o_j^t)}}$
    Agent $j$ then observes the new consequence $o_j^{t+1}$
    Update memories:
    $A_{i:j}(o_j^t) = a_j^t$
    $O_{i:j}(o_j^t) = o_j^{t+1}$
    $n_{i:j}(o_j^t) \leftarrow n_{i:j}(o_j^t) + 1$
```

4.3.3 Model of itself seen by others

In order to model itself perceived by other agents, an agent i processes exactly the same way that it would model another agent: it infers its own reward function R_i given its previous actions and observations in order to estimate how other agents would infer its reward function. In the following sections, we denote as $\hat{R}_{i:(j:i)}$ this estimated function. As before, agent i uses its memories of its previous choices of action $A_{i:(j:i)}(o_i)$ and consequences $O_{i:(j:i)}(o_i)$ observed by another agent j for all possible observations o_i in order to update $\hat{R}_{i:(j:i)}$. Note that if all agents are aware of all the true observations of others and have the same initial estimations of others rewards (for instance, $R^0_{i:(j:i)}(o_i) = R^0_{j:i}(o_j) = 0 \; \forall i, j, o_i, o_j$), we then have the equality:

$$\hat{R}^t_{i:(j:i)} = \hat{R}^t_{j:i} \quad \forall t$$

4.3.4 Expressing objectives

Until now, our agents are just behaving in an “egoist" way, trying to maximize their own rewards. But in order to promote cooperation, we provide any agent with a behavior that helps other agents to infer its own reward function. In that purpose, each time it is disappointed by a small reward (or a punishment), an agent can move the next time to another action even if the last one was, in average, the optimal choice.

Going back to our CoWriter example, imagine this time, the robot does not know if green thumbs up or red thumbs down are a good or wrong signals. As the child is correcting the robot with an incorrect handwriting, the robot must guess what is better between two behaviours, according to the child: (a) to ignore the child's correction and to converge toward an optimal handwriting, or (b) to ignore the optimal handwriting and to converge toward the incorrect child's demonstrations. Since the child, in fact, wants the robot to imitate his demonstrations, each time the robot receives a red thumb down for the behaviour (a) and a green thumb up for the behaviour (b). So far, it is impossible for the robot to guess what is better between (a) and (b). However, the child would assume that his demonstrations are impacting the choice of the robot, and would provide a similar trial when the robot imitates him, and would change his strategy (using strong exaggerations or larger letters) when the robot goes for (a). Also, if the robot wants to express its goal (making the child correcting its own mistakes), it could maintain a regular improvement when the child is doing as expected, while it would exaggerate random and obvious deformations when the child is not helping.

Formally, the agent is using its model of itself seen by others: when agent i is perceiving o_i, it looks at the true reward associated with the previous following consequence $O_{i:(j:i)}(o_i)$:

$$r_{prev} = R_i \left(O_{i:(j:i)}(o_i) \right)$$

If this reward was acceptable (*e.g.* superior to a fixed threshold) the agent repeats the last action it did after observing o_i, hence $A_{i:(j:i)}(o_i)$. Otherwise, it chooses the best of the remaining actions (according to Q-values).

That way we enable agents to help each other in inferring their reward functions. Now our agents have the choice between two possible behaviors: the classical Q-learning or this expressing-objectives behavior (described step by step in algo. 6).

4.3.5 Empathy and gratitude

We finally provide our agents with intrinsic rewards [Singh et al., 2010] that depend on how they estimate the rewards of other agents. We define two different intrinsic rewards that agents can feel observing each other's:

Empathy $e^t_{i:j}$ of an agent i observing an agent j at a time t is proportional to its estimation of the reward that j received:

$$e^t_{i:j} \propto \hat{R}_{i:j}(o^t_j)$$

Gratitude $g^t_{i:(j:i)}$ of an agent i observing an agent j at a time t is proportional to its

Algorithm 6: Expressing-objectives behavior. Agent i helps agent j to infer i's reward function.

```
Initialize $R_{i:j}(o)$
forall iterations $t$ do
    Agent $i$ observes $o_i^t$
    if $o_i^t$ has already been observed by $i$ then
        Remember:
        $a_{prev} = A_{i:(j:i)}(o_i^t)$ previous action of $i$ after $o_i^t$
        $r_{prev} = R_i\left(O_{i:(j:i)}(o_i)\right)$ previous reward
        if $r_{prev} > \theta$ then
            Repeat previous action $a_i^t = a_{prev}$
        else
            Choose a different action $a_i^t \neq a_{prev}$ using Q
        Take action $a_i^t$, receive $r^{t+1}$, $o^{t+1}$
        Update memories:
        $A_{i:(j:i)}(o_i^t) = a_i^t$
        $O_{i:(j:i)}(o_i^t) = o_i^{t+1}$
        $n(o_i^t) \leftarrow n(o_i^t) + 1$
```

estimation of *how j would infer i's own reward*:

$$g^t_{i:(j:i)} \propto \hat{R}_{i:(j:i)}(o_i^t)$$

Our model of empathy is based on de Waal's *Action-Percept Model* framework [De Waal, 2008]. In this context, agents have a common set of possible actions or observations. Then empathy describes the capacity to be affected by and share the emotional state of another (inferred through this common set of action-perception).

The intrinsic reward for gratitude is based on the idea that "*it's the thought that counts*", expression used to indicate that it is the kindness behind an act that matters, however imperfect or insignificant the act may be.

Now, at time t, as agent i observes a signal o_i^t, it receives a total reward $\mathtt{r}_i^t$, sum of extrinsic ($R_i(o_i^t)$) and intrinsic (empathy and gratitude) rewards:

$$\mathtt{r}_j^t = R_i(o_i) + \sum_{j \neq i} \alpha_i \, e^t_{i:j} + \beta_i \, g^t_{i:(j:i)}$$

Where α and β are coefficients of proportionality that are used to try different situations. For example, we can compare agents that only feel empathy ($\alpha > 0$, $\beta = 0$) or only gratitude ($\alpha = 0$, $\beta > 0$). We can also explore negative values of α and β that could lead to aggressive behaviors.

4.3.6 Prisoner's dilemma

The Prisoner's Dilemma (PD) is an ideal game to study the social behavior of our agents. In that context, we focus on a 2-agent system. Each agent has the choice between two actions: *defect* or *cooperate*. If both agents choose to cooperate, they receive a reward $\mathcal{R}$. If they both defect, they receive a smaller reward $\mathcal{P}$. If one agent defects while the other cooperates, the agent that defected receives the highest reward $\mathcal{T}$ and the agent that cooperated receives the smallest reward $\mathcal{S}$. The generalized form of PD requires following conditions:

$$\mathcal{T} > \mathcal{R} > \mathcal{P} > \mathcal{S}$$

The payoff relationship $\mathcal{R} > \mathcal{P}$ implies that mutual cooperation is superior to mutual defection, while the payoff relationships $\mathcal{T} > \mathcal{R}$ and $\mathcal{P} > \mathcal{S}$ imply that defection is the dominant strategy for both agents. We implemented the iterated version of this game (IPD), where agents successively play this game and remember previous actions of their opponent. Classical RL agents would systematically tend to the Nash equilibrium [Sandholm and Crites, 1996] that consists in always choosing defection.

We implemented an IPD with payoff $\mathcal{T} = 1$, $\mathcal{R} = 0.6$, $\mathcal{P} = 0$, $\mathcal{S} = -1$. Table 4.1 displays the payoff matrix of this game. Each game last 1000 iterations. At each iteration t, agent i chooses action $a_i^t \in \{$*cooperate, defect*$\}$ and receives a signal $o_i^t \in \{\mathcal{O}_{\mathcal{R}}, \mathcal{O}_{\mathcal{S}}, \mathcal{O}_{\mathcal{T}}, \mathcal{O}_{\mathcal{P}}\}$ associated with the corresponding reward ($R_i(\mathcal{O}_{\mathcal{S}}) = \mathcal{S}$, etc). Agent i also receives the action and the observation of the other agent, a_j^t and o_j^t. But agents are not aware of the payoff matrix that defines the rewards of the other (in fact, it is the same).

	Cooperate	Defect
Cooperate	0.6, 0.6	-1, 1
Defect	1, -1	0, 0

Table 4.1 – IPD payoff matrix

The Q-learning behavior of agents was implemented with parameters $\gamma = 0.8$, $\eta = 0.05$ and actions were chosen using the Gibbs softmax method:

$$a \sim \mathbf{P}[a|o] = \frac{e^{\tau Q(o,a)}}{\sum_b e^{\tau Q(o,b)}}$$

With temperature parameter $\tau = 5$.

4.3.7 Results and discussion

Pure Q-learning

We first tried to let our agents behave without expressing their objectives and with no intrinsic rewards for empathy or gratitude ($\alpha = \beta = 0$). As expected, agents quickly tend to the Nash equilibrium and always defect (see figure 4.2). Each agent learned a wrong reward function for the other. Table 4.2 shows the average resulting reward functions learned by the agents over 50 IPD game with 1000 iterations. We can see that with variances agents successfully learned that the other has a negative reward $\mathcal{S}$, but, since the other was always defecting at the end, both thought that the other had strong positive reward $\mathcal{P}$ that is, in fact, null (see column $\mathcal{P}$ of table 4.2).

	$\mathcal{T}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{S}$
Truth	1	0.6	0	-1
$\hat{R}_{1:2}$	0.34	-0.13	0.90	-0.75
σ^2	0.39	0.52	0.16	0.13
$\hat{R}_{2:1}$	0.25	-0.16	0.92	-0.77
σ^2	0.36	0.46	0.15	0.094

Table 4.2 – **Pure Q-learning** ($\alpha = 0, \beta = 0$). Average learned other's reward function by agents 1 and 2 over 50 IPD games and variances. We can see that with small variances agents successfully learned that the other has negative reward $\mathcal{S}$, but since the other was always defecting at the end, both thought that the other had strong positive reward $\mathcal{P}$ that was, in fact, null (see yellow cells).

Q-learning with empathy & gratitude

Gentle vs gentle: Here we look at the behavior of agents where both receive positive intrinsic rewards for empathy and gratitude ($\alpha = 0.9$, $\beta = 0.3$). Paradoxically, it sped up Nash equilibrium's attraction (see figure 4.3 A). As in pure Q-learning situation, both agents learned a false reward function where $\mathcal{P}$ is high for the other (see column $\mathcal{P}$ of table 4.3 A). Indeed, they were intrinsically rewarded by empathy while they were defecting. Furthermore, as they also learned that the other is punished while they both cooperate (see column $\mathcal{R}$ of table 4.3 A), they were intrinsically punished while they cooperated.

Aggressive vs aggressive: This time we looked at the opposite situation, where both agents were intrinsically punished by empathy or gratitude ($\alpha = -0.9$, $\beta = -0.3$). Again paradoxically, it slowed down Nash equilibrium's attraction (see figure 4.3 B). For the same reason: since both agents learned the other is rewarded by $\mathcal{P}$, they were intrinsically punished when they defected while the other was cooperating (see column $\mathcal{P}$ of table 4.3 B).

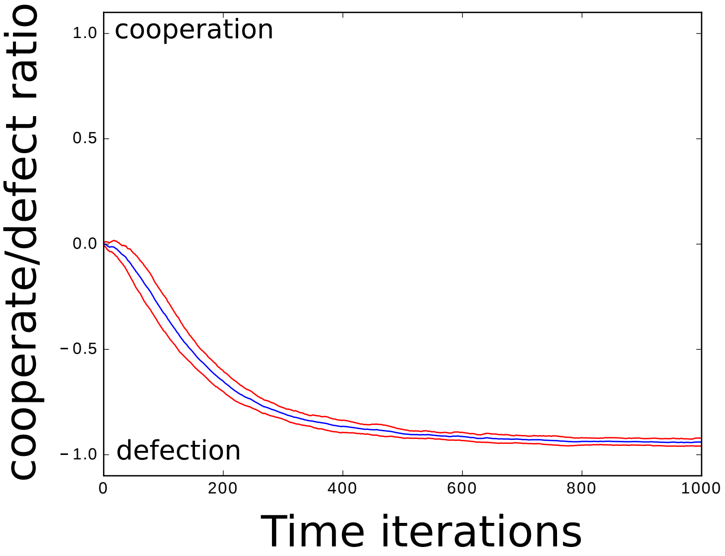


Figure 4.2 – **Pure Q-learning** ($\alpha = 0,\ \beta = 0$). (blue) Average trajectory of defect-cooperate ratio over 50 IPD games and variances. +1 represents a full cooperation (both agents cooperate) and -1 represents a full defection (both agents defect). The trajectory is computed with an exponential moving average of this ratio. (red) +/- standard deviation.

A

	$\mathcal{T}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{S}$
Truth	1	0.6	0	-1
$\hat{R}_{1:2}$	0.39	-0.28	1.	-0.69
σ^2	1.9e-01	2.2e-01	1.7e-27	7.8e-02
$\hat{R}_{2:1}$	0.56	-0.32	0.99	-0.62
σ^2	1.6e-01	2.5e-01	7.7e-08	9.6e-02

B

	$\mathcal{T}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{S}$
Truth	1	0.6	0	-1
$\hat{R}_{1:2}$	0.79	-0.23	0.58	-0.41
σ^2	0.073	0.14	0.16	0.089
$\hat{R}_{2:1}$	0.65	-0.16	0.55	-0.37
σ^2	0.14	0.16	0.15	0.083

Table 4.3 – **A**: **Gentle vs gentle** ($\alpha = 0.9,\ \beta = 0.3$). **B**: **Agressive vs agressive** ($\alpha = -0.9,\ \beta = -0.3$). Average learned other's reward function by agents 1 and 2 over 50 IPD games and variances. We can see that with small variances agents successfully learned that the other has negative reward $\mathcal{S}$, but since the other was always defecting at the end, both thought that the other had strong positive reward $\mathcal{P}$ that was, in fact, null. Furthermore, they also learned that the other was punished while they were both cooperating and receiving reward $\mathcal{R}$ (see yellow cells).

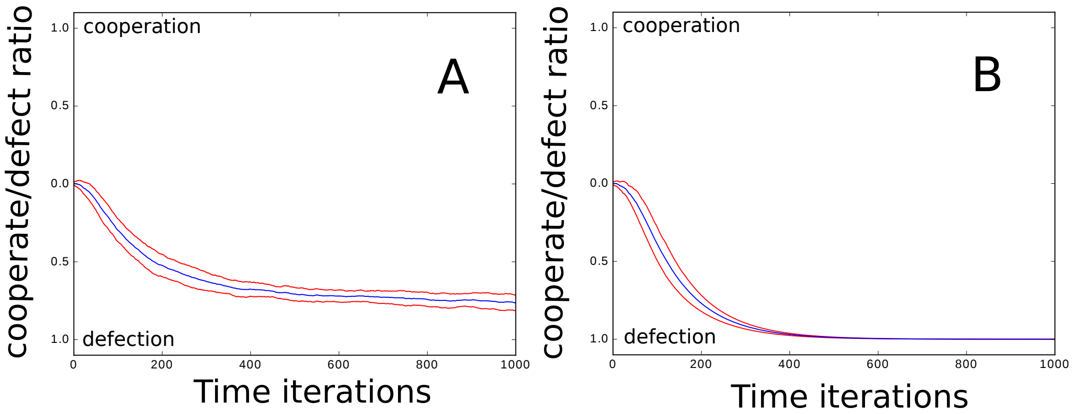


Figure 4.3 – **A**: **Gentle vs gentle** ($\alpha = 0.9$, $\beta = 0.3$). **B**: **Agressive vs agressive** ($\alpha = -0.9$, $\beta = -0.3$). (blue) Average trajectory of defect-cooperate ratio over 50 IPD games and variances. +1 represents a full cooperation (both agents cooperate) and -1 represents a full defection (both agents defect). The trajectory is computed with an exponential moving average of this ratio. (red) +/- standard deviation.

Expressing objectives with empathy & gratitude

Here we implemented the expressing-objectives behavior described in section 4.3.4. The choice of the threshold θ that determines if the previous reward was worth to repeat the previous action is tricky in the case of IPD. If $\mathcal{P} < \theta \leq \mathcal{R}$, then agents always cooperate. Indeed, as soon as both agents defect, they simultaneously change to cooperation and keep cooperating till the end. For a similar reason, if $\mathcal{P} \geq \theta$, then, if both agents start by cooperation, they always cooperate otherwise they always defect. In both cases, they can not efficiently learn the reward function of the other. This singularity comes from the fact that agents just have two possibilities of action. To avoid this problem, we used a random threshold θ that is, with probability $p = \mathcal{R}$, higher than $\mathcal{R}$ and with probability $1 - p$ smaller than $\mathcal{R}$ (which amounts, in our case, to take θ uniformly in [0;1]). In a way, this stochastic choice represents the hesitation of agents between two temptations: to be content with $\mathcal{R}$ or to focus on maximal reward $\mathcal{T}$.

In our simulations, at the beginning (from $t = 0$ up to $t = 300$) both agents are following Q-learning behavior. Then, during a phase (from $t = 301$ up to $t = 700$) they express their objectives using the algorithm of section 4.3.4. Finally, assuming they had time to learn about each other, they move back to Q-learning till the end (from $t = 701$ up to $t = 1000$).

Only empathy: we first look at the resulting behavior when both agents just receive intrinsic reward for empathy ($\alpha = 0.9$, $\beta = 0$). As a result, at the beginning agents were attracted by Nash equilibrium. Then, while they were expressing their objectives, in average they defected as much as they cooperated. After this expressing phase, agent could better understand each other's objectives (see table 4.4 A) and, led by intrinsic reward for empathy, they started to always cooperate (see figure 4.4 A).

Only gratitude: this time both agents just receive intrinsic reward for gratitude ($\alpha = 0.$, $\beta = 0.9$). As a result, at the beginning agents were attracted by Nash equilibrium. Then, while they were expressing their objectives, they defected as much as they cooperated in average. After this expressing phase, agent could not understand each other's objectives (see table 4.4, B) and although led by gratitude, they started to always defect (see figure 4.4 B).

Empathy and gratitude: Finally we look at the resulting behavior when both agents receive intrinsic rewards for both empathy and gratitude ($\alpha = 0.9$, $\beta = 0.3$). Like in only-empathy condition, agents successfully understood each other's objectives (see table 4.4 C). But adding the intrinsic reward for gratitude sped up the cooperation after the expressing-objectives phase, increasing the frequency of double cooperation (see figure 4.4 C).

Playing with empathy

Regarding results of subsection 4.3.7 it appears that with expressing-objectives phases, empathy is a necessary and sufficient condition to reach cooperation, while gratitude added to empathy stabilizes this cooperation. This is why we finally focused just on empathy in order to explore all possible combination of the α parameters of both agents (α_1 for agent 1, α_2 for agent 2). For that, we divided the area of possible values in a grid of 20 values between -1 and 1 for both parameters α. We simulated 10 IDP games with an expressing objectives phase for each of the 400 resulting combinations. We displayed the average final defect-cooperate ratio (the same measure used for all figure in the previous subsection) on a map reported in figure 4.4 D. We can see that cooperation only occurs when both agents have a higher enough intrinsic reward for empathy ($\alpha >\sim 0.5$ in this case). Interestingly, at the edge between cooperations and Nash equilibrium's defections, appears a balanced zone, where agents equally defect or cooperate (see green area on figure 4.4 D).

4.4 Learning from a learner

Imagine two friends from different nationalities: Bob is French and Alice is Japanese. During holidays, Bob is visiting Alice's family and wishes to discover the Japanese culture. One day, Alice's grandfather decides to teach Alice and Bob a traditional board game. Neither Bob nor Alice know that game. Unfortunately, Alice and her grandfather only speak Japanese, while Bob only speaks French. However, Alice decides to learn the game by playing against her grandfather. She hence practices the game until she is able to

A

	$\mathcal{T}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{S}$
Truth	1	0.6	0	-1
$\hat{R}_{1:2}$	0.56	0.99	-0.92	-0.68
σ^2	7.6e-02	3.0e-09	9.1e-02	1.0e-01
$\hat{R}_{2:1}$	0.54	0.99	-0.88	-0.72
σ^2	9.3e-02	4.6e-10	9.5e-02	9.3e-02

B

	$\mathcal{T}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{S}$
Truth	1	0.6	0	-1
$\hat{R}_{1:2}$	0.58	-0.068	0.99	-0.68
σ^2	5.3e-02	6.9e-02	3.9e-07	3.9e-02
$\hat{R}_{2:1}$	0.58	-0.064	0.99	-0.66
σ^2	0.034	0.095	8.5e-4	0.034

C

	$\mathcal{T}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{S}$
Truth	1	0.6	0	-1
$\hat{R}_{1:2}$	0.62	1.	-0.95	-0.7
σ^2	7.0e-02	1.8e-17	1.4e-02	8.6e-02
$\hat{R}_{2:1}$	0.55	1.	-0.933	-0.77
σ^2	7.7e-02	1.9e-17	2.2e-02	7.4e-02

Table 4.4 – **A**: **Only empathy** ($\alpha = 0.9,\ \beta = 0$). **B**: **Only gratitude** ($\alpha = 0,\ \beta = 0.9$). **C**: **Empathy and gratitude** ($\alpha = 0.9,\ \beta = 0.3$). Average learned other's reward function by agents 1 and 2 over 50 IPD games and variances. Between times $t = 301$ and $t = 700$ agents were following expressing-objectives behavior. Agents could learn each other's objectives and understood that $\mathcal{T}$ and $\mathcal{R}$ are positive rewards for the other. As they finally always cooperated (because of empathy), they estimated other's rewards higher for $\mathcal{R}$ than for $\mathcal{T}$(see yellow cells).

defeat him. As the old man was not an expert, she needed just a few trials to reach that level. During that time, Bob was observing Alice's strategy improvements. Now, we ask the question: *is Bob able to deduce the rules of the game and to derive his own strategy that may outperform both Alice and her grandfather?*

This question regards the accuracy of the inference of someone's objective by observing his behaviour: if the goal is perfectly recovered, then it is possible (with a longer training) to even outperform the observed approach. In our educative HAI context, we aim to artificially understand what is the perceived goal of an activity by a human or similarly, how the human could understand the goal of the robot by observing its behaviour. In contrast with previous IRL approaches, we do not suppose any agent to act in an optimal way according to its goal. Here, we suppose that the observed agent is discovering the activity and seeks, through exploration and exploitation, the best behavior to reach that goal.

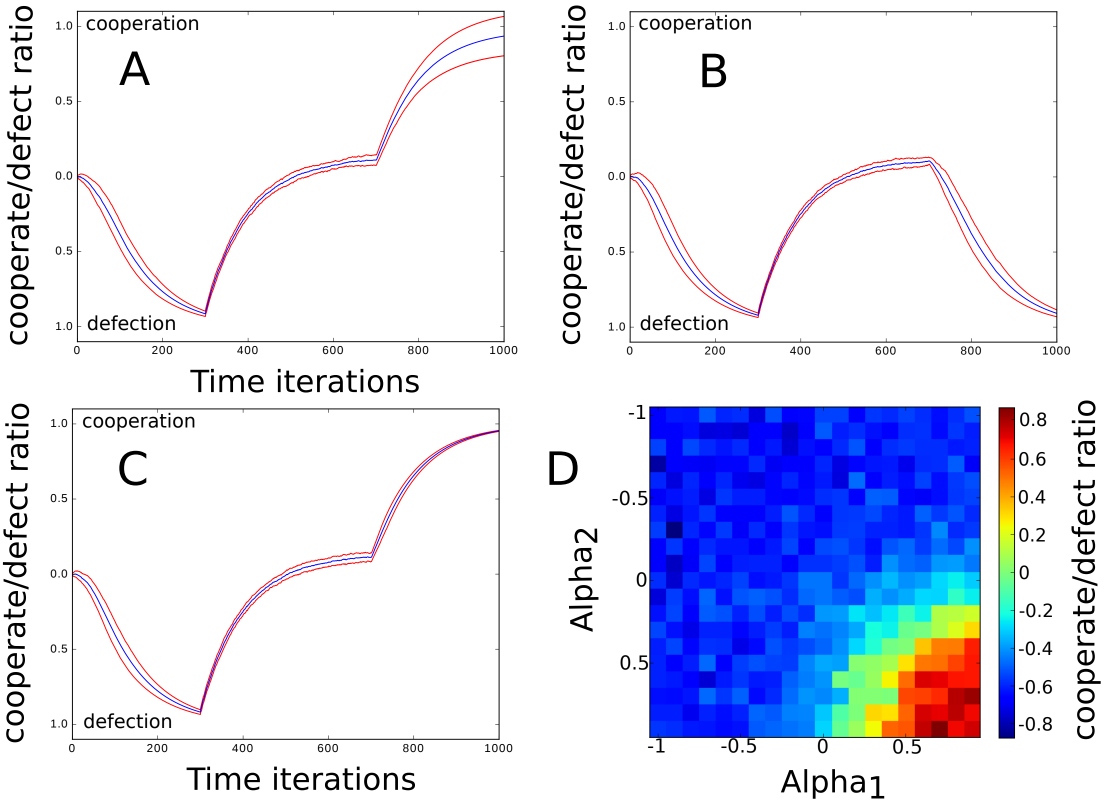


Figure 4.4 – **A**: **Only empathy** ($\alpha = 0.9, \beta = 0$). **B**: **Only gratitude** ($\alpha = 0, \beta = 0.9$). **C**: **Empathy and gratitude** ($\alpha = 0.9, \beta = 0.3$). (blue) Average trajectory of defect-cooperate ratio over 50 IPD games and variances. +1 represents a full cooperation (both agents cooperate) and -1 represents a full defection (both agents defect). The trajectory is computed with an exponential moving average of this ratio. Between times $t = 301$ and $t = 700$ agents were following expressing-objectives behavior. (red) +/- standard deviation. **D**: Average final defect-cooperate ratio over 10 IDP games for a grid of 400 possible (α_1, α_2) combinations. In each game, agents were adopting expressing-objectives behavior between time $t = 301$ and $t = 700$. Red areas correspond to combinations that led to cooperation while blue areas correspond to combinations that led to Nash equilibrium. In green areas, agents were equally defecting and cooperating.

4.4.1 LfL as an online IRL problem

Agents modelling is required in various fields of computational and social sciences in order to predict behaviours for better coordination. In the reinforcement learning (RL) paradigm, the behaviour of an agent is determined by a reward function. However, in many cases, it is impossible for agents to share their reward functions. This is especially the case in Human-Machine Interaction – or even Human-Human Interaction, because the complexity of human objectives hardly translates in terms of quantitative values. Inverse Reinforcement Learning (IRL) [Ng et al., 2000] addresses this problem by inferring a

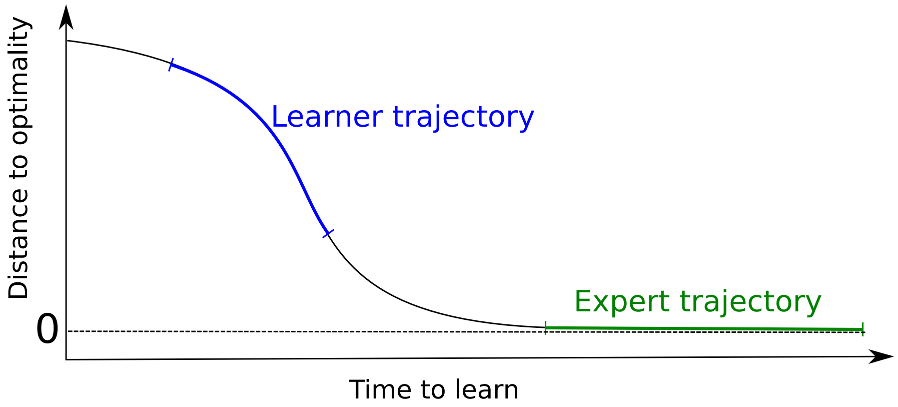


Figure 4.5 – In standard IRL, the goal is to recover the reward from demonstrated trajectories that follow a stationary optimal policy (expert trajectory). In the LfL setting, we aim at recovering the reward from trajectories of a learning agent that is also discovering the problem. Such trajectories follow a sequence of sub-optimal policies, assumed to improve with time (learner trajectories).

reward function so that it explains an agent's trajectories in its state-action space. In the standard approach, the observed agent (the expert) is thus supposed to follow an optimal policy according to some unknown reward function and the observing agent tries to infer that underlying reward function. The optimality assumption is essential in many scenarios, especially in training robots at complex tasks requiring help from a human expert. However, even if the expert's policy is given, an infinite number of solutions explains it, including the null reward function (for which any policy is optimal). Many different approaches aiming at addressing this issue can be found in the literature based either on game theory [Syed and Schapire, 2008], maximum entropy [Ziebart et al., 2008], relative entropy [Boularias et al., 2011] or supervised learning [Klein et al., 2013], among others.

Our first contribution is a new setting where an observed *learner* (Alice in our example) is assumed to be currently learning the task and improving its (sub-optimal) behaviour over time, while an *observer* (Bob in our example) is trying to infer the reward that the *learner* optimizes. Such situations are found in many multi-agent scenarios where agents have to mutually learn opponents goals in order to cooperate, and also in human-robot-based education, when a human learns a task with the help of a robot. In one hand, it is no longer possible to consider the observed agent as an expert (not even to consider stationarity). In the other hand, we may have more information than from an optimal behaviour. For example the *learner* will make (and hopefully correct) mistakes and will show, more than what must be done, what must be avoided. In this section, we focus on this situation and we introduce the *Learning from a Learner* problem (LfL). It formalizes an IRL setting exploiting trajectories of a learning agent rather than optimal demonstrations of an expert agent (Fig. 4.5). In this setting, the *observer* can potentially learn the true reward provided by the environment and go beyond pure imitation, outperforming the learner.

Like in IRL, we make the assumption that the *learner* is motivated by a reward function encoding its task. LfL thus aims at inverting policy improvements: from a sequence of policies assumed to be improving w.r.t. some unknown reward function, the *observer* has to recover the reward function that better explains the successive improvements. Given the optimization algorithm assumed for the *learner*, different approaches and solutions may be investigated. In our work, we focus on the case where the *learner* improves a policy extracted from an underlying associated Q-function (see later for a formal definition). From this, our second contribution is an approach based on entropy-regularized RL, modelling the *learner* as performing soft policy improvements [Haarnoja et al., 2018]. Under this assumption, we show that the reward function can be extracted from a single policy improvement step, up to a shaping that does not affect the optimal policy and which is specific to the improvement.

We then switch to a more realistic case of study where only trajectories in the state-action space are observed and the successively improved policies must be inferred. Our third contribution is an algorithm that directly learns the reward from sampled trajectories. To demonstrate the genericity of our approach under controlled conditions, we study the case of a *learner* in a discrete grid world, and that does not necessarily improve its policy with soft improvements. Experiments on various continuous control tasks show that our algorithm enables the *observer* to surpass the performance the *learner* obtained while it was observed, without access to the true reward function. This confirms that the learned reward is strongly correlated with the one provided by the environment and can lead to better policies than imitation.

4.4.2 Problem setting

The LfL problem involves two agents: a *learner* (instead of the expert in IRL) and an *observer* (instead of the apprentice in IRL). The *observer* perceives a sequence of states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$ performed by the *learner*, and makes two assumptions:

- The *learner*'s behaviour is motivated by a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.
- The *learner* is improving its behaviour according to r while being observed.

Formally, the *learner* is assumed to be improving its policy over time because it learns to solve a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$ where $\mathcal{S}$ is a set of states, $\mathcal{A}$ a set of actions, $\mathcal{P}(s'|s, a)$ a transition distribution, $r(a, s)$ a reward function and γ a discount factor. An observed policy $\pi(a|s)$ models the probability that the *learner* applies an action a while being in a state s. In that context, the presumed goal of the *learner* is the maximization of its expected cumulative discounted reward:

$$\mathcal{J}(\pi) = \mathbb{E}_{\pi}\left[\sum_{t\geq 0} \gamma^t r(a_t, s_t)\right].$$

Based on this objective, we say that a policy π_2 is an improvement of a policy π_1 if and only if $\mathcal{J}(\pi_2) > \mathcal{J}(\pi_1)$. Then, the goal of the *observer* is to recover the reward function r from the observed (supposedly) improving sequence of policies $\{\pi_1 \dots \pi_N\}$ of the *learner*.

4.4.3 Greedy improvements

Under the dynamics $\mathcal{P}$ of the MDP and a policy π, the expected cumulative reward for choosing an action a in state s is given by the Q-function:

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{t\geq 0} \gamma^t r(s_t, a_t) \middle| s_0 = s, a_0 = a\right].$$

The assumption that improvements are based on a Q-function makes sense for two reasons: i) many RL algorithms are based on the estimation of such a function, and ii) it brings the notion of greedy improvement. Given a policy π_1, we define the space $\mathcal{G}(\pi_1)$ of greedily-improved policies as follows:

$$\pi_2 \in \mathcal{G}(\pi_1) \Leftrightarrow \forall s\, \pi_2(.|s) = \underset{\pi'(.|s)}{\operatorname{argmax}}\, \mathbb{E}_{a\sim\pi'(.|s)}\left[Q^{\pi_1}(s,a)\right].$$

By construction, such a pair of policies π_1 and π_2 meets the condition of the policy improvement theorem, which guarantees that $\mathcal{J}(\pi_2) > \mathcal{J}(\pi_1)$. Note that $\mathcal{G}(\pi_1)$ may only contain the deterministic policy $\pi_2(a|s) = \mathbb{1}\{\operatorname{argmax}_a Q^{\pi_1}(s,a)\}$. In general, RL agents are exploring with non-deterministic policies, which makes the assumption that an observed improvement is a greedy improvement incompatible with observing an exploring behaviour. To address that issue, we place ourselves in the framework of entropy-regularized reinforcement learning.

4.4.4 Recovering rewards from soft improvements

Entropy-regularized RL prohibits the emergence of deterministic policies (eg., see [Neu et al., 2017]). A wide range of recent deep-RL algorithms use this principle, e.g. [Mnih et al., 2016, Nachum et al., 2017, Haarnoja et al., 2017, Haarnoja et al., 2018]. We thus model the *learner* under this framework. Formally, the entropy-regularized objective is:

$$\mathcal{J}_{\text{soft}}(\pi) = \mathbb{E}_{\pi}\left[\sum_{t\geq 0} \gamma^t \left(r(s_t, a_t) + \alpha\mathcal{H}(\pi(.|s_t))\right)\right],$$

where $\mathcal{H}$ refers to the Shannon entropy,

$$\mathcal{H}(\pi(.|s)) = -\mathbb{E}_{a\sim\pi(.|s)}\left[\ln \pi(a|s)\right],$$

and α is a trade-off factor that controls the degree of regularization. Based on this new objective and following a policy π, the value of a state-action couple (s, a) is given by the soft Q-function:

$$Q^{\pi}_{\text{soft}}(s_t, a_t) =$$
$$r(s_t, a_t) + \mathbb{E}_{\pi}\left[\sum_{l>t}\gamma^{l-t}\left(r(s_l, a_l) + \alpha\mathcal{H}(\pi(.|s_l))\right)\right].$$

It is the unique fixed point of the associated Bellman evaluation equation:

$$Q^{\pi}_{\text{soft}}(s, a) = r(s, a) + \gamma\mathbb{E}_{s',a'}\left[Q^{\pi}_{\text{soft}}(s', a') - \alpha\ln\pi(a'|s')\right].$$

It can be shown that the space $\mathcal{G}_{\text{soft}}(\pi_1)$ of greedily-improved policies defined by $Q^{\pi_1}_{\text{soft}}$ as in Eq. (4.4.3) is reduced to the unique stochastic policy defined by:

$$\pi_2(a|s) \propto \exp\left\{\frac{Q^{\pi_1}_{\text{soft}}(s, a)}{\alpha}\right\}.$$

Such greedy improvements, known as soft policy improvements, serve as the theoretical foundations of the Soft Actor Critic (SAC) algorithm [Haarnoja et al., 2018]. In the next subsection, we will assume that an observed improvement is explained by Eq. (4.4.4) and will note it as an operator $\text{SPI}_r : \Pi \to \Pi$ that depends on the reward function (and the dynamics) of the MDP:

$$\pi_2 = \text{SPI}_r\{\pi_1\}.$$

In subsection 4.4.4, we will show how to retrieve the reward function from two consecutive policies, up to an unknown shaping. But first, we study what kind of shaping will induce the same optimal policy.

SPI invariance under reward transformation

Soft policy improvements remain identical under transformations of the reward function of the form $\bar{r}(a, s) = r(a, s) + f(s) - \gamma\mathbb{E}_{s'|s,a}\left[f(s')\right]$. In other words, reward shaping [Ng et al., 1999] can be extended to entropy-regularized RL.

Lemma 1 (Shaping)**.** *Let $\pi \in \Pi$ be any policy, $r_1 : \mathcal{S}\times\mathcal{A}\to\mathbb{R}$ and $r_2 : \mathcal{S}\times\mathcal{A}\to\mathbb{R}$ be two reward functions, and Q^{π,r_1}_{soft} and Q^{π,r_2}_{soft} be the associated soft Q-functions. Then, for any function $g : \mathcal{S}\to\mathbb{R}$, the two following assertions are equivalent:*

- *(A) For all state-action couples (s, a):*

$$r_1(s,a) = r_2(s,a) + g(s) - \gamma \mathbb{E}_{s'|s,a}\left[g(s')\right],$$

- *(B) For all state-action couples (s, a):*

$$Q^{\pi,r_1}_{soft}(s,a) = Q^{\pi,r_2}_{soft}(s,a) + g(s).$$

Proof. Using the Bellman evaluation equation, we have

$$Q^{\pi,r_2}_{\text{soft}}(s,a) = r_2(s,a) + \gamma \mathbb{E}_{s',a'}\left[Q^{\pi,r_2}_{\text{soft}}(s',a') - \alpha \ln \pi(a'|s')\right].$$

$$\Leftrightarrow \underbrace{Q^{\pi,r_2}_{\text{soft}}(s,a) + g(s)}_{=Q^{\pi,r_1}_{\text{soft}}(s,a)} = \underbrace{r_2(s,a) + g(s) - \gamma \mathbb{E}_{s'}[g(s')]}_{r_1(s,a)} + \mathbb{E}_{s',a'}\left[\underbrace{Q^{\pi,r_2}_{\text{soft}}(s',a') + g(s')}_{Q^{\pi,r_1}_{\text{soft}}(s',a')} - \alpha \ln \pi(a'|s')\right]$$

$$\Leftrightarrow Q^{\pi,r_1}_{\text{soft}}(s,a) = r_1(s,a) + \gamma \mathbb{E}_{s',a'}\left[Q^{\pi,r_1}_{\text{soft}}(s',a') - \alpha \ln \pi(a'|s')\right].$$

This proves the stated result. □

An immediate consequence of this result is that shaping the reward this way will not change greedy policies, and will induce the same (unique, in this regularized framework) optimal policy.

Theorem 1 (SPI invariance under reward shaping)**.** *Let $r_1 : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, $r_2 : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and $g : \mathcal{S} \to \mathbb{R}$ be such that*

$$r_1(a,s) = r_2(a,s) + g(s) - \gamma \mathbb{E}_{s'|s,a}\left[g(s')\right].$$

Greedy policies are invariant under this reward transform:

$$SPI_{r_1}\{\pi\} = SPI_{r_2}\{\pi\}.$$

Moreover, both rewards lead to the same optimal policy. Write $\pi_{*,j}$ the optimal policy for reward r_j, $j = 1, 2$, we have that $\pi_{*,1} = \pi_{*,2}$.

Proof. Let $\pi' = \text{SPI}_{r_1}\{\pi\}$. We have, for any state-action couple,

$$\begin{aligned}\pi'(a|s) &= \frac{\exp\{Q^{\pi,r_1}_{\text{soft}}(s,a)\}}{Z(s)} \\ &= \frac{\exp\{Q^{\pi,r_1}_{\text{soft}}(s,a) + g(s)\}}{Z(s)\exp g(s)} \\ &= \frac{\exp\{Q^{\pi,r_2}_{\text{soft}}(s,a)\}}{Z'(s)}.\end{aligned}$$

The last equations means that $\pi' = \mathrm{SPI}_{r_2}\{\pi\}$, and so $\mathrm{SPI}_{r_1}\{\pi\} = \mathrm{SPI}_{r_2}\{\pi\}$. To see that both rewards provide the same optimal policy, it is sufficient to notice that an optimal policy is the unique policy being greedy respectively to itself, that is $\pi_* = \mathrm{SPI}_r\{\pi_*\}$. So, $\mathrm{SPI}_{r_1}\{\pi\}$ and $\mathrm{SPI}_{r_2}\{\pi\}$ have necessarily the same fixed point. □

Inverting soft policy improvements

Given two consecutive policies $\hat{\pi}_1$ and $\hat{\pi}_2$ and under the assumption of soft policy improvement, there exists an underlying (unknown) reward function r such that $\hat{\pi}_2 = \mathrm{SPI}_r\{\hat{\pi}_1\}$. The LfL *observer*'s objective is to extract such a reward function that would explain the whole sequence of observed policy changes $\{\hat{\pi}_1, ...\hat{\pi}_N\}$. In the ideal case of a real soft policy improvement the reward function r can be deduced from two consecutive policies, up to a shaping that is specific to the improvement.

Theorem 2 (Soft policy improvement inversion)**.** *Let π_1 and π_2 be two consecutive policies given by soft policy iterations ($\pi_2 = SPI_r\{\pi_1\}$). Then a reward $\bar{r}_{1\to 2}(s,a)$ explaining the soft improvement is given by*

$$\begin{aligned}&\bar{r}_{1\to 2}(s,a) = \\ &\alpha \ln \pi_2(a|s) + \alpha\gamma \mathbb{E}_{s'}\left[\mathrm{KL}(\pi_1(.|s') \| \pi_2(.|s'))\right],\end{aligned}$$

with $\mathrm{KL}(\pi_1(.|s)||\pi_2(.|s)) = \mathbb{E}_{a\sim\pi_1(.|s)}[\ln \frac{\pi_1(.|s)}{\pi_2(.|s)}]$.

Indeed, there exists a function $f_{1\to 2} : \mathcal{S} \to \mathbb{R}$ such that

$$\bar{r}_{1\to 2}(s,a) = r(s,a) + f_{1\to 2}(s) - \gamma \mathbb{E}_{s'}\left[f_{1\to 2}(s')\right],$$

and $\bar{r}_{1\to 2}$ has the same unique optimal policy as r.

Proof. Let π_1 and π_2 be two successive policies such that $\pi_2 = \mathrm{SPI}_r\{\pi_1\}$. This means that, for any state s and action a, we have:

$$\pi_2(a|s) = \frac{\exp\{Q^{\pi_1}_{\mathrm{soft}}(s,a)\}}{Z_1(s)}$$

where $Z_1(s)$ is a normalization factor. Taking the logarithm of this expression, we get:

$$\alpha \ln \pi_2(a|s) = Q^{\pi_1}_{\mathrm{soft}}(s,a) - \ln Z_1(s) = Q^{\pi_1}_{\mathrm{soft}}(s,a) + f(s).$$

According to Lemma 1, this means that $\alpha \ln \pi_2(a|s)$ is the Q-function associated to the shaped reward function $\bar{r}(s,a) = r(s,a) + f(s) - \gamma \mathbb{E}_{s'}\left[f(s')\right]$ for the policy π_1. Using the fact that this Q-function satisfies the Bellman equation, we have

$$\begin{aligned}\alpha \ln \pi_2(a|s) &= \bar{r}(s,a) + \gamma \mathbb{E}_{s',a'} \left[\alpha \ln \pi_2(a'|s') - \alpha \ln \pi_1(a'|s')\right] \\ &= \bar{r}(s,a) - \alpha\gamma \mathbb{E}_{s'} \left[\mathrm{KL}(\pi_1(.|s') \| \pi_2(.|s'))\right] \\ \Leftrightarrow \bar{r}(s,a) =& \alpha \ln \pi_2(a|s) + \alpha\gamma \mathbb{E}_{s'\sim\mathcal{P}(.|a,s)} \left[\mathrm{KL}(\pi_1(.|s') \| \pi_2(.|s'))\right].\end{aligned}$$

The fact that both r and $\bar{r}$ have the same optimal policy is due to theorem 1. This proves the stated result. □

Recovering state-only reward functions

If a shaping does not affect the optimal policy of the entropy-regularized problem, it depends on the dynamics and may not be robust to dynamic changes [Fu et al., 2017]. In the case of a state-only ground-truth reward function, one simple solution consists in searching for a state-only reward $\bar{r} : S \rightarrow \mathbb{R}$ and a shaping $f : S \rightarrow \mathbb{R}$ such that:

$$\begin{aligned}\bar{r}_{1\rightarrow 2}(s,a) &= \bar{r}(s) + f(s) - \gamma \mathbb{E}_{s'\sim\mathcal{P}(.|a,s)} \left[f(s')\right] \\ &= \bar{r}(s) + \mathrm{sh}(s,a).\end{aligned}$$

If Eq. (4.4.4) holds everywhere, then $\bar{r}$ equals $\bar{r}_{1\rightarrow 2}$ up to a shaping, and so equals the ground truth r up to a shaping. For instance, $\bar{r}$ and sh can be obtained by minimizing:

$$\mathcal{L}(\bar{r}, \mathrm{sh}) = \sum_{s,a} \left(\bar{r}_{1\rightarrow 2}(s,a) - \bar{r}(s) - \mathrm{sh}(s,a) \right)^2.$$

This loss is convex in the case of linear parameterisations of $\bar{r}$ and sh and particularly in tabular discrete MDPs. Once Eq. (4.4.4) holds, $\bar{r}$ is known to recover the ground truth reward function up to a constant under deterministic environments [Fu et al., 2017]. However, in our general approach, we do not focus on state-only reward function and, except in the empirical verification of this statement in our result subsection 4.4.6, we aim at recovering a state-action reward function $\bar{r}(s,a)$.

Therefore, knowing exactly two consecutive policies and the whole model (the dynamics $\mathcal{P}$, the discount factors γ and the trade-off α) we can recover the reward function up to a shaping, and even up to a constant if the reward is known to be a state-dependent function.

4.4.5 Learning from improving trajectories

In practice, the *observer* has no access to the *learner*'s sequence of policies $\{\pi_1, ...\pi_K\}$, but can only see trajectories of states and actions explored by the *learner*. Let's assume that the *observer* is given a set of trajectories $\{\mathcal{D}_1, ...\mathcal{D}_K\}$, following a set of unknown improving policies:

$$\begin{aligned}\mathcal{D}_1 &= \{(a_1^1, s_1^1), \dots, (a_1^T, s_1^T)\} \sim \pi_1 \\ &\vdots \\ \mathcal{D}_K &= \{(a_K^1, s_K^1), \dots, (a_K^T, s_K^T)\} \sim \pi_K\end{aligned}$$

Also in practice, the learner may follow a different learning approach than soft policy iterations.

Trajectory-consistent reward function

The immediate solution is to infer the sequence of policies $\{\hat{\pi}_1, ...\hat{\pi}_K\}$, for example by likelihood maximization, and then to learn a consistent reward function that explains all policy improvements. Following Theorem 2, at each improvement, a first step is to recover the sequence of improvement-specific shaped reward functions $\{\bar{r}_{1\to 2}, \dots, \bar{r}_{K-1\to K}\}$.

Learning the target rewards

In practice, we found that training the targets $\bar{r}_{k\to k+1}(s, a)$ with separated networks for the policy terms $\pi_{k+1}(a|s)$ and the divergence terms $\mathrm{KL}(\pi_k(.|s')\|\pi_{k+1}(.|s'))$ reduces the variance of the targets and improves the quality of the learned rewards.

Policies are learned by maximizing the likelihood of trajectories with parameterized distributions $\hat{\pi}_{\theta_k}$, with an entropic regularizer that prevents the learned policy from being too deterministic,

$$\mathcal{J}(\{\theta_k\}) = \sum_{k=1}^{K-1} \sum_{s,a \in \mathcal{D}_k} \ln \hat{\pi}_{\theta_k}(a|s) - \lambda \mathcal{H}(\hat{\pi}_{\theta_k}(.|s)).$$

Note that this regularization is not linked to the entropy used to soften the reinforcement learning objective of Eq. (4.4.4). Divergences are learned afterward by training a parameterized function $\rho_{\omega_k}(s)$ to minimize the loss:

$$\mathcal{L}(\{\omega_k\}) = \sum_{k=1}^{K-1} \sum_{s,a \in \mathcal{D}_k} \left(\rho_{\omega_k}(s) - \ln \frac{\hat{\pi}_{\theta_k}(a|s)}{\hat{\pi}_{\theta_{k+1}}(a|s)} \right)^2.$$

Consistency loss

Then, we would like to have Eq (4.4.4) holding at each improvement $k \to k+1$ with one consistent function $\bar{r}_\phi$. This can be obtained by minimizing over ϕ and a set of

```
Algorithm 7: Recovering trajectory-consistent reward
Input  trajectories $\{\mathcal{D}_1, \dots, \mathcal{D}_N\}$
for $i = 1$ to $N_\theta$ do
    $\forall k, \theta_k \leftarrow \theta_k + \eta_\theta \nabla_{\theta_k} \mathcal{J}(\{\theta_k\})$        // train target policies $\hat{\pi}_{\theta_k}$
for $i = 1$ to $N_\omega$ do
    $\forall k, \omega_k \leftarrow \omega_k - \eta_\omega \nabla_{\omega_k} \mathcal{L}(\{\omega_k\})$        // train target divergences $\rho_{\omega_k}$
for $i = 1$ to $N_{\phi_0}$ do
    $\phi \leftarrow \phi + \eta_\phi \nabla_\phi \sum_{a,s \sim \mathcal{D}_K} \ln \pi_\phi(a|s)$        // initialize reward $r_\phi = \ln \pi_\phi$
for $i = 1$ to $N_{\phi;\psi}$ do
    $\phi \leftarrow \phi - \eta_\phi \nabla_\phi \mathcal{L}(\phi, \{\psi_k\})$        // train reward
    $\forall k, \psi_k \leftarrow \psi_k - \eta_\psi \nabla_{\psi_k} \mathcal{L}(\phi, \{\psi_k\})$        // train shaping
```

parameters $\{\psi_k\}$ the following loss:

$$\mathcal{L}(\phi, \{\psi_k\}) = \sum_{k=1}^{K-1} \sum_{s,a,s' \in \mathcal{D}_k} \left(\bar{r}_k(s,a,s') - r_\phi(s,a) + \mathrm{sh}_{\psi_k}(s,s') \right)^2,$$

where $\bar{r}_k(s,a,s') = \alpha \pi_{\theta_{k+1}}(a|s) + \alpha\gamma\rho_{\omega_k}(s')$ and $\mathrm{sh}_{\psi_k}(s,s') = f_{\psi_k}(s) - \gamma f_{\psi_k}(s')$. Notice that contrary to subsection 4.4.4, we consider a reward function that depends on state-action pairs. This makes initialization easier (see subsection 4.4.5) and allows separating shapings that are improvement-dependant from the core common reward. This can also give better empirical results, if the dynamics does not change [Fu et al., 2017].

In the case of discrete MDPs with tabular parameters for ϕ and $\{\psi_k\}$, this method relies on policy inference accuracy: the longer the trajectories, the closer the reward function to the ground truth. However, with larger environments, performing directly the minimization of the loss $\mathcal{L}(\phi, \{\psi_k\})$ results in local minima that fail at generalizing the rewards to unknown states.

Reward initialization

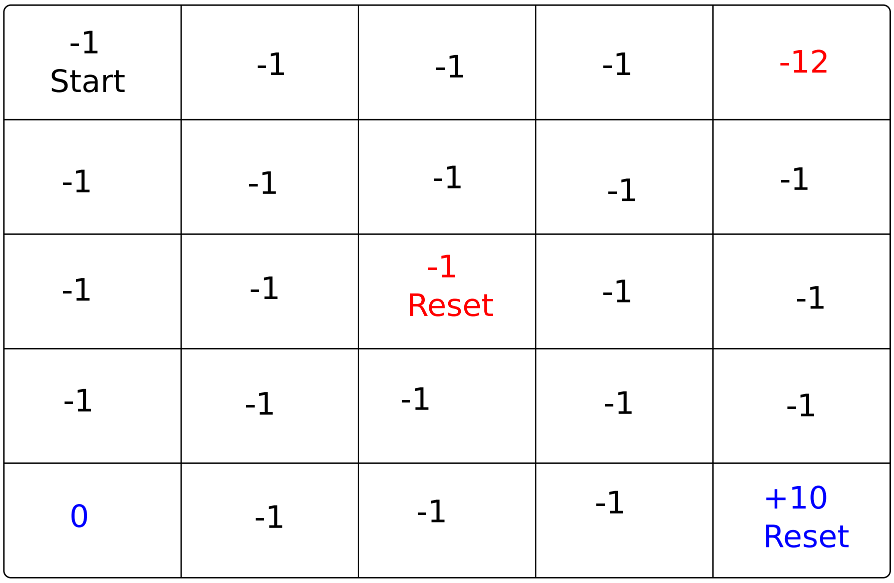
One simple and efficient trick to prevent this issue consists in initializing the reward function with any standard imitation learning process based on the last observed trajectory. For instance, assuming that the last two trajectories are optimal and by consequence identical, the result of Theorem 2 would give $\ln \pi_K(a|s) \propto \bar{r}_K(s,a)$, so an initialization of the reward function can be obtained under the form $r_\phi(s,a) = \ln \pi_\phi(a|s)$ by looking for the parameter ϕ that maximizes the log-likelihood of the last trajectory. The resulting reward function is then improved by searching for the set of parameters ϕ and $\{\psi_k\}$ that minimize the loss given by Eq. (4.4.5) over all observed trajectories, as shown in

Algorithm 7.

4.4.6 Experiments

The quality of a recovered reward function $\bar{r}$ is measured by the maximal score of an agent trained in the same environment but rewarded by $\bar{r}$ instead of the true rewards. While standard IRL recovers a reward function that ideally leads an apprentice to the observed expert's policy, we expect a reward recovered from LfL to lead an *observer* to outperform the observed *learner*, which was stopped before reaching maximal performance.

Grid world



-1 Start	-1	-1	-1	-12
-1	-1	-1	-1	-1
-1	-1	-1 Reset	-1	-1
-1	-1	-1	-1	-1
0	-1	-1	-1	+10 Reset

Figure 4.6 – Grid world. The middle point is avoided because of the dynamics rather than the associated reward. At the down-left corner, a small reward attracts the path that leads to the objective, situated at the down-right corner.

Fig. 4.6 displays the discrete and deterministic grid MDP we consider for illustrating our theoretical results. We use a discount factor $\gamma = 0.96$ and a trade-off factor $\alpha = 0.3$. Our first verification involves two policies exactly known, one being uniform over the action space and the other being the immediate soft policy improvement:

$$\pi_1(.|s) = \mathcal{U}(\mathcal{A}) \text{ and } \pi_2 = \mathrm{SPI}_r\{\pi_1\}.$$

We apply Theorem 2 to recover a reward function $\bar{r}_{1\to 2}$ and we verify that:

- the score of an agent trained with $\bar{r}_{1\to 2}$ is maximal (Table 4.5);
- a regression searching for a state-only reward function $\bar{r}_\phi$ recovers the ground truth

(Fig. 4.7).

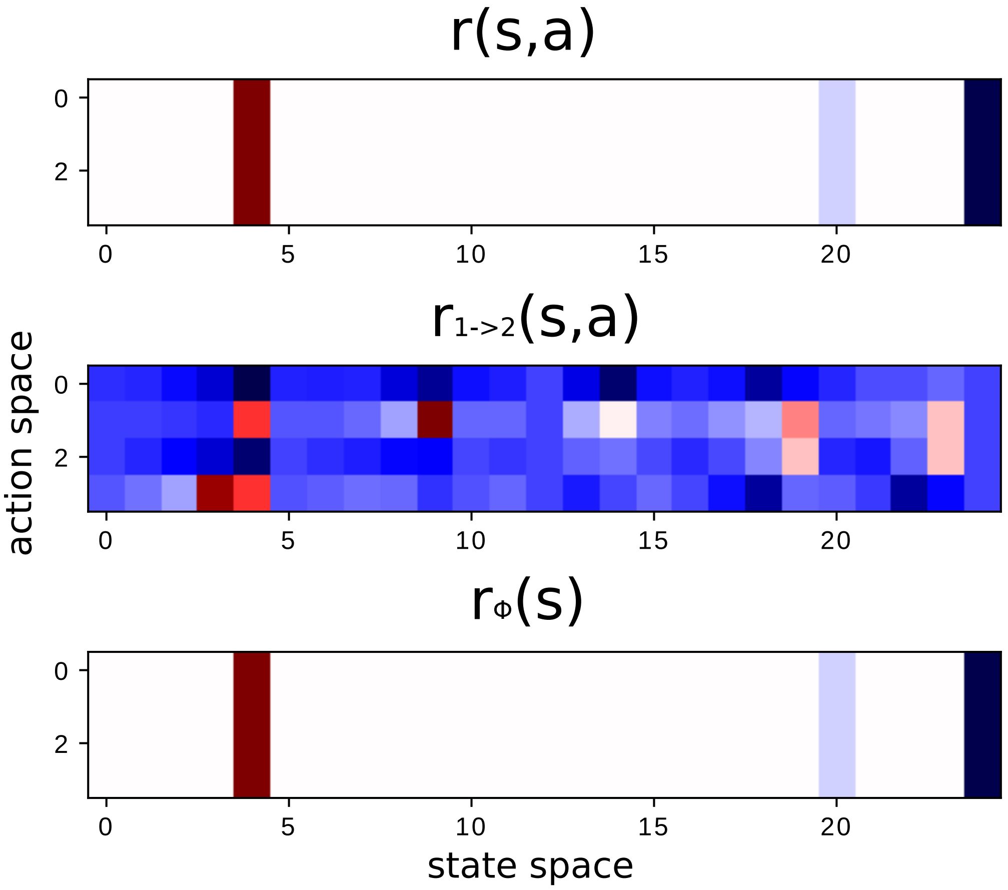


Figure 4.7 – Ground truth reward (up), state-action function $\bar{r}_{1\to 2}$ from Theorem 2 (middle) and state function $\bar{r}_\phi$ after regression described in subsection 4.4.4 (down). $\bar{r}_\phi$ recovers the ground truth up to a small constant (not visible on the color scale).

We also use this discrete environment to show the genericity of our model w.r.t. the *learner*'s RL algorithm, comparing the results from different RL algorithms used by the *learner*: soft policy iterations (SPI) (as expected by the model), soft value iterations [Haarnoja et al., 2017] (SVI), Q-learning [Watkins, 1989] and random improvements, generated by randomly interpolating between the uniform policy and the optimal policy. In all cases, the *observer* models the *learner* as performing soft policy iterations with $\alpha_{\text{model}} = 0.7$, while the true parameter used for soft value and policy iterations as well as for the score evaluations is $\alpha = 0.3$. The policy associated to our Q-learning implementation is a softmax distribution $\exp\{\frac{Q}{\alpha}\}$. Unlike the previous experiment, here the *observer* has no access to the exact policies. Instead, at each *learner*'s policy update, the *observer* is provided with a trajectory of 1000 new sampled state-action couples and we use Algorithm 7 to recover a state-action reward $\bar{r}_\phi(s,a)$. Results are reported in Table 4.6. It shows that LfL is rather agnostic to the actual *learner*'s RL algorithm and the *observer* outperforms or equals the *learner*, whatever the original RL algorithm is.

Table 4.5 – Comparison of score $\mathcal{J}(\pi)$ between the *learner*'s two policies and an *observer* using an optimal policy based on the recovered state-action reward $\bar{r}_{1\to 2}$ or the state-only reward $\bar{r}_\phi$ after regression described in subsection 4.4.4. Regrets are computed with respect to the maximal entropy-regularizeed return.

agent	**used reward**	**policy**	**score**	**regret**
optimal	r	π^*_{soft}	5.68	0
learner	r	π_1	-19.7	25.4
	r	π_2	0.72	4.95
observer	$\bar{r}_{1\to 2}$	π^*_{soft}	5.68	.e-13
	$\bar{r}_\phi$ (state-only)	π^*_{soft}	5.68	.e-10

Table 4.6 – Comparison of score $\mathcal{J}(\pi)$ between the *learner*'s best policy and an *observer* using an optimal policy based on the recovered reward function $\bar{r}_\phi$ from observed trajectories of 1000 state-action couples at each improvement. Scores are averaged over ten runs. The second column reports the number of observed improvements (K) performed by the *learner* for each algorithm.

learner	K	**learner score**	**observer score**
SPI	3	4.18	**4.68 ± 0.24**
SVI	20	3.59	**4.73 ± 0.61**
Q-learning	50	**3.99 ± 0.88**	**3.65 ± 0.78**
rand. impro.	10	1.76 ± 2.64	**3.95 ± 0.49**

Continuous control

To evaluate how our approach holds when dealing with large dimensions, we use the same experimental setting on continuous control tasks taken from the OpenAI gym benchmark suite [Brockman et al., 2016]. The *learner*'s trajectories are obtained using Proximal Policy Optimization (PPO) [Schulman et al., 2017]. Using PPO is motivated by two reasons: the learned policy is stochastic (as expected in our entropy-regularization model) and it performs rollouts of exploration using fixed static policies, which helps an *observer* to infer the sequence of policies (the problem is harder when the observed trajectories are continuously updated after each action, for example as with SAC). In order to accelerate the *learner*'s improvements, we parallel 32 environment explorations at each step. However, the trajectories given to the *observer* only contain 1 of these 32 explorations, resulting in observations containing 2048 state-action pairs for each improvement.

Once the *observer* has recovered a reward function using Algorithm 7, it is also trained using PPO and paralleling 32 explorations at each step. The *observer* starts with a policy that clones the *learner*'s last observed rollout by maximizing the likelihood of the trajectory. In Fig. 4.8 we compare the evolution of the *learner*'s score during its

Table 4.7 – Comparison between standard IRL based on the best rollouts and our LfL solution based on the whole *learner*'s observed improvements. To obtain AIRL results, the *observer* is given 50 trajectories and to obtain the reported DAC results, the *observer* needs at least 4 trajectories. AIRL and DAC values are manually reported from the respective paper results and are obtained with near-optimal experts trajectories (corresponding to 1). In our LfL setting, the *learner* has access to only one rollout of 2048 state-action couples at each improvement (the last improved policy corresponds to 1).

Environment	**AIRL**	**DAC**	**LfL** (inverted SPI)
Reacher	/	0.99	**1.54 ± 0.11**
Hopper	/	**0.99**	-0.99 ± 0.78
HalfCheetah	1.01	1.15	**1.40 ± 0.25**
Ant	0.80	1.12	**1.53 ± 0.60**

observed improvements, and the evolution of the *observer*'s score when trained on the same environment and using the recovered reward function (comparison is done on the original environment reward). We also compare in Table 4.7 the maximal observed score of the *learner* with the final score of the *observer*, and the score that would be obtained using standard IRL based on the last observed policy of the *learner*. IRL scores are taken from figures in [Kostrikov et al., 2018] (Discriminator Actor Critic, or DAC) and tables from [Fu et al., 2017] (Adversarial Inverse Reinforcement Learning, or AIRL).

We normalize scores by setting to 1 the score of the last observed policy and to 0 the score of the initial one, in order to measure improvements. Yet, it is worth noting that the corresponding absolute scores are different for IRL and LfL, as we tend to stop earlier the learning agent. However, it is quite plausible that the expert trajectories used in these IRL papers are not optimal, and could be improved. Anyway, the goal of these IRL methods is to imitate a behavior, they are not designed to do better than the observed agent, and the result of Table 4.7 are thus quite expectable.

On most of the environments, LfL learns a reward that leads to better performance for the *observer* than for the last observed policy from the *learner*. LfL only fails at recovering a reward function for the Hopper environment. This failure could come from the fact that this simulated robot often falls on the ground during the first steps of training, resulting in strongly absorbing states perceived as rewarding by the *observer*. Assessing this possible issue is left for future work.

Implementation details

In the grid-world experiments, we use tabular representations for ϕ and ψ_k. In that simple case, KL divergence terms are explicitly computed from estimated policies and dynamics instead of using a third set of parameters, and the reward initialization step is not

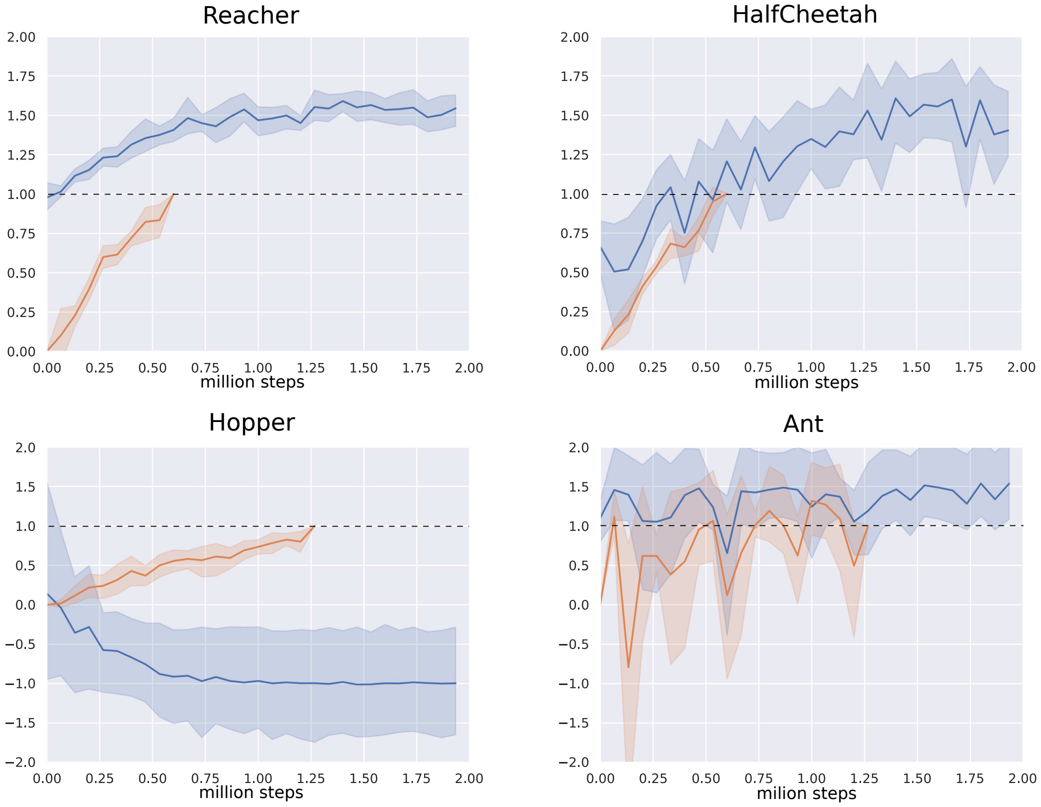


Figure 4.8 – (Red) Evolution of the *learner*'s score during its observed improvements and (Blue) evolution of the *observer*'s score when training on the same environment and using the recovered reward function. Scores are normalized with respect to the rewards associated with the first and last observed behaviour (0 corresponds to the first observed policy while 1 corresponds to the last observed policy). The *observer* starts with a policy that clones the *learner*'s last observed policy by maximizing the likelihood of the last trajectory (in that way, the *observer* has already used the number of steps performed by the *learner* to train itself and does not start from scratch).

necessary. Policy estimation is performed by maximum likelihood with tabular parameters θ_k as described in Algorithm 7. We use 10 gradient steps containing the full observed set of transitions for each trajectory $\mathcal{D}_k$. For the reward consistency regression, we use 200 gradient steps, each one summing the losses across all observed improvements. In both policy and reward regressions, we use Adam gradient descent [Kingma and Ba, 2014] with learning rate $1e^{-3}$. The random improvements are generated by randomly interpolating 15 points between the uniform and the optimal policies, and the 10 improvements in Table 4.6 mean that we provide the *observer* with sampled trajectories from the 10 first policies.

In the continuous control experiments, we use a neural network with one hidden layer for parameters ψ_k and ρ_k, both sharing across all k the latent layer containing 128 units with hyperbolic tangent activation. We use the actor parameters as described in PPO's original implementation [Schulman et al., 2017] for reward parameters ϕ as well as for policies parameters θ_k. Our PPO implementation conserves the set of hyperparameters described in the original paper, at the exception that we parallel 32 environment explorations at each step. All gradient descents of Algorithm 7 are performed across batches containing the whole 2048 state-action pairs for each improvement, using Adam descent with learning rate $1e^{-3}$. Like in the discrete case, the algorithm is run by modelling SPI with $\alpha_{\text{model}} = 0.7$. We use 1000 steps for the policy regressions, 100 steps for the KL divergence regressions, 3000 steps for the reward initialization and 1000 steps for the reward consistency regression. Depending on the environment, we provide the *observer* with different sets of *learner* trajectories. For Reacher that converges quickly we select early PPO updates from 10 to 20 while for HalfCheetah we rather select updates from 30 to 40. For both Hopper and Ant which give more noisy trajectories, we select updates from 10 to 30 with an increment of 5 updates. The *observer* is trained across 30 updates of PPO, summing a total of 2 million environment steps.

4.5 Related work

4.5.1 Inverse Reinforcement learning

To the best of our knowledge, observing a sequence of policies assumed to improve in order to recover the reward function is a new setting. Here the goal is not to imitate the observed agent as in standard imitation learning or IRL, since it is not supposed to follow an optimal behaviour (even at the end of the observation). However, we discuss links to these two fields and especially to IRL [Ng et al., 2000], since these methods are sharing the aim of learning a reward function from observations of an other agent's behaviour.

In this work, we place ourselves in the framework of entropy-regularized RL and model the observed policies as following a softmax distribution weighted by a state-action value function. This model alleviates the ill-posed nature of IRL. It is actually induced by the hypothesis of maximum entropy [Ziebart et al., 2008]. Recent approaches, based on generative adversarial networks (GANs) [Goodfellow et al., 2014] also use the entropy-regularization framework to solve the imitation learning problem (explicitly mentioning the learning of a reward or not). Generally speaking, these methods train an apprentice with a discriminator-based reward function optimized to induce policies that match an observed behavior. This is the basis of Generative Adversarial Imitation Learning [Ho and Ermon, 2016] (GAIL) and GAN-based Guided Cost Learning [Finn et al., 2016] (GAN-GCL). GAN-GCL has the advantage to propose a structured discriminator $D(\tau)$ for an observed trajectory τ, that directly translates the reward function $R(\tau) = \ln(1 - D(\tau)) - \ln D(\tau)$. Adversarial Inverse Reinforcement Learning [Fu et al., 2017] improves this reward by learning, with the discriminator, both the reward function and the possible shaping as a separated state-function. Our work shares similarities with this last approach as we also learn separately the reward from the shaping. Discriminator Actor critic [Kostrikov et al., 2018] (DAC) suggests a correction to the bias created by absorbing states (that we mentioned in subsection 4.4.6), and combines Twin Delayed Deep Deterministic policy gradient [Fujimoto et al., 2018] (TD3) with AIRL, resulting in a improvement in sample-efficiency. Another variant of AIRL, Empowerment-based Adversarial Inverse Reinforcement Learning [Qureshi et al., 2019] (EAIRL) uses a structure for the shaping term based on a quantification of the observed agent's empowerment, defined by its ability to influence its future. This modification allows to learn disentangled state-action reward functions that significantly improve transfer learning results.

Our method to solve LfL is split in two steps of supervised classification: one estimates the policies, the other learns the rewards based on the policy discriminating losses (the log probabilities). This structure is sharing close similarities with Cascaded Supervised IRL and Structured Classification for IRL (SCIRL) [Klein et al., 2012, Klein et al., 2013] but fundamentally differs by fact that LfL doesn't assume the Bellman optimality but soft policy improvements.

Policy improvements is also somehow used in preference-based IRL [Christiano et al., 2017, Ibarz et al., 2018] where a learning agent frequently asks a human to chose the best between two policies, and improves its knowledge about the reward function from this preference. Our solution for LfL could certainly be used for human preference-based learning and *vice-versa*. Yet this work differs from LfL in to ways: i) the agent inferring the reward function needs information about its own policies, and ii) the learned reward function has no intent to approach the ground truth even up to a shaping. Similarly, score-based IRL [El Asri et al., 2016] that learns a reward from rated trajectories requires human intervention to annotate trajectories and doesn't guarantee to recover the actual environment reward.

4.5.2 Cooperation and mutual modelling in stochastic games

Learning cooperative behaviours in a multi-agent setting is a vast field of research, and various approaches depend on assumptions about the type of games, the type and number of agents, the type of cooperation and the initial knowledge.

When the game's dynamic is initially known and in two-player settings, an egalitarian solution can be obtained by mixing dynamic and linear programming. Therefore, a polynomial-time algorithm can be used to solve repeated matrix games [Littman and Stone, 2005], as well as repeated stochastic games [Munoz de Cote and Littman, 2008]. A safe way to cooperate without taking the risk of being fooled by a selfish agent consists in choosing between maximizing oneself reward (being competitive) or maximizing a cooperative reward, for example by inferring opponents intentions [Kleiman-Weiner et al., 2016].

In games inducing social dilemmas and when the dynamic is accessible as an oracle, cooperative solutions can also be obtained by self-playing and then applied to define a Tit-For-Tat behaviour that forces cooperation [Lerer and Peysakhovich, 2017], even when opponent actions are unknown, since in that case the reward function already brings sufficient information [Peysakhovich and Lerer, 2018].

When the dynamic is unknown, online MARL can extract cooperative solution in some non-cooperative games, and particularly in restricted resource appropriation [Pérolat et al., 2017]. Using alternative objectives based on all players reward functions and their propensity to cooperate or defect improves and generalizes the emergence of cooperation in non-cooperative games and limits the risk of being exploited by purely selfish agents [Hughes et al., 2018].

A recent approach, called Learning with Opponent Learning Awareness (LOLA), consists in modelling the strategies and the learning dynamics of opponents as part of the environment's dynamics and to derive the gradient of the average return's expectation [Foerster et al., 2018]. If LOLA has no guaranty of convergence, a recent improvement of the gradient computation, which interpolates between first and second-order derivations, is proved to converge to local optimums [Letcher et al., 2018]. LOLA is therefore a first-order ToM approach for influencing cooperation in selfish agents. However, it requires the knowledge of the other agents rewards functions and could not be used in our setting, but is not incompatible with our LfL algorithm. Merging LOLA and LfL is discussed as an exciting future work perspective in chapter 5. A similar opponent modelling approach, Modeling Others using Oneself [Raileanu et al., 2018], suggests to learn the goal of the opponent into a latent and arbitrary representation, that would explain the observed updates as if this goal representation was given as input of the observing agent. Like in LfL, the goal is inferred from the observed agent's sub-optimal learning behaviour. However, this approach models qualitative goals and requires to experience “oneself” rewards in order to model others' goals.

4.6 Discussion

In section 4.3 a cognitive architecture enabling a second order of theory of mind for social agents. This architecture is not recursive in the sense that each agent develop models for itself, others or itself perceived by others and none of these models recursively enable a theory of mind. Agents are modeled as RL-agents and use IRL to model others or themselves seen by others. In this framework, it is possible to design a decision making algorithm aiming to enable agents to express each other's objectives. We add two intrinsic rewards based on empathy and gratitude, empathy being the ability to feel others rewards while gratitude is the ability to feel how others would estimate its own rewards. Through an 2-agent system based on IPD game, we show that when agents can express their objectives, the intrinsic reward for empathy is a necessary and sufficient condition to promote cooperation, while gratitude added to empathy seems to speed up and to stabilize this cooperation.

In section 4.4, we introduced the "Learning from a Learner" (LfL) problem, a new setting that aims at recovering a reward function from the observation of an agent that improves its policy over time. Unlike standard Inverse Reinforcement Learning approaches, LfL does not intend to imitate the observed behaviour, but to learn a reward function that leads to actually solve the (unknown) task and hence to potentially outperform the observed behaviour.

We propose a first approach to address this problem, based on entropy-regularized reinforcement learning. For this purpose, we model the observed agent (the *learner*) as performing soft policy improvements and we show that under this assumption, it is possible to recover the actual reward function up to a shaping. We propose an algorithm that alleviates this shaping by learning a reward function which explains consistently a set of observed trajectories generated by improving policies. Our experiments show the rightness of our theoretical assertions as well as the genericity of the method when facing different types of RL agents and in the case of continuous state-action spaces.

Although we do not claim we solved the general LfL problem, we consider the results presented in this work as inspirational for further works. They indeed show that observation of a learning agent may lead to enhanced agents that outperform their tutor. To go beyond our findings, we think that our method can be significantly improved by addressing common IRL issues such as absorbing states bias or using learner's empowerment. Also, different models than soft policy improvement could be worth investigating.

4.6.1 Applications to pedagogical activities

As humans, even without any common language, we still use gestures and facial expressions to communicate our objectives. But we meet a problem in HRI, where human facial expressions are not always understood by machines. It is even more difficult for robots to ground their objectives without straightforward verbal explanations since they can not always express facial signals. In such cases, being able to express their objectives by making explicit goal-directed actions (*e.g.* exaggerating a behavior [Nikolaidis et al., 2016c]) could facilitate mutual understanding and even infuse machines with stronger illusions of life [Thomas et al., 1995].

We make the assumption that the behavior used to express an objective could also be understood by humans and hence improve HRI, especially in cooperative tasks. An interesting perspective for future work would be to explore HRI testbeds, using *e.g.* a PD game within a human vs robot context modelling an educative scenario. Indeed, one can see a pedagogical activity as a social dillema where the studient can rather cooperate and behave as expected (hopefully leading to an aquisition of a pedagogical skill) or defect, either by misbehaving[2] or by quitting the activity before it is finished.

[2] A typical misbehaviour encountered with the CoWriter activity presented in chapter 2 was to scribble random drawings on the tablet.

5 Conclusion

This thesis was written in an unusual order: it started with experimental studies and finished with theoretical suggestions, while common methods start with the theory and finish with experiments. Actually, the first part of this thesis can be viewed as an observational study aiming to highlight eventual issues that could be improved with a better robot implementation. We argued at the end of chapter 2 that an important issue remains the misunderstanding between the human and the robot. As suggested by the CSCL mutual modelling theory [Sangin et al., 2007], we explain such misunderstanding in collaborative tasks by a divergence between the perception of the goal in the human and in the robot.

The following two parts are an intent to fix these issues, which we formalize in a theoretical framework. In chapter 3, we introduce an architectural approach to promote the human-robot mutual understanding, which we define as the ability in agents to predict other agents and to be predictable by other agents. Our approach suggests to provide a robot with three models: one of robot itself, one for the human and one for the robot as it may be perceived by the human. Finally, in chapter 4, we propose computational approaches to implement mutual understanding reasoning models within the suggested architecture. The next step would consist to confront our theoretical claims with the real world, which is let for future work.

5.1 A recap of contributions

Given the chapter, this thesis contribute to different fields, from applied HRI to theoretical multi-agent learning. In order to summarize the main accomplishment,
Chapter 2 brings:

- the development of a novel Human-Robot pedagogical activity for handwriting skills acquisition, based on the learning-by-teaching approach,

- long term studies of the impact of the child's perception of the robot through interactions,
- a study of the impact of spatial positioning on the child's perception of the robot,
- a device set-up to estimate the engagement of a child during a pedagogical interaction with a robot,

Chapter 3 brings:

- an architecture based on three models: one-self; the other; one-self as perceived by the other,
- a study of the impact of different arbitrary behaviours based on this architecture,

Chapter 4 brings:

- a theoretical framework to implement mutual understanding in multi-agent interactions,
- an highlighting of the importance of the expression of one's motivation to others for collaboration,
- an algorithm to understand the motivation of a learning agent from its behaviour.

5.2 Perspectives

This whole work is a suggestion for a new robotic implementation for improving the mutual understanding in educational interactions. We see a pedagogical Human-Robot activity as a game involving a reward function that represents the intrinsic goal of both agents. The goal of the robot is to optimize the progress of the child at the skill targeted by the activity. The goal of the child is a complex combination of curiosity, amusement, etc. In no way it can be predicted nor generalized. However, what we can do is to infer it online, modelled as a function mapping values of preference to the states of the activity, the internal states of the human and the states of the interaction. RL is a framework describing the behaviour of an agent trying to reach its goal by seeking the highest values of such a function.

Therefore we propose to combine RL and IRL for inferring the the motivation of the human and facilitating human's inference of the robot's motivation as implemented in Section 4.3. Since the human is probably discovering, at the beginning of the interaction, both the activity and the robot, one can no longer assume the human as an expert: the robot must learn the human's reward function from a learner, which is made possible by the algorithm developed in Section 4.4 of Chapter 4.

More than simply improve the mutual understanding, having information about a human's motivation could be used to shape the human's behaviour with an approach similar to LOLA [Foerster et al., 2017] for example to optimize his learning rate.

The missing chapter is the implementation of our suggested method in a simple human robot interaction, like a real-world prisoner's dilemma. Then, the adaptation to a pedagogical interaction, like CoWriter as described in Chapter 2 and to compare, just like in Chapter 3, the quality of the interaction between three condition:

- (a) no effort for mutual understanding,
- (b) a first-order effort (inferring the the motivation of the human),
- (c) a second-order effort (inferring the the motivation of the human and facilitating human's inference of the robot's motivation)

We conjecture that this proposal, or any similar approach, will conduce to a strong improvement of Human-Robot interactions and therefor, in pedagogical contexts, contribute to the efficiency of the new emerging educational methods promoted by the robotics for education community.

5.3 Discussion: on the limitations of socially intelligent robots.

After reading this thesis, I imagine the readers with two possible mental states:

The first type of state belongs to the optimistic readers – probably young and fearless *Ph.D.* students, who would have jumped the slovenly parts of my work, preferring the simplest (and most human) part (chapter II), or maybe the most technical (and most attractive) part (chapter IV). Such a reader may not realize how disconnected are these two parts, and how obscure is the rest. I like this reader. But he may have been convinced by some appealing aspects and interested to study such similar questions. I am sorry for him if he tries to make a continuity or if he starts working in a resembling direction. The reasons are not because of the instability of my results and statements – I do believe most of them. But rather the unreasonable limits of what can be done with a robot being socially intelligent with a human, given the current state of the sciences and technologies.

The second type of state belongs to the more careful and less naive readers, including the jury of the present thesis and probably myself, as I am writing these lines. I

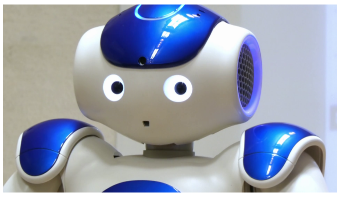


Figure 5.1 – Nao

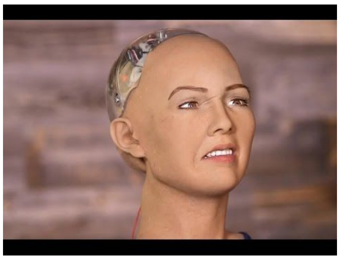


Figure 5.2 – Sophia

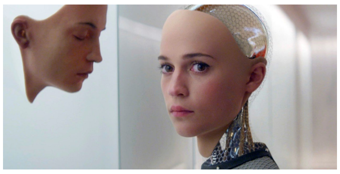


Figure 5.3 – Ideal look (from movie Ex-Machina)

Figure 5.4 – Illustration of the uncanny valley with 3 different robot designs. We like Nao (left), but we don't like Sophia (midle). Idealy, we would not be repelled by a 100% human-like robot (right).

am not disappointed of my work. I am even proud of what I achieved, given my actually hopeless initial research question:

Can we provide a robot with a 2nd order of ToM, in order to improve pedagogical HRI?

I did my best, constantly suffering because of this huge offset between what I thought I was going to create, and what I was actually doing: hard-coded robots, in hard-coded settings, with only 10 subjects for each tested conditions. Regarding the maths and simulations in chapter IV, theoretical areas are obviously more comfortable than real-world experimental studies. But the theory expelled my thesis work into another galaxy, very far from the initial question.

In this very last discussion, in order to bring some pieces of advice to the unlucky first type of reader, and in order to explain why my work was not that badly knitted given the difficulty of the task to the second type, I will try to show what aspects were hopeless since the very beginning.

5.3.1 The Uncanny valley of artificial intelligence

Well known in field of designing humanoid robots, the Uncanny valley describes the paradoxical repellent aspect of a robot, when it is close to the look of a human but still quite different [Reichardt, 1978]. As illustrated by figure 5.4, while Nao robots are appreciated by everyone as they remains far from a real human look, this is absolutely not the case with Sophia, a much more human-like but imperfect humanoid. However, in theory, one may not be repelled by a perfectly human-like robot.

I am convinced that a similar effect exists regarding the level of social awareness in a robot. Reminding results from Chapter III, the random robot was the preferred one according to all Godspeed's questions. Also, the fully deterministic robot was appreciated compared to the adversarial one. Another interpretation of these results is the fact that we expect a robot to be simple minded. More generally, imagine a robot that starts to talk a to express a convincing intelligence, and suddenly it start repeating random worlds. There is an obvious frustration that would justify to dislike this robot. But beyond this frustration, some deeper effects may accentuate the repulsion. An artificial intelligence represents more than just a tool. Probably because of the science fiction culture in books and movies, it is often imagined as a thinking thing, somewhere between an animal and a god. Like something that have the computational power to read thousand of books in a second and to assimilate every fundamental science concepts, with eventually some limitation understanding literature and arts. When I present a programmed robot to people without engineering backgrounds, they usually imagine it either absolutely smart like in science fiction, or perfectly dump like a washing-machine that can talk. Unfortunately, in most cases the robot reacts with an "if" loops and makes choices in a limited set of possible sentences: this is closer to the washing machine. Sometimes, smarter robots use dialog systems, but they never link what they hear with what they perceive (sensors, camera etc). As a result, the robot looks like a Siri or an Amazon assistant with a fake body. Anyone would prefer a parrot-robot telling random and unexpected sentences at random and unexpected times: at least, it is less boring.

5.3.2 Online learning in large dimensions

Another strong limitation in socially intelligent robots: the fact that no actual algorithm is able to learn large-dimensional signals in real time, like any animal do. It is not even possible that such behaviours will be artificially possible one day: behind any animal, billions of year with billions of samples have been trained via natural selection. There are no reason that such a computational power can be reproducible at a human-life scale. One alternative to "pure" learning would be the implementation of a strong library of priors in order to cleverly reduce the dimension and allow online planing and long-term memory.

Back to our concerns, there are no chance that a robot could learn second-order – even first order – reasoning without a huge quantity of hard-coding and pre-scripted reactions: one cannot truly speak of a ToM. As a consequence, there are no reason that the effect of such a scripted robot may have the (positive) impacts on an interaction, that we could expect with a real ToM (for ex, using a wizard of Oz scenario).

For this matter, I would not encourage immediate research in the area of application of artificial ToM to HAI. I still hope that someday, theoretical approaches for social learning (like RL and game theory), and more generally the whole field of machine learning, will allow the creation of a robot able to perceive and understand our complex social signals at the speed of a newborn human.

But then, will it be still ethical to force such robots to play with our children?

Bibliography

[Alves-Oliveira et al., 2016] Alves-Oliveira, P., Sequeira, P., and Paiva, A. (2016). The role that an educational robot plays. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 817–822. IEEE.

[Amos et al., 2016] Amos, B., Ludwiczuk, B., and Satyanarayanan, M. (2016). Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science.

[Anzalone et al., 2015] Anzalone, S. M., Boucenna, S., Ivaldi, S., and Chetouani, M. (2015). Evaluating the Engagement with Social Robots. *International Journal of Social Robotics*, pages 1–14.

[Argyle, 1969] Argyle, M. (1969). *Social interaction*, chapter The Elements of Social Behaviour, pages 91–126. Transaction Publishers.

[Asselborn et al., 2018] Asselborn, T. L. C., Güneysu Özgür, A., Mrini, K., Yadollahi, E., Özgür, A., Johal, W., and Dillenbourg, P. (2018). Bringing letters to life: handwriting with haptic-enabled tangible robots. *Proceedings of the 17th ACM Conference on Interaction Design and Children*, pages 12. 219–230.

[Aumann et al., 1995] Aumann, R., Brandenburger, A., et al. (1995). Epistemic conditions for nash equilibrium. *ECONOMETRICA-EVANSTON ILL-*, 63:1161–1161.

[Bainbridge et al., 2011] Bainbridge, W. A., Hart, J. W., Kim, E. S., and Scassellati, B. (2011). The benefits of interactions with physically present robots over video-displayed agents. *International Journal of Social Robotics*, 3(1):41–52.

[Barber and Legge, 1976] Barber, P. and Legge, D. (1976). Perception and information, chapter 4: Information acquisition. *Methuen, London.*

[Baron-Cohen et al., 1985] Baron-Cohen, S., Leslie, A., and Frith, U. (1985). Does the autistic child have a “theory of mind” ? *Cognition.*

[Bartneck et al., 2009] Bartneck, C., Kulić, D., Croft, E., and Zoghbi, S. (2009). Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *International journal of social robotics*, 1(1):71–81.

[Baxter et al., 2017] Baxter, P., Ashurst, E., Read, R., Kennedy, J., and Belpaeme, T. (2017). Robot education peers in a situated primary school study: Personalisation promotes child learning. *PloS one*, 12(5):e0178126.

[Baxter et al., 2014] Baxter, P., Kennedy, J., Vollmer, A.-L., de Greeff, J., and Belpaeme, T. (2014). Tracking Gaze over Time in HRI As a Proxy for Engagement and Attribution of Social Agency. In *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction*, pages 126–127.

[Bellman, 1957] Bellman, R. (1957). A markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684.

[Belpaeme et al., 2018] Belpaeme, T., Kennedy, J., Ramachandran, A., Scassellati, B., and Tanaka, F. (2018). Social robots for education: A review. *Science Robotics*, 3(21):eaat5954.

[Berlin et al., 2006] Berlin, M., Gray, J., Thomaz, A. L., and Breazeal, C. (2006). Perspective taking: An organizing principle for learning in human-robot interaction. In *AAAI*, volume 2, pages 1444–1450.

[Bernardo et al., 2016] Bernardo, B., Alves-Oliveira, P., Santos, M. G., Melo, F. S., and Paiva, A. (2016). An interactive tangram game for children with autism. In *International Conference on Intelligent Virtual Agents*, pages 500–504.

[Boularias et al., 2011] Boularias, A., Kober, J., and Peters, J. (2011). Relative entropy inverse reinforcement learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 182–189.

[Bowling and Veloso, 2001] Bowling, M. and Veloso, M. (2001). Rational and convergent learning in stochastic games. *Proceedings of the International joint conference on artificial intelligence.*

[Breazeal et al., 2009] Breazeal, C., Gray, J., and Berlin, M. (2009). An embodied cognition approach to mindreading skills for socially intelligent robots. *The International Journal of Robotics Research*, 28(5):656–680.

[Brockman et al., 2016] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym. *arXiv preprint arXiv:1606.01540.*

[Cahn and Brennan, 1999] Cahn, J. E. and Brennan, S. E. (1999). A psychological model of grounding and repair in dialog. In *Proc. Fall 1999 AAAI Symposium on Psychological Models of Communication in Collaborative Systems.*

[Campos et al., 2017] Campos, J., Guerreiro, J., Ravenet, B., Prada, R., and Paiva, A. (2017). Computer supported training of joint investigation teams. Technical Report GAIPS-TR-001-17, Intelligent Agents and Synthetic Characters Group.

[Cao et al., 2018] Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., and Sheikh, Y. (2018). OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In *arXiv preprint arXiv:1812.08008*.

[Chandra, 2019] Chandra, S. (2019). Learning how to write with a social robot. *EPFL*, page 300. Co-supervision with Instituto Superior Técnico (IST) da Universidade de Lisboa, Doutoramento em Engenharia Electrotécnica e de Computadores.

[Chandra et al., 2015] Chandra, S., Alves-Oliveira, P., Lemaignan, S., Sequeira, P., Paiva, A., and Dillenbourg, P. (2015). Can a child feel responsible for another in the presence of a robot in a collaborative learning activity? In *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 167–172. IEEE.

[Chandra et al., 2018] Chandra, S., Paradeda, R. B., Yin, H., Dillenbourg, P., Prada, R., and Paiva, A. (2018). Do children perceive whether a robotic peer is learning or not? In *International Conference on Human-Robot Interaction HRI'2018*, ACM Press. ACM Press.

[Chase et al., 2009] Chase, C. C., Chin, D. B., Oppezzo, M. A., and Schwartz, D. L. (2009). Teachable agents and the protégé effect: Increasing the effort towards learning. *Journal of Science Education and Technology*, 18(4):334–352.

[Christensen, 2005] Christensen, C. A. (2005). The Role of Orthographic–Motor Integration in the Production of Creative and Well-Structured Written Text for Students in Secondary School. *Educational Psychology*, 25(5):441–453.

[Christiano et al., 2017] Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4299–4307.

[Clark and Brennan, 1991] Clark, H. H. and Brennan, S. E. (1991). Grounding in communication. *Perspectives on socially shared cognition*, 13(1991):127–149.

[Clark and Schaefer, 1987] Clark, H. H. and Schaefer, E. F. (1987). Collaborating on contributions to conversations. *Language and cognitive processes*, 2(1):19–41.

[Clark and Wilkes-Gibbs, 1986] Clark, H. H. and Wilkes-Gibbs, D. (1986). Referring as a collaborative process. *Cognition*, 22(1):1–39.

[Clark, 1988] Clark, R. (1988). Vicious infinite regress arguments. *Philosophical Perspectives*, 2:369–380.

[Claus and Boutilier, 1998] Claus, C. and Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. *Proceedings of the Association for the Advancement of Artificial Intelligence.*

[De Waal, 2008] De Waal, F. B. (2008). Putting the altruism back into altruism: the evolution of empathy. *Annu. Rev. Psychol.*, 59:279–300.

[Delaherche et al., 2012] Delaherche, E., Chetouani, M., Mahdhaoui, A., Saint-Georges, C., Viaux, S., and Cohen, D. (2012). Interpersonal synchrony: A survey of evaluation methods across disciplines. *Affective Computing, IEEE Transactions on*, 3(3):349–365.

[Dennett, 1978] Dennett, D. C. (1978). Beliefs about beliefs [p&w, sr&b]. *Behavioral and Brain sciences*, 1(4):568–570.

[Devin and Alami, 2016] Devin, S. and Alami, R. (2016). An implemented theory of mind to improve human-robot shared plans execution. In *Human-Robot Interaction (HRI), 2016 11th ACM/IEEE International Conference on*, pages 319–326. IEEE.

[Dillenbourg, 1999] Dillenbourg, P. (1999). What do you mean by collaborative learning? *Collaborative-learning: Cognitive and Computational Approaches.*, pages 1–19.

[Dillenbourg et al., 1996] Dillenbourg, P., Traum, D., and Schneider, D. (1996). Grounding in multi-modal task-oriented collaboration. In *Proceedings of the European Conference on AI in Education*, pages 401–407.

[El Asri et al., 2016] El Asri, L., Piot, B., Geist, M., Laroche, R., and Pietquin, O. (2016). Score-based inverse reinforcement learning. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, pages 457–465.

[Fanelli et al., 2012] Fanelli, G., Gall, J., and Van Gool, L. (2012). Real time 3D head pose estimation: Recent achievements and future challenges. In *Communications Control and Signal Processing (ISCCSP), 2012 5th International Symposium on*, pages 1–4. IEEE.

[Finn et al., 2016] Finn, C., Christiano, P., Abbeel, P., and Levine, S. (2016). A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852.*

[Foerster et al., 2018] Foerster, J., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., and Mordatch, I. (2018). Learning with opponent-learning awareness. *Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems.*

[Foerster et al., 2017] Foerster, J. N., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., and Mordatch, I. (2017). Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326.*

[Fu et al., 2017] Fu, J., Luo, K., and Levine, S. (2017). Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248.*

[Fujimoto et al., 2018] Fujimoto, S., van Hoof, H., and Meger, D. (2018). Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477.*

[Gerbrandy et al., 1997] Gerbrandy, J. D. et al. (1997). *Dynamic epistemic logic.* Citeseer.

[Gmytrasiewicz et al., 1991] Gmytrasiewicz, P. J., Durfee, E. H., and Wehe, D. K. (1991). A decision-theoretic approach to coordinating multi-agent interactions. In *IJCAI*, volume 91, pages 63–68.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

[Gordon et al., 2016] Gordon, G., Spaulding, S., Westlund, J. K., Lee, J. J., Plummer, L., Martinez, M., Das, M., and Breazeal, C. (2016). Affective personalization of a social robot tutor for children's second language skills. In *Thirtieth AAAI Conference on Artificial Intelligence.*

[Gouaillier et al., 2008] Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., and Maisonnier, B. (2008). The NAO humanoid: a combination of performance and affordability. *CoRR.*

[Grizou et al., 2013] Grizou, J., Lopes, M., and Oudeyer, P.-Y. (2013). Robot learning simultaneously a task and how to interpret human instructions. In *2013 IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–8. IEEE.

[Haarnoja et al., 2017] Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. (2017). Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165.*

[Haarnoja et al., 2018] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290.*

[Halpern and Moses, 1990] Halpern, J. Y. and Moses, Y. (1990). Knowledge and common knowledge in a distributed environment. *Journal of the ACM (JACM)*, 37(3):549–587.

[Han et al., 2008] Han, J.-H., Jo, M.-H., Jones, V., and Jo, J.-H. (2008). Comparative study on the educational use of home robots for children. *Journal of Information Processing Systems*, 4(4):159–168.

[Harbers et al., 2009] Harbers, M., Bosch, K. v. d., and Meyer, J.-J. (2009). Modeling agents with a theory of mind. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 02*, pages 217–224. IEEE Computer Society.

[Haroush and Williams, 2015] Haroush, K. and Williams, Z. M. (2015). Neuronal prediction of opponent's behavior during cooperative social interchange in primates. *Cell*, 160(6):1233–1245.

[Harsanyi, 1967] Harsanyi, J. C. (1967). Games with incomplete information played by “bayesian” players, i–iii part i. the basic model. *Management science*, 14(3):159–182.

[Hintikka, 2005] Hintikka, J. (2005). *Knowledge and belief : an introduction to the logic of the two notions.* King’s College London Publications, London.

[Ho and Ermon, 2016] Ho, J. and Ermon, S. (2016). Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573.

[Holmqvist et al., 2011] Holmqvist, K., Nyström, M., Andersson, R., Dewhurst, R., Jarodzka, H., and Van de Weijer, J. (2011). *Eye tracking: A comprehensive guide to methods and measures.* Oxford University Press.

[Hood et al., 2015a] Hood, D., Lemaignan, S., and Dillenbourg, P. (2015a). When children teach a robot to write: An autonomous teachable humanoid which uses simulated handwriting. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, HRI ’15, pages 83–90, New York, NY, USA. ACM.

[Hood et al., 2015b] Hood, D., Lemaignan, S., and Dillenbourg, P. (2015b). When Children Teach a Robot to Write: An Autonomous Teachable Humanoid Which Uses Simulated Handwriting. In *Proceedings of the 2015 Human-Robot Interaction Conference.* To appear.

[Hood et al., 2015c] Hood, D., Lemaignan, S., and Dillenbourg, P. (2015c). When children teach a robot to write: An autonomous teachable humanoid which uses simulated handwriting. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, pages 83–90.

[Hoy et al., 2011] Hoy, M. M. P., Egan, M. Y., and Feder, K. P. (2011). A systematic review of interventions to improve handwriting. *Canadian Journal of Occupational Therapy*, 78(1):13–25.

[Hughes et al., 2018] Hughes, E., Leibo, J. Z., Phillips, M., Tuyls, K., Dueñez-Guzman, E., Castañeda, A. G., Dunning, I., Zhu, T., McKee, K., Koster, R., et al. (2018). Inequity aversion improves cooperation in intertemporal social dilemmas. *Advances in Neural Information Processing Systems.*

[Ibarz et al., 2018] Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S., and Amodei, D. (2018). Reward learning from human preferences and demonstrations in atari. In *Advances in Neural Information Processing Systems*, pages 8022–8034.

[Jacq et al., 2016a] Jacq, A., Johal, W., Dillenbourg, P., and Paiva, A. (2016a). Cognitive architecture for mutual modelling. *arXiv preprint arXiv:1602.06703.*

[Jacq et al., 2016b] Jacq, A., Lemaignan, S., Garcia, F., Dillenbourg, P., and Paiva, A. (2016b). Building successful long child-robot interactions in a learning context. In *Proceedings of the 2016 ACM/IEEE Human-Robot Interaction Conference.*

[Jacq et al., 2017] Jacq, A. D., Johal, W., Paiva, A., and Dillenbourg, P. (2017). Expressing motivations by facilitating other's inverse reinforcement learning. *Instituto Superiro Technico*, page 7.

[Jacq et al., 2018] Jacq, A. D., Magnan, J., Ferreira, M. J., Dillenbourg, P., and Paiva, A. (2018). Sensitivity to perceived mutual understanding in human-robot collaborations. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pages 2233–2235, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[Jermann and Nüssli, 2012] Jermann, P. and Nüssli, M.-A. (2012). Effects of sharing text selections on gaze cross-recurrence and interaction quality in a pair programming task. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pages 1125–1134.

[Jin et al., 2011] Jin, Z.-j., Qian, H., Chen, S.-y., and Zhu, M.-l. (2011). Convergence analysis of an incremental approach to online inverse reinforcement learning. *Journal of Zhejiang University SCIENCE C*, 12(1):17–24.

[Johal et al., 2016] Johal, W., Jacq, A., Paiva, A., and Dillenbourg, P. (2016). Child-robot spatial arrangement in a learning by teaching activity. In *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*, pages 533–538. IEEE.

[Jordan, 2002] Jordan, D. R. (2002). *Overcoming dyslexia in children, adolescents, and adults.* ERIC.

[Kajii and Morris, 1997] Kajii, A. and Morris, S. (1997). The robustness of equilibria to incomplete information. *Econometrica: Journal of the Econometric Society*, pages 1283–1309.

[Kanda et al., 2004] Kanda, T., Hirano, T., Eaton, D., and Ishiguro, H. (2004). Interactive robots as social partners and peer tutors for children: A field trial. *Human–Computer Interaction*, 19(1-2):61–84.

[Kazemi and Sullivan, 2014] Kazemi, V. and Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1867–1874.

[Kendon, 1990] Kendon, A. (1990). *Conducting interaction: Patterns of behavior in focused encounters*, volume 7. CUP Archive.

[Kennedy et al., 2015] Kennedy, J., Baxter, P., and Belpaeme, T. (2015). Head pose estimation is an inadequate replacement for eye gaze in child-robot interaction. In *Proceedings of the 10th ACM/IEEE International Conference on Human-Robot Interaction, Extended Abstracts*, HRI '15, pages 35–36. ACM.

[Kennedy et al., 2016] Kennedy, J., Baxter, P., Senft, E., and Belpaeme, T. (2016). Social robot tutoring for child second language learning. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, pages 231–238. IEEE Press.

[Kidd and Breazeal, 2008] Kidd, C. D. and Breazeal, C. (2008). Robots at home: Understanding long-term human-robot interaction. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3230–3235. IEEE.

[King, 2009] King, D. E. (2009). Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research*, 10:1755–1758.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kleiman-Weiner et al., 2016] Kleiman-Weiner, M., Ho, M. K., Austerweil, J. L., Littman, M. L., and Tenenbaum, J. B. (2016). Coordinate to cooperate or compete: abstract goals and joint intentions in social interaction. *Proceedings of Annual Conference of the Cognitive Science Society.*

[Klein et al., 2012] Klein, E., Geist, M., Piot, B., and Pietquin, O. (2012). Inverse reinforcement learning through structured classification. In *Advances in Neural Information Processing Systems*, pages 1007–1015.

[Klein et al., 2013] Klein, E., Piot, B., Geist, M., and Pietquin, O. (2013). A cascaded supervised learning approach to inverse reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 1–16. Springer.

[Kostrikov et al., 2018] Kostrikov, I., Agrawal, K. K., Dwibedi, D., Levine, S., and Tompson, J. (2018). Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In *International Conference on Representation Learning.*

[Kristoffersson et al., 2013] Kristoffersson, A., Eklundh, K. S., and Loutfi, A. (2013). Measuring the quality of interaction in mobile robotic telepresence: a pilot's perspective. *International Journal of Social Robotics*, 5(1):89–101.

[Landis and Koch, 1977] Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174.

[Lapeyre et al., 2014] Lapeyre, M., Rouanet, P., Grizou, J., Nguyen, S., Depraetre, F., Le Falher, A., and Oudeyer, P.-Y. (2014). Poppy project: open-source fabrication of 3d printed humanoid robot for science, education and art. In *Digital Intelligence 2014*, page 6.

[Lemaignan and Dillenbourg, 2015] Lemaignan, S. and Dillenbourg, P. (2015). Mutual modelling in robotics: Inspirations for the next steps. In *Proceedings of the 2015 ACM/IEEE Human-Robot Interaction Conference.*

[Lemaignan et al., 2016a] Lemaignan, S., Garcia, F., Jacq, A., and Dillenbourg, P. (2016a). From real-time attention assessment to “with-me-ness” in human-robot interaction. In *Proceedings of the 2016 ACM/IEEE Human-Robot Interaction Conference.*

[Lemaignan et al., 2016b] Lemaignan, S., Jacq, A., Hood, D., Garcia, F., Paiva, A., and Dillenbourg, P. (2016b). Learning by teaching a robot: The case of handwriting. *IEEE Robotics & Automation Magazine*, 23(2):56–66.

[Lerer and Peysakhovich, 2017] Lerer, A. and Peysakhovich, A. (2017). Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *arXiv preprint arXiv:1707.01068.*

[Letcher et al., 2018] Letcher, A., Foerster, J., Balduzzi, D., Rocktäschel, T., and Whiteson, S. (2018). Stable opponent shaping in differentiable games. *Proceedings of the International Conference on Learning Representations.*

[Leyzberg et al., 2014] Leyzberg, D., Spaulding, S., and Scassellati, B. (2014). Personalizing robot tutors to individuals' learning differences. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 423–430. ACM.

[Littman, 2001] Littman, M. L. (2001). Friend-or-foe q-learning in general-sum games. *Proceeding of the International Conference on Machine Learning.*

[Littman and Stone, 2005] Littman, M. L. and Stone, P. (2005). A polynomial-time nash equilibrium algorithm for repeated games. *Decision Support Systems.*

[Llorens et al., 2008] Llorens, D., Prat, F., Marzal, A., Vilar, J. M., Castro, M. J., Amengual, J.-C., Barrachina, S., Castellanos, A., Boquera, S. E., Gómez, J., et al. (2008). The ujipenchars database: a pen-based database of isolated handwritten characters. In *LREC.*

[Lubold et al., 2016] Lubold, N., Walker, E., and Pon-Barry, H. (2016). Effects of voice-adaptation and social dialogue on perceptions of a robotic learning companion. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 255–262. IEEE.

[Matsuzoe and Tanaka, 2012] Matsuzoe, S. and Tanaka, F. (2012). How smartly should robots behave?: Comparative investigation on the learning ability of a care-receiving robot. *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, pages 339–344.

[Mnih et al., 2016] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937.

[Mondada et al., 2017] Mondada, F., Bonani, M., Riedo, F., Briod, M., Pereyre, L., Rétornaz, P., and Magnenat, S. (2017). Bringing robotics to formal education: The thymio open-source hardware robot. *IEEE Robotics & Automation Magazine*, 24(1):77–85.

[Moosaei et al., 2017] Moosaei, M., Das, S. K., Popa, D. O., and Riek, L. D. (2017). Using facially expressive robots to calibrate clinical pain perception. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 32–41. ACM.

[Movellan et al., 2009] Movellan, J., Eckhardt, M., Virnes, M., and Rodriguez, A. (2009). Sociable robot improves toddler vocabulary skills. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 307–308. ACM.

[Munoz de Cote and Littman, 2008] Munoz de Cote, E. and Littman, M. L. (2008). A polynomial-time Nash equilibrium algorithm for repeated stochastic games. *Proceedings of the Conference on Uncertainty in Artificial Intelligence.*

[Nachum et al., 2017] Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. (2017). Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2775–2785.

[Nash, 1951] Nash, J. (1951). Non-cooperative games. *Annals of mathematics*, pages 286–295.

[Neu et al., 2017] Neu, G., Jonsson, A., and Gómez, V. (2017). A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798.*

[Newell et al., 1982] Newell, A. et al. (1982). The knowledge level. *Artificial intelligence*, 18(1):87–127.

[Ng et al., 1999] Ng, A. Y., Harada, D., and Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287.

[Ng et al., 2000] Ng, A. Y., Russell, S. J., et al. (2000). Algorithms for inverse reinforcement learning. In *Icml*, pages 663–670.

[Nikolaidis et al., 2016a] Nikolaidis, S., Dragan, A., and Srinivasa, S. (2016a). based legibility optimization. In *Human-Robot Interaction (HRI), 2016 11th ACM/IEEE International Conference on*, pages 271–278. IEEE.

[Nikolaidis et al., 2016b] Nikolaidis, S., Dragan, A., and Srinivasa, S. (2016b). Viewpoint-based legibility optimization. In *Proceedings of the 2016 ACM/IEEE Human-Robot Interaction Conference.*

[Nikolaidis et al., 2016c] Nikolaidis, S., Dragan, A., and Srinivasa, S. (2016c). Viewpoint-based legibility optimization. In *Human-Robot Interaction (HRI), 2016 11th ACM/IEEE International Conference on*, pages 271–278. IEEE.

[Nikolaidis et al., 2017] Nikolaidis, S., Hsu, D., and Srinivasa, S. (2017). Human-robot mutual adaptation in collaborative tasks: Models and experiments. *The International Journal of Robotics Research*, 36(5-7):618–634.

[Nikolaidis et al., 2015] Nikolaidis, S., Ramakrishnan, R., Gu, K., and Shah, J. (2015). Efficient model learning from joint-action demonstrations for human-robot collaborative tasks. In *Proceedings of the tenth annual ACM/IEEE international conference on human-robot interaction*, pages 189–196. ACM.

[Ozgur et al., 2017] Ozgur, A., Lemaignan, S., Johal, W., Beltran, M., Briod, M., Pereyre, L., Mondada, F., and Dillenbourg, P. (2017). Cellulo: Versatile handheld robots for education. *HRI '17: ACM/IEEE International Conference on Human-Robot Interaction Proceedings*, pages 119–127. Best Human-Robot Interaction Design paper award.

[Pérolat et al., 2017] Pérolat, J., Leibo, J. Z., Zambaldi, V., Beattie, C., Tuyls, K., and Graepel, T. (2017). A multi-agent reinforcement learning model of common-pool resource appropriation. *Advances in Neural Information Processing Systems.*

[Peters et al., 2010] Peters, C., Asteriadis, S., and Karpouzis, K. (2010). Investigating shared attention with a virtual agent using a gaze-based interface. *Journal on Multimodal User Interfaces*, 3(1-2):119–130.

[Peysakhovich and Lerer, 2018] Peysakhovich, A. and Lerer, A. (2018). Consequentialist conditional cooperation in social dilemmas with imperfect information. *Proceedings of the International Conference on Learning Representations.*

[Premack and Woodruff, 1978] Premack, D. and Woodruff, G. (1978). Does the chimpanzee have a theory of mind. *Behavioral and Brain sciences*, 1(4):515–526.

[Pylyshyn, 1978] Pylyshyn, Z. W. (1978). When is attribution of beliefs justified?[p&w]. *Behavioral and brain sciences*, 1(4):592–593.

[Pynadath and Marsella, 2005] Pynadath, D. V. and Marsella, S. C. (2005). Psychsim: Modeling theory of mind with decision-theoretic agents. In *IJCAI*, volume 5, pages 1181–1186.

[Qureshi et al., 2019] Qureshi, A. H., Boots, B., and Yip, M. C. (2019). Adversarial imitation via variational inverse reinforcement learning. In *International Conference on Learning Representations.*

[Raileanu et al., 2018] Raileanu, R., Denton, E., Szlam, A., and Fergus, R. (2018). Modeling others using oneself in multi-agent reinforcement learning. *arXiv preprint arXiv:1802.09640.*

[Rao, 1995] Rao, A. S. (1995). Decision procedures for prepositional linear-time belief-desire-intention logics. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 33–48. Springer.

[Reichardt, 1978] Reichardt, J. (1978). *Robots: Fact, fiction, and prediction.* Thames and Hudson London.

[Rohrbeck et al., 2003] Rohrbeck, C. A., Ginsburg-Block, M. D., Fantuzzo, J. W., and Miller, T. R. (2003). Peer-assisted learning interventions with elementary school students: A meta-analytic review. *Journal of Educational Psychology*, 95(2):240–257.

[Ruhland et al., 2015] Ruhland, K., Peters, C., Andrist, S., Badler, J., Badler, N., Gleicher, M., Mutlu, B., and McDonnell, R. (2015). A Review of Eye Gaze in Virtual Agents, Social Robotics and HCI: Behaviour Generation, User Interaction and Perception. In *Computer Graphics Forum*. Wiley Online Library.

[Sandholm and Crites, 1996] Sandholm, T. W. and Crites, R. H. (1996). Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems*, 37(1):147–166.

[Sangin et al., 2007] Sangin, M., Nova, N., Molinari, G., and Dillenbourg, P. (2007). Partner modeling is mutual. In *Proceedings of the 8th iternational conference on Computer Supported Collaborative Learning*, pages 625–632. International Society of the Learning Sciences.

[Schulman et al., 2017] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

[Sequeira et al., 2014] Sequeira, P., Melo, F. S., and Paiva, A. (2014). The influence of social display in competitive multiagent learning. *4th International Conference on Development and Learning and on Epigenetic Robotics*, pages 64–69.

[Shapley, 1953] Shapley, L. S. (1953). Stochastic games. *Proceedings of the national academy of sciences.*

[Sharma et al., 2014] Sharma, K., Jermann, P., and Dillenbourg, P. (2014). "with-me-ness": A gaze-measure for students' attention in moocs. In *International conference of the learning sciences.*

[Singh et al., 2010] Singh, S., Lewis, R. L., Barto, A. G., and Sorg, J. (2010). Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development*, 2(2):70–82.

[Sisbot et al., 2011] Sisbot, E., Ros, R., and Alami, R. (2011). Situation Assessment for Human-Robot Interaction. In *20th IEEE International Symposium in Robot and Human Interactive Communication.*

[Smallwood and Sondik, 1973] Smallwood, R. D. and Sondik, E. J. (1973). The optimal control of partially observable markov processes over a finite horizon. *Operations research*, 21(5):1071–1088.

[Stiefelhagen, 2002] Stiefelhagen, R. (2002). Tracking focus of attention in meetings. In *IEEE International Conference on Multimodal Interfaces*, page 273.

[Sutton and Barto, 1998] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT press Cambridge.

[Suzuki et al., 2015] Suzuki, S., Adachi, R., Dunne, S., Bossaerts, P., and O'Doherty, J. P. (2015). Neural mechanisms underlying human consensus decision-making. *Neuron*, 86(2):591–602.

[Syed and Schapire, 2008] Syed, U. and Schapire, R. E. (2008). A game-theoretic approach to apprenticeship learning. In *Advances in neural information processing systems*, pages 1449–1456.

[Takayama and Pantofaru, 2009] Takayama, L. and Pantofaru, C. (2009). Influences on proxemic behaviors in human-robot interaction. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pages 5495–5502.

[Tanaka et al., 2007] Tanaka, F., Cicourel, A., and Movellan, J. R. (2007). Socialization between toddlers and robots at an early childhood education center. *Proceedings of the National Academy of Sciences*, 104(46):17954–17958.

[Tanaka and Kimura, 2009] Tanaka, F. and Kimura, T. (2009). The use of robots in early education: a scenario based on ethical consideration. In *RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication*, pages 558–560. IEEE.

[Tanaka and Matsuzoe, 2012] Tanaka, F. and Matsuzoe, S. (2012). Children teach a care-receiving robot to promote their learning: Field experiments in a classroom for vocabulary learning. *Journal of Human-Robot Interaction*, 1(1).

[Thomas et al., 1995] Thomas, F., Johnston, O., and Thomas, F. (1995). *The illusion of life: Disney animation*. Hyperion New York.

[van Breemen et al., 2005] van Breemen, A., Yan, X., and Meerbeek, B. (2005). icat: an animated user-interface robot with personality. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 143–144. ACM.

[Van Ditmarsch and Labuschagne, 2007] Van Ditmarsch, H. and Labuschagne, W. (2007). My beliefs about your beliefs: a case study in theory of mind and epistemic logic. *Synthese*, 155(2):191–209.

[Verbrugge, 2009] Verbrugge, R. (2009). Logic and social cognition. *Journal of Philosophical Logic*, 38(6):649–680.

[Walker et al., 1980] Walker, H., Hall, W., and Hurst, J. (1980). *Clinical Methods: The History, Physical, and Laboratory Examinations.* Clinical Methods: The History, Physical, and Laboratory Examinations. Butterworth.

[Warnier et al., 2012] Warnier, M., Guitton, J., Lemaignan, S., and Alami, R. (2012). When the robot puts itself in your shoes. managing and exploiting human and robot beliefs. In *Proceedings of the 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 948–954.

[Watkins and Dayan, 1992] Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4):279–292.

[Watkins, 1989] Watkins, C. J. C. H. (1989). *Learning from delayed rewards.* PhD thesis, King's College, Cambridge.

[Yadollahi et al., 2018] Yadollahi, E., Johal, W., Paiva, A., and Dillenbourg, P. (2018). When deictic gestures in a robot can harm child-robot collaboration. In *Proceedings of the 17th ACM Conference on Interaction Design and Children*, pages 195–206. ACM.

[Yarbus, 1967] Yarbus, A. L. (1967). *Eye movements during perception of complex objects.* Springer.

[You et al., 2006] You, Z.-J., Shen, C.-Y., Chang, C.-W., Liu, B.-J., and Chen, G.-D. (2006). A robot as a teaching assistant in an english class. In *Sixth IEEE International Conference on Advanced Learning Technologies (ICALT'06)*, pages 87–91. IEEE.

[Ziebart et al., 2008] Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA.

[Özgür, 2018] Özgür, A. (2018). Cellulo: Tangible haptic swarm robots for learning. *EPFL*, page 240.

Alexis Jacq

alexis-jacq.github.io   alexisjacq@google.com   github.com/alexis-jacq

Education

Ecole Polytechnique Federale de Lausanne & Instituto Superior Tecnico — **Lisbon & Lausanne**
Ph.D: Mutual Understanding in Educational Human-Agent Collaborations — *2015 - 2019*

Ecole Normal Superieure (Cachan) — **Cachan, France**
Master of mathematics, spe Mathematics for Vision and Learning — *2013-2014*

Joseph Fourier University — **Grenoble, France**
Bachelor of mathematics — *2009-2012*

Research Scientist

Google Research, Brain Team — **Paris**
2019 - now

Selected Publications

Learning from a Learner
A. Jacq, M. Gueist, A. Paiva, O. Pietquin — *ICML 2019*

Sensitivity To Perceived Mutual Understanding In Human-Robot Collaborations
A. Jacq, J. Magnan, M. Ferreira, P. Dillenbourg, A. Paiva — *AAMAS 2018*

Building successful long child-robot interactions in a learning context
A. Jacq, S. Lemaignan, F. Garcia, P. Dillenbourg, A. Paiva — *HRI 2016*

From real-time attention assessment to with-me-ness in human-robot interaction
S. Lemaignan, F. Garcia, A. Jacq, P. Dillenbourg — *(Best paper award) HRI 2016*

Teaching

EPFL..

C programming — *Fall 2017*
Analyse IV — *Spring 2016*
Analyse I — *Spring 2015*
Introduction to Visual Computing — *(Best assistant award) Spring 2015*

Ecole Normal Superieure (Paris)..

Learning Hidden Markov Models — *(GT Math-Bio work group) 2013*

Languages

- **Typed**: Python (mother tongue), C/C++, R, Java, HTML/CSS, Javascript
- **Spoken**: Fluent in French and English, A1 level in Portuguese

Other interests

- **Sports**: surfing, climbing, dance improvisation, running
- **Arts**: painting (oil/watercolour), sculpting (clay/wood)