Department of Electrical Engineering and Automation

Teacher-Learner Interaction for Robot Active Learning

Mattia Racca

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 145/2020

Teacher-Learner Interaction for Robot Active Learning

Mattia Racca

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Electrical Engineering, remote connection at aalto.zoom.us/j/61397232666, on the 30th of October 2020 at 14:00 EET.

Aalto University
School of Electrical Engineering
Department of Electrical Engineering and Automation
Intelligent Robotics Group

Supervising professor
Professor Ville Kyrki, Aalto University, Finland

Preliminary examiners
Professor Ross Knepper, Cornell University (former), United States of America
Professor Elin Anna Topp, Lund University, Sweden

Opponent
Professor Tony Belpaeme, Ghent University, Belgium, and University of Plymouth, United Kingdom

Aalto University publication series
DOCTORAL DISSERTATIONS 145/2020

ISBN 978-952-64-0054-9 (printed)
ISBN 978-952-64-0055-6 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-64-0055-6

Images: Cover Illustration by Ada Peiretti

Unigrafia Oy
Helsinki 2020

Finland

Printed matter
4041-0619

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Mattia Racca

Name of the doctoral dissertation
Teacher-Learner Interaction for Robot Active Learning

Publisher School of Electrical Engineering

Unit Department of Electrical Engineering and Automation

Series Aalto University publication series DOCTORAL DISSERTATIONS 145/2020

Field of research Automation, Systems and Control Engineering

Manuscript submitted 16 June 2020 **Date of the defence** 30 October 2020

Permission for public defence granted (date) 4 September 2020 **Language** English

☐ **Monograph** ☒ **Article dissertation** ☐ **Essay dissertation**

Abstract

Robots are being adopted in an increasing number of new application areas, such as health care, logistics, and domestic services. Far from the structured environments of industrial settings, interaction between robots and humans will often become necessary and potentially beneficial. Simultaneously, the target audience of robots will grow to include users who lack the technical skills needed to program robots in the traditional manner. This dissertation proposes interactive learning methods based on Active Learning (AL) and Learning from Demonstration (LfD) that allow robots to learn by interacting with humans-in-the-loop.

First, LfD, a learning paradigm that allows the user to program robots by providing examples of the desired behaviour, is adopted for the programming of in-contact tasks. To handle the variability of demonstrations collected through kinesthetic teaching, a probabilistic approach to the encoding of force profiles is proposed. Robot learning is then analysed as a collaborative task, showing how LfD often requires the human teacher to be an expert not only at the task in question but also at teaching a robot.To address the consequences of this rarely met requirement, AL, a learning paradigm that allows robots to participate actively in the teaching process by making queries to their teachers, is proposed for two applications: in combination with LfD for the learning of temporal task models, and as an aid for the tuning of robot programs in an End User Programming (EUP) scenario. The proposed AL approaches and their queries are designed taking into account the interaction between the robot and its human teacher.

Investigating the interaction aspect of the proposed AL approaches revealed how their efficient queries may not always be optimal when human teachers are not considered ever-present, infallible sources of information. To investigate this issue, a AL approach that selects queries by taking into account the effort needed for the teacher to answer them is proposed and compared to traditional AL strategies, showing how these strategies impact the teacher's error rates, response times, and workload. Studying the interaction aspect further revealed the importance of transparency and the need for tools that expose robots' decisions to their users. A model-agnostic method that generates natural language explanations for robot policies is therefore presented, along with a study of the effect of its explanations on the user's understanding of the robot's policy. In summary, this dissertation investigates robot learning methods, emphasizing how their design should account for the interaction aspect of the training process.

Keywords Robotics, Human-Robot Interaction, Active Learning, Learning from Demonstration

ISBN (printed) 978-952-64-0054-9 **ISBN (pdf)** 978-952-64-0055-6

ISSN (printed) 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2020

Pages 152 **urn** http://urn.fi/URN:ISBN:978-952-64-0055-6

Preface

This dissertation is the result of four years of research in the Intelligent Robotics group at Aalto University, including a research visit to the Human-Centered Robotics lab at the University of Washington. My work has been financially supported by the Aalto ELEC Doctoral School, the Academy of Finland through the ROSE project, and the Ernst Wirtzen fund, to whom I am grateful for the funding.

First and foremost, I want to deeply thank Professor Ville Kyrki for his guidance during my doctoral endeavour. As my supervisor, he helped me to never miss a chance to become a better researcher, supporting my ideas and plans, and giving clear guidance when direly needed. I truly appreciated his thoughtfulness about the well-being of the research group, always striving to create a pleasant working environment for everybody. I am also extremely thankful to Professor Maya Cakmak, who welcomed me in her lab at the University of Washington for an incredible research visit. Her insight and contagious passion for research motivated me to further improve as a Human-Robot Interaction (HRI) researcher. I am honoured to have our work included in this dissertation.

I want to thank my pre-examiners, Professor Elin Anna Topp and Professor Ross Knepper, whose insightful comments helped me refine this dissertation to its current shape. Many thanks to the co-authors of the articles included in this dissertation: to Professor Antti Oulasvirta, for the insights from the Human-Computer Interaction community; to Professor Joni Pajarinen and Alberto Montebelli, for helping me turn my Master's thesis into my first publication; to Oliver Struckmeier, for all the hard work and for keeping up with my, at times excessive, meticulousness. I extend this gratitude to the HRI community as a whole: a welcoming community of passionate and brilliant researchers that pushed me to the highest standards, and provided great feedback in the process.

Thanks to all the colleagues I had the pleasure to work with, both at Aalto and at the University of Washington. Special thanks to Roel Pieters, for guiding my first steps both in academia and in the climbing hall; to Markku Suomalainen, for the early morning coffee breaks in the deserted department and the adventurous conference hikes; to Francesco Verdoja, for the great teamwork, all the discussions about work and life, and the many hobbies we shared; to all

the office mates I had over the years, in particular to Jens Lundell and David Blanco Mulero for all the serious and lighthearted conversations.

As the proverb goes, “all work and no play makes Jack a dull boy”. I therefore want to thank my closest friends that kept my dullness in check while I was pursuing my PhD. Thanks to Iryna, Ada, and Francesco, for the adventure-fuelled afternoons and the amazing mökki vacations; to Filippo, for all the memorable moments both in Helsinki and Turin, the hikes and climbs, the memes, and the hours spent playing a 20 years-old video game; to Camilla, for the chatty bouldering sessions, discussing the joys and sorrows of life as PhD students; to Wendy and Luke, for all the fun activities in Seattle; to Gianluca, Marco, Teo, and Vittorio, for always finding the time to meet when I was back home and making it look like I never left – grazie fieuj!

Finally, I want to thank my parents and my sister Angelica for all the support during these years. Thanks also to my grandparents, who would have definitively preferred to have me back at home rather than around the world, but never failed to remotely cheer me up. Last but certainly not least, many thanks to my wonderful Ulrike, for making every day of our journey together better than the previous one. Love you!

Helsinki, September 14, 2020,

Mattia Racca

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Mattia Racca, Joni Pajarinen, Alberto Montebelli and Ville Kyrki. Learning In-Contact Control Strategies from Demonstration. In *International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, South Korea. pp. 688–695, September 2016.

II Mattia Racca and Ville Kyrki. Active Robot Learning for Temporal Task Models. In *International Conference on Human-Robot Interaction (HRI)*, Chicago, USA. pp. 123–131, February 2018.

III Mattia Racca, Antti Oulasvirta and Ville Kyrki. Teacher-Aware Active Robot Learning. In *International Conference on Human-Robot Interaction (HRI)*, Daegu, South Korea. pp. 335–343, March 2019.

IV Mattia Racca, Ville Kyrki and Maya Cakmak. Interactive Tuning of Robot Program Parameters via Expected Divergence Maximization. In *International Conference on Human-Robot Interaction (HRI)*, Cambridge, UK. pp. 629–638, March 2020.

V Oliver Struckmeier, Mattia Racca and Ville Kyrki. Autonomous Generation of Robust and Focused Explanations for Robot Policies. In *International Conference on Robot and Human Interactive Communication (RO-MAN)*, New Delhi, India. pp. 1–8, October 2019.

Author's Contribution

Publication I: "Learning In-Contact Control Strategies from Demonstration"

This publication is an extension of the Master's thesis of Mattia Racca, the author of this dissertation. The authors defined the learning problem together and devised the proposed Learning from Demonstration (LfD) method. Mattia Racca implemented the method, conducted the experiments and analysed the results. Mattia Racca and Joni Pajarinen were responsible for the writing of the publication.

Publication II: "Active Robot Learning for Temporal Task Models"

The authors developed together the Active Learning (AL) framework and defined the research questions. Mattia Racca implemented the learning framework, designed and conducted the user study, and analysed the results. Mattia Racca was responsible for writing the publication.

Publication III: "Teacher-Aware Active Robot Learning"

The authors defined together the problem and the research questions. Mattia Racca implemented the learning framework. The user study was designed by all authors and conducted by Mattia Racca. Mattia Racca analysed the results and was the main author of the publication.

Publication IV: "Interactive Tuning of Robot Program Parameters via Expected Divergence Maximization"

This publication is the result of Mattia Racca's internship at the University of Washington, under the supervision of professor Maya Cakmak. Mattia Racca and Maya Cakmak defined the tuning problem together. The AL solution was devised through the combined effort of all authors. Mattia Racca implemented the proposed tuning framework and the End-User Programming (EUP) interface for the Panda robot arm. The experiments were designed by all authors and conducted by Mattia Racca, together with the analysis of the results. Mattia Racca and Maya Cakmak were responsible for writing the publication.

Publication V: "Autonomous Generation of Robust and Focused Explanations for Robot Policies"

This publication is an extension of the Master's thesis of its first author, Oliver Struckmeier. Mattia Racca instructed Oliver Struckmeier together with professor Ville Kyrki. The explanation framework was designed by all authors. Oliver Struckmeier was responsible for the implementation and testing of the framework, regularly instructed by Mattia Racca. Mattia Racca and Oliver Struckmeier were responsible for the design of the user study and the writing of the publication.

Language check

The language of my dissertation has been checked by Matthew Billington. I have personally examined and accepted/rejected the results of the language check one by one. This has not affected the scientific content of my dissertation.

List of Acronyms

ACT-R Adaptive Control of Thought–Rational

AL Active Learning

AwA2 Animals with Attributes 2

BIC Bayesian Information Criterion

CIC Cartesian Impedance Controller

DMP Dynamic Movement Primitive

DoF Degree of Freedom

DQ Disambiguation Query

DSL Domain Specific Language

EM Expectation-Maximization

EUP End-User Programming

FQ Frequency Query

GMM Gaussian Mixture Model

GMR Gaussian Mixture Regression

GPR Gaussian Process Regression

GUI Graphical User Interface

HMM Hidden Markov Model

HRI Human-Robot Interaction

HSMM Hidden semi-Markov Model

IC Impedance Control

IL Imitation Learning

IML Interpretable Machine Learning

IRL Inverse Reinforcement Learning

LfD Learning from Demonstration

LIME Local Interpretable Model-agnostic Explanations

LWR Locally Weighted Regression

MC Markov Chain

MDP Markov Decision Process

ML Machine Learning

NASA TLX NASA Task Load Index

PbD Programming by Demonstration

PL Passive Learning

POMDP Partially Observable Markov Decision Process

ProMP Probabilistic Movement Primitive

RL Reinforcement Learning

ToM Theory of Mind

XAI Explainable Artificial Intelligence

Symbols

Notation:

$\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \ldots \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \ldots$	Vector variables
$a, b, c, \ldots$	Scalar variables
$A, B, C, \ldots$	Matrix variables
$\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots$	Sets

Symbols:

$\mathcal{A}$	Action set
b	Query budget
$\text{Ber}(\cdot\|\theta)$	Bernoulli distribution
$\text{Beta}(\cdot\|\alpha, \beta)$	Beta distribution
c_{msr}	Consistency measure
$\mathcal{C}$	Category set
$\text{Cat}(\cdot\|\boldsymbol{\alpha})$	Categorical distribution
$\boldsymbol{d}_c$	Damping parameter
d_i	Label for the i-th dimension of the state space
d_{msr}	Describability measure
$\mathcal{D}$	Demonstration set
$\text{Dir}(\cdot\|\boldsymbol{\alpha})$	Dirichlet distribution
$\mathcal{E}$	Entity set
$\hat{\boldsymbol{f}}$	Feasible parameter range

$\boldsymbol{f}_{\text{cmd}}$	Commanded force
$f_{\max}(x)$	Distribution of the sample maximum
$f_{\min}(x)$	Distribution of the sample minimum
$\boldsymbol{f}_t$	Force reading at the end-effector at time t
$\mathcal{F}$	Frequency adverb set
$h_{i,t}$	Belief over HSMM state i at time t
$\mathbb{H}(p)$	Entropy of a distribution p
$\boldsymbol{J}$	Jacobian
$\mathbb{JS}(p,\ q)$	Jensen-Shannon divergence of distributions p and q
$\boldsymbol{k}_c$	Stiffness parameter
$\mathbb{KL}(p,\ q)$	Kullback-Leibler divergence of distributions p and q
$\hat{l}$	Lower-bound of the feasible parameter range $\hat{\boldsymbol{f}}$
N	Number of states in a Markov Model
$\mathrm{N}(\cdot\|\mu,\sigma)$	Normal distribution (univariate)
$\mathrm{N}(\cdot\|\boldsymbol{\mu},\Sigma)$	Normal distribution (multivariate)
$\boldsymbol{q}_t$	Orientation of the end-effector at time t
q^*	Selected query
$\langle q, r\rangle$	Query-answer pair
$\mathcal{Q}$	Query pool
r_{msr}	Relevance measure
r^*	Obtained answer
s_{msr}	Stability measure
$\mathbb{S}^k$	Simplex, given by $\left\{(s_0,\ldots,s_k)\in\mathbb{R}^{k+1}\middle\vert\sum_{i=0}^{k}s_i=1\wedge s_i\geq 0,\forall i\right\}$
T	Transition matrix of Markov Models
$\mathcal{T}$	Entity-Category tree
$\hat{u}$	Upper-bound of the feasible parameter range $\hat{\boldsymbol{f}}$
v	Number of samples for local measures
$\boldsymbol{v}_t$	Translational velocity of the end-effector at time t

w	Number of samples for global measures
$w_{c,e}$	Relevance of category c for entity e
$\boldsymbol{x}_{\text{cmd}}$	Commanded end-effector pose
$\boldsymbol{x}_{\text{msr}}$	Measured end-effector pose
$\boldsymbol{x}_t$	Position of the end-effector at time t
$\boldsymbol{X}_t$	Observation at time t
$\boldsymbol{X}_t^I$	Policy input at time t
$\boldsymbol{X}_t^O$	Policy output at time t
γ_i^*	Descriptor of dimension d_i for the current state $\boldsymbol{x}$
δ	Scale parameter for the **Memory** strategy
η	Threshold for local measures
$\boldsymbol{\pi}$	Starting probabilities of Markov Models
ρ	Sampling radius for local measures
σ	Trade-off parameter for the **Hybrid** strategy
τ	Duration of control time-step
ϕ	Cut-off parameter for **Threshold** strategy
$\boldsymbol{\omega}_t$	Rotational velocity of the end-effector at time t

1. Introduction

Humankind's interest in building machines that act autonomously is at least 2000 years old. Long before robots were even called *robots* [1], inventors such as Heron of Alexandria, Ismail al-Jazari, and Leonardo da Vinci created marvellous automata [2]. Today, robots are becoming increasingly present in our society. Firmly established as key components of modern manufacturing processes, robots are now boldly taking their first steps in application areas like health care, logistics, agriculture, entertainment and domestic services. As robots conquer highly unstructured environments where interaction with humans becomes inevitable, two characteristics become paramount to their widespread adoption: *programmability* and *adaptability*.

Programmability is the ability of a robot to accept instructions from its user and alter its behaviour accordingly [3]. Programmability represents the main advantage of robots over standard automation: being programmable allows a robot to perform a variety of tasks, increasing its usefulness and cost effectiveness. Most modern robots feature programming interfaces of some sort, with research continuously improving and innovating such interfaces.

Adaptability refers instead to the ability of a robot to alter its behaviour autonomously *after* it has been programmed, for instance while interacting with the environment. As it is unrealistic to program a robot for every situation it will ever encounter, making robots adaptable has become a major goal for the robotics community. The challenge of providing robots with adaptability is often approached as a Machine Learning (ML) problem, with the goal of allowing robots to *learn* in order to adapt to new situations, environments, and their users' preferences.

When deployed in everyday environments, robots will be more likely to interact with humans. Hence, this dissertation argues that programmability and adaptability of robots can be achieved by leveraging their interaction with people. Joining a corpus of research adopting ML techniques to solve robotics problems, this dissertation presents learning techniques, based on Learning from Demonstration (LfD) and Active Learning (AL), that leverage the presence of the human-in-the-loop in intuitive ways.

Having robots learn from humans nevertheless presents unique challenges.

With the deployment of robots beyond industrial settings, their target audience will grow to include a wide variety of users who, while being experts in their professional field, may lack the technical skills to understand how robots perceive, act, and learn. Furthermore, people's time, patience and attention are limited resources that require careful managing during the interaction with robots. This dissertation therefore pays particular attention to the Human-Robot Interaction (HRI) aspect of robot learning, investigating how the aforementioned learning methods influence and are influenced by the interactive nature of the training process.

1.1 Motivation and Contributions

The overarching goal of this dissertation is to provide programmability and adaptability to service robots that interact with novice users.[1] This endeavour is motivated by the fact that (i) programming robots requires a complex set of skills from the fields of computer science and engineering, making unreasonable to assume that every user possesses such a skill set, and that (ii) it is unfeasible, from an engineering standpoint, to program robots to be able to face every situation in dynamic and unstructured environments. Instead, novice users should be able to program their robots and customize their behaviour in natural and intuitive ways [4]. In other words, novices should be able to program robots by other means than writing lines of code, such as providing examples of desired behaviours, or specifying a desired goal without the need to indicate the steps required to achieve it.

The following sections introduce the publications included in the dissertation, presenting the core ideas and highlighting the contribution of each work.

1.1.1 Robot Learning from Demonstration

Learning from Demonstration (LfD), also known as Programming by Demonstration (PbD) and Imitation Learning (IL), is a way of programming robots by providing *demonstrations*, *i.e.*, examples of the desired behaviour or skill [5]. Inspired by the imitation capabilities observed in humans and animals alike [6], the paradigm requires a teacher to provide a set of demonstrations of the target skill. LfD approaches vary widely in how the demonstrations are collected, how many demonstrations are needed, what models are used to encode the learned skill, and how the skill is finally reproduced [7]. Nevertheless, the main strength of LfD lies in the intuitiveness of providing demonstrations, unlocking robot programmability for novice users.

LfD is beneficial also for expert users. Giving multiple demonstrations of a skill allows the building of models that generalize to new situations, achieving

[1] In this dissertation, robot users are referred to as *novices* or *non-experts* if they lack the technical skills or education background to program robots in a traditional manner.

varying degrees of adaptability. Furthermore, some skills can be extremely hard to encode in declarative terms but are easily demonstrated by an expert provided with an intuitive demonstration interface. This is especially true for skills involving complex velocity and acceleration profiles (like table tennis strokes [8]) or the fine exertion of forces on the environment (*e.g.*, tying a knot [9]).

In Publication I, we exploited this last feature of LfD, targeting the learning of *in-contact tasks*, *i.e.*, tasks that require an accurate exertion of forces in order to succeed. The main contributions of Publication I are

1. a statistical approach to LfD, using a combination of Hidden semi-Markov Models (HSMMs) and Gaussian Mixture Regression (GMR) to model both the spatio-temporal information of the skill and the relevant force profiles from human kinesthetic demonstrations, and

2. a technique that modulates the stiffness of a Cartesian Impedance Controller (CIC) during the reproduction of the task based on the learned HSMM, in order to correctly execute both the in-contact and the free-space portions of the taught skill.

1.1.2 Teaching as a Collaborative Task

When people engage in a collaborative task, they create over time *common ground*, *i.e.*, the knowledge, beliefs, and suppositions they believe they share about the task [10]. Analysing human teaching from a Theory of Mind (ToM) perspective reveals it to be a collaborative task [11], where the teacher must understand the learners' mental models (knowledge, beliefs, desires) to intentionally recognize gaps in their knowledge and act appropriately to reduce them. At the same time, the learner should support the teacher's task by exposing such information.

When humans teach robots, common ground can be hard to find due to the different nature of the agents involved: robots and humans may not, in fact, share the same representation of the concept to be taught or the surrounding environment. Moreover, novice users may misunderstand how robots learn and consequently apply human teaching techniques that are unlikely to be optimal for arbitrary ML agents [12]. These discrepancies can prevent the creation of common ground, lowering the effectiveness of the teaching process and potentially resulting in mistrust or over-reliance [13].

Learning techniques based on the collection of demonstrations often assume that the human teacher is able to provide *informative demonstrations*, *i.e.*, demonstrations that allow the ML agent to learn effectively. Given the aforementioned discrepancies between the agents involved, the assumption that the human teacher will be an expert not only of the target skill but also at teaching a robot is rather unrealistic [14, 15]. The issue is further exacerbated by the interaction model of traditional LfD techniques, where the teacher is solely

responsible for providing the data required for the training. In the unfortunate case that the teacher is unable to provide informative demonstrations, traditional LfD approaches are forced merely to attempt to cope with such poor demonstrations.

One solution to this issue is to have the robot behave less passively and participate in the training process, sharing the responsibility with the human teacher. In other words, we argue that the teaching should be treated as a *collaborative task* between the user and the robot. To achieve such collaboration, the robot can influence the training, for example by selecting which human demonstrations to learn from and discarding others [16] or using demonstrations only to kick-start learning techniques based on self-exploration, as Reinforcement Learning (RL) [17] and Inverse Reinforcement Learning (IRL) [18]. This dissertation explores the idea of robots that can make requests to their teachers to address their current knowledge gaps.

1.1.3 Robot Active Learning

Recognizing gaps in the learner's knowledge is a key skill of successful teachers. As mentioned before, in the case of learning robots, human teachers may, however, lack such skill, hindering the training process. One possible solution is to equip robots with ways to report their knowledge gaps during training. This can be achieved, for example, by showing failure cases [19, 20] or by exposing uncertainties about the current predictions [21] in order to influence the user's future teaching.

Going one step forward, learning robots could inspect their current knowledge during training and actively request information to address knowledge gaps. These active requests, commonly referred to as *queries*, represent the core idea behind AL, a ML paradigm where the agent chooses what to learn from [22, 23, 24], steering the training process to cover its current knowledge gaps. Instead of waiting for labelled data to become available (*e.g.*, chosen by humans) or requesting labels for randomly selected samples, AL agents can query *informative samples* to learn more efficiently, *i.e.*, with less labelled data [22, 25].

With the goal of making robots rely less on the possibly suboptimal demonstrations of human teachers, in Publication II we developed a mixed AL-LfD approach for the learning of temporal task models. The main contributions of Publication II are

1. a Bayesian learning approach combining demonstrations and robot-initiated queries to fit the parameters of a Markov Chain (MC) modelling a temporal task,

2. the design and integration in the training process of queries that are user friendly yet informative for the learning task at hand, and

3. a study of the HRI aspects of the proposed AL-LfD technique, focusing on

ease of teaching, transparency of the training process, and user perception of robot queries.

1.1.4 Active Learning-aided End-User Programming

ML approaches such as the ones based on LfD and AL introduced in the previous sections have been successfully employed as alternatives to traditional robot programming [4, 5, 26, 27, 28]. However, one disadvantage of these data driven techniques compared to traditional programming is their opaqueness [29, 30], *i.e.*, the fact that the final product of the training (*e.g.*, the model encoding the skill learned from demonstrations) is often not directly interpretable by humans. Model opaqueness makes standard debugging and refining of such models hard to achieve. For example, many LfD approaches lack the option to be point-wise modified and instead require the collection of corrective demonstrations to adjust the model.

In parallel to these ML-based approaches, researchers have also investigated how traditional robot programming can be modified to be accessible to people with little or no programming experience. The research area of End-User Programming (EUP) for robotics aims at this democratization through novel user interfaces, programming languages, and techniques to aid or fully automate robot programming [31, 32].

Research in this area has produced tools that allow novice users to create complex programs from discrete robot actions, with input modalities like visual programming [33, 34, 35], kinesthetic teaching [36, 37, 38, 39], and natural language commands [40, 41]. Many of these approaches have been recently adopted by robotics companies like Franka Emika or Rethink Robotics, whose robots come with intuitive programming interfaces. Robot actions such as linear motions and grasping actions are the building blocks of these EUP interfaces. These actions are often parametrized, with the number and complexity of parameters depending on the level of abstraction adopted in their design. While intuitive ways of specifying certain parameters have been proposed (*e.g.*, specifying the goal pose of a robot motion with kinesthetic teaching), other parameters must be manually tuned via Graphical User Interfaces (GUIs). For example, the speed of robot motions is often tuned with GUI elements such as sliders, as kinesthetically moving a robot arm with many Degrees of Freedom (DoFs) at the desired speed while following the desired trajectory can be challenging [42].

These less intuitive GUI-driven tuning approaches often require the user to adopt a *trial-and-error* strategy and execute each action or the whole program a number of times to find the correct parameter value. In Publication IV, we tackled the challenges of tedious manual parameter tuning, proposing an interactive approach whereby the robot iteratively suggests parameter values and gathers the user's feedback to find feasible parameter ranges. The main contributions of Publication IV are

1. the formulation of a 1-dimensional parameter search as an AL problem,

along with a Bayesian framework that can encode prior knowledge about such parameter values, and

2. the design of queries as robot action executions and the integration of *directional answers* into the interactive tuning process.

1.1.5 Interacting with Learning Robots

Along with the previously highlighted benefits, making learning robots more interactive creates new HRI challenges. Learning approaches that involve humans-in-the-loop should take into account their interactive nature [4, 43], paying close attention to factors like control over the training process [44], its transparency [21, 45], and people's ability to be good teachers for machine learners [12].

These interaction aspects become extremely relevant in the case of AL, where the robot learner and the human teacher engage in a tight dyadic interaction structured around the answering of questions. Researchers have therefore studied the interactive nature of AL robots, investigating their design [44, 46], their queries [46, 47, 48, 49, 50, 51], the users' ability to answer such queries [45, 52], and their timing [53].

In Publication II, we investigated some of these aspects for our AL-LfD approach by conducting a user study with novice users comparing three different query selection strategies. Interestingly, we observed how standard AL selection strategies, solely aimed at efficient information gathering, would choose queries (or sequences of queries) deemed difficult to answer and distracting by the study participants. When attempting to answer such queries, the user may teach at a lower pace or inadvertently introduce errors in the training, ultimately reducing the efficiency of traditional AL approaches. These observations raised the following questions: what happens when we do not consider teachers an ever-present, infallible source of information? And could query selection strategies adapt to real users' idiosyncrasies and potentially perform better than traditional strategies?

To answer these questions, in Publication III we challenged the idea that AL sample efficiency reduces the effort required of the user acting as a teacher. More specifically, the relationship between subsequent queries was considered as the primary source of difficulty for the teacher. We studied an information gathering problem where an AL robot learns about the *attributes* of *entities* grouped by *categories*. With the assumption that entities in the same category are likely to share the same attribute value, the learning agent could select informative queries with *Uncertainty Sampling* [54], a traditional query selection strategy. We hypothesized that this traditional strategy, while maximizing information gain, would query about entities that were as distant as possible[2] from each

[2]Distance with respect to the entities' membership of the provided categories. Following the animal topic used in Publication III, cows and ibexes are considered close, as

other, increasing the workload of human teachers, making them slower and more prone to errors. Informed by cognitive models of memory retrieval [55], we then proposed a strategy aware of the teacher's memory efforts that minimizes the distance between consecutive queries. The main contributions of Publication III are

1. integration of the concept of memory retrieval into a query selection strategy,
2. simulation study of the performance of such a strategy, compared to a traditional AL strategy, and
3. study of the effects of these different selection mechanisms with human teachers, analysing their mental workload, error rates, response times, and the overall training performance.

1.1.6 Robot Transparency through Policy Explanation

While introducing our research on AL, the argument was made that robot teaching with humans-in-the-loop should be considered a collaborative task requiring the creation of *common ground*, with both agents sharing information and influencing each other's actions. In addition, attention was drawn to the difficulty of building common ground between robots and humans, due to substantial differences between these agents. Nevertheless, the accurate perception of a robot's capabilities, intent, and limitations – referred to in the literature as *transparency* [13, 56, 57] – is pivotal if the users are to trust the system [58, 59] and correctly calibrate their reliance on it.

As robots' autonomous capabilities increase, the accountability, fairness, and safety of intelligent systems become pressing issues. Motivated by the present popularity of black-box approaches as primary tools to achieve such autonomous capabilities, researchers have begun to investigate transparency mechanisms to interpret and explain the internal representations and decision making of autonomous systems [30, 60, 61, 62, 63].

After observing the effects of transparency (and the lack of it) in our AL robots, in Publication V we tackled the more general problem of explaining robot policies through natural language explanations. In particular, we proposed a method that generates explanations for policies defined on a continuous state space with discrete actions. The main contributions of Publication V are

1. an explanation generation method that is model agnostic and can be applied to black-box policy representations as long as the dimensions of its state space can be described in natural language terms,

---

they share membership of many categories (*e.g.*, mammals, ruminants and bovidae). Conversely, lions and cows are to be considered distant, as they share membership of fewer categories.

2. a mechanism to make such explanations *focused*, omitting dimensions of the state space that, for example, do not locally influence the policy's choice or that cannot be reliably described with the available vocabulary, and

3. a study of the effects of these focused explanations on the user's understanding of the robot's policy.

1.2 Structure of the Dissertation

The dissertation presents the included publications following the order adopted in Section 1.1. After an overview of robot LfD, Chapter 2 presents the proposed LfD framework for in-contact tasks. After introducing the core ideas behind AL and surveying the robotics literature on AL approaches, Chapter 3 presents both Publication II and Publication IV, covering the technical details and highlighting design considerations to be made when humans are present in the learning loop. Chapter 4 completes the presentation of our work on AL robots, examining their HRI aspect and introducing the major results from Publication II and Publication III. Investigating the problem from the interaction perspective will lead to the challenges of robot transparency and to the policy explanation technique of Publication V, presented in Chapter 5. Finally, Chapter 6 reiterates the main contributions of this dissertation, discussing open research questions and future challenges.

2. Learning In-contact Tasks from Demonstrations

Programming robots to perform tasks that require contact with the environment, such as ironing clothes or pulling door handles, is rather complex. To accomplish such tasks, robot manipulators require extra sensing capabilities (*e.g.*, joint torque sensing to detect contact and collisions), specific programming primitives (*e.g.*, guarded motions), and appropriate control strategies (*e.g.*, impedance control). More complex tasks may also require the robot to exert specific force and torque profiles while in contact with the environment (*e.g.*, pushing with suitable force while kneading a mass of dough). As these profiles are hard to encode in a declarative way, traditional programming of *in-contact tasks* can quickly become prohibitively complex.

To address the programming of in-contact tasks, in Publication I we proposed a LfD approach to allow robots both to learn trajectories and force profiles from human demonstrations, therefore avoiding the declarative encoding of complex force profiles. This chapter presents the proposed LfD approach, focusing on its three main aspects: the collection of demonstrations via kinesthetic teaching, the learning of a suitable task representation through a combination of HSMMs and GMR, and the task execution using a CIC with varying stiffness.

2.1 The Learning from Demonstration Pipeline

As introduced in Chapter 1, LfD, also known as PbD and IL, is a learning paradigm that allows the programming of robots by providing examples of the desired behaviour [5]. LfD is appealing for non-expert users because it translates their ability to perform a desired task (in the form of demonstrations) into their ability to program a robot to achieve the task. Viewing the problem from a ML perspective, LfD is an instance of supervised learning, with the robot learning a task representation from a labelled dataset (*i.e.*, the demonstrations). Thus, the generic LfD pipeline consists of two steps: the collection of the demonstrations and the learning of a suitable task model.

2.1.1 Collecting Demonstrations

For the demonstration collection, the key design choices are (i) the choice of the demonstrator, (ii) the nature of the demonstrations, and (iii) the choice of demonstration interface [5]. The demonstrator is often a human (referred to also as *teacher*), although demonstrations can be provided by other agents, such as other robots. What a demonstration practically is (*i.e.*, what kind of data is considered a demonstration) depends instead on the taught task and the level of abstraction adopted in the learning phase. Commonly, demonstrations are time series of sensor readings and commands, but higher level representations are possible (*e.g.*, datasets of labelled images, to detect pre and post-conditions for learning task plans). Finally, the choice of demonstration interface depends on both of the previously presented choices and the sensors available for the recording. A threefold categorization of demonstration interfaces is presented in [28], distinguishing between *teleoperation*, *kinesthetic teaching*, and *passive observation*.

With teleoperation, the teacher demonstrates the task through an external input device. Teleoperation approaches can take advantage of already available input devices, such as joysticks and teach pendants. However, depending on the requirements of the specific application, more advanced input devices can be used, including haptic devices or virtual-reality interfaces. The teacher can therefore provide demonstrations for any robot equipped with a suitable input device; furthermore, demonstrations can be collected for remotely located robots, unlocking LfD for applications in remote and hazardous environments. In turn, the main drawbacks of teleoperation are (i) the extra effort required to develop new interfaces or adapt existing ones to the collection of demonstrations, and (ii) the availability and cost of the chosen input devices.

With kinesthetic teaching, the teacher demonstrates the task by physically displacing a backdrivable or gravity compensated robot while the demonstrations are recorded using the robot's proprioceptive sensors (*e.g.*, joint positions and velocities, joint torques, and loads). While providing an intuitive interface that requires little training for the teacher [64], having the teacher physically move the robot places stricter requirements regarding the safeness of interaction. Moreover, the robot requires specific features, such as backdrivable motors and gravity compensation. Finally, kinesthetic teaching effectiveness decreases as the number of a robot's DoFs increases, as simultaneously operating many DoFs in a coordinated and smooth fashion is extremely challenging. These requirements restrict the use of kinesthetic teaching to lightweight manipulators such as the KUKA LWR4+ and the Franka Emika Panda, excluding systems that are not interaction-safe (*e.g.*, hydraulic industrial manipulators) and systems with a high DoF count (*e.g.*, humanoid robots).

Finally, with passive observation interfaces, demonstrations are collected without directly using the robot. Instead, the teachers perform the target task themselves while their activity is tracked through vision systems, such

as cameras or motion capture systems, or with sensors placed directly on their bodies. With passive observation, teachers are able to demonstrate the task in the most natural way, *i.e.*, by using their own embodiment. This increases the intuitiveness of providing demonstrations even for systems with many DoF, such as humanoid robots or complex robotic hands. Furthermore, as with teleoperation, the co-presence of the teacher and robot is not required. However, the main drawback lies in the correspondence problem between the teacher's and the robot's embodiment [65]. As demonstrations are recorded in the teacher's embodiment, a mapping to the robot's embodiment needs to be either manually specified or learned, further complicating the subsequent learning problem. Moreover, passive observation interfaces inherit the limitations of the sensors used for the recording, such as occlusion for vision systems.

2.1.2 Model Learning

Once the demonstrations are collected, LfD essentially becomes a supervised learning problem. We can categorize LfD approaches based on (i) what is learned from the demonstrations (the nature of the learned model) and (ii) what actual combination of model and learning method is used. Regarding the nature of the learned model, a threefold categorization is presented in [5, 28], distinguishing between learning *policies*, *reward functions*, and *plans*.

The most common approach adopted in the literature is directly learning a policy, *i.e.*, a function mapping the information available to the robot to an appropriate action space. As this family of approaches is the most relevant for the work of Publication I, the other two approaches are only introduced briefly here. For a more in-depth discussion, numerous surveys are available on the topic [4, 5, 26, 28].

With IRL, demonstrations are used to infer the function that informs the robot of what action is beneficial in different situations, *i.e.*, a reward function [14, 18]. By learning the reward function from demonstrations, IRL techniques allow the adoption of RL techniques in robot learning problems when it is difficult to manually specify a reward function but it is instead easy to provide examples of highly rewarding behaviours. These techniques therefore inherit RL's advantages (*e.g.*, self learning based on a reward signal) and disadvantages (*e.g.*, the credit assignment problem and the exploration-exploitation dilemma). Another option is to use the demonstrations as high-level descriptors of the task (*e.g.*, action pre and post-conditions and task goals) and learn sequential or hierarchical plans of discrete actions. While plans allow the encoding of more complex tasks, additional techniques (*e.g.*, motion level models) are often required for the robot to achieve the actual task execution.

With policy methods, the goal is to learn a mapping $\pi : S \rightarrow A$ from the state space S of the robot to its action space A from the demonstration set $\mathcal{D}$. These methods are particularly suited to trajectory-level encodings, where policies are often referred to as motion primitives [27].

Policy methods can be further categorized based on their input and output spaces [28]. The input of the policy can simply be time, with demonstrations being time series of any variable necessary for the robot to follow the desired trajectory; these models are analogous to open-loop controllers, where the robot's actions are computed based solely on the current time, without additional feedback. If perturbations must be handled during task execution, the state space S can include any variable useful for computing a corrective command, in a fashion similar to close-loop controllers. In this case, demonstrations consist of state-action pairs, for example the torque that must be applied at each joint in order to follow the demonstrated trajectory. Regarding the nature of action space A, *i.e.*, the output of the policy, the main distinction is between discrete and continuous action spaces. The nature of the robot actions dictates the underlying ML problem to be solved: classification for discrete action spaces, regression for continuous ones.

Finally, the learned policy can be deterministic or stochastic. The following question provides a good rule of thumb for this design choice: is the provided demonstration a perfect representation of the desired behaviour, or just one of many, similarly acceptable, behaviours? In the former case, the policy should be deterministic, as there is no need to deviate from the perfect demonstration provided by the teacher. Conversely, in the latter case, a stochastic policy may be more appropriate, as it can inherently model the uncertainties of the provided demonstrations.

Once the characteristics of the policy are identified based on the task at hand and the nature of the collected demonstrations, a suitable model and related learning approach are selected. Several solutions have been proposed for learning trajectory-level tasks, from classic regression methods like Locally Weighted Regression (LWR), Gaussian Process Regression (GPR) and GMR, to dynamical systems like Dynamic Movement Primitives (DMPs) [66] and Probabilistic Movement Primitives (ProMPs) [67]. In this dissertation, the combination of HSMM and GMR as used in Publication I is briefly presented; the reader is referred to [27] for an extensive review of motion primitive learning.

2.2 Learning from Demonstration for In-contact Tasks

Publication I focused on LfD for in-contact tasks. Three aspects of in-contact tasks informed our choices regarding the design of the LfD pipeline. First, in-contact tasks exhibit a tight temporal coupling between the pose requirements, *i.e.*, the trajectory to follow, and the force requirements, *i.e.*, the force profiles to exert on the environment. Second, in-contact tasks require the teacher to demonstrate multiple, and equally important, aspects of the task simultaneously (*e.g.*, the trajectory, the force profile, the speed of the motion). Third, the manipulator must be compliant with the environment and be able to perform both free space motions and compliant motions.

Based on these requirements, we made the following design choices. Regarding the collection of the demonstrations, we adopted kinesthetic teaching with the simultaneous recording of trajectories and force profiles. To encode the demonstrated tasks, we chose a trajectory representation based on HSMMs and GMR, for their ability to encode the fine temporal aspects of the task and to cope with the foreseeable noise in the users' demonstrations. Finally, we opted for a CIC with varying stiffness and a feed-forward force term for the execution of the learned task on our manipulator of choice, a KUKA LWR4+ [68].

2.2.1 Demonstrating In-contact Tasks

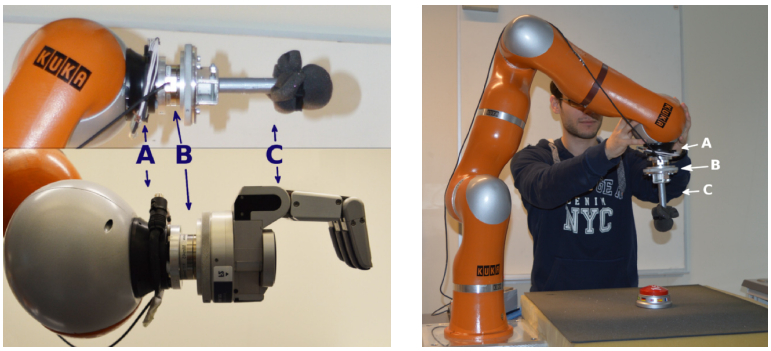
The recording of forces is pivotal for the execution of in-contact tasks, and demonstration interfaces must therefore allow for the reliable recording of force profiles. While promising work has shown how contacts can be estimated using RGB-D cameras when the manipulated objects are known a priori [69], estimating contact forces through vision is far from reliable, making passive observation interfaces barely usable. Similarly, placing tactile sensors on the teacher is not yet a cost effective solution, although research in this area shows great potential for LfD applications [70, 71].

For teleoperation and kinesthetic teaching, the recording of contact forces presents challenges that are less related to the actual sensing and more to the teacher side of the demonstration. Simple teleoperation interfaces, like joysticks and teach pendants, can rarely provide feedback to the teacher on the interaction of the robot with the environment (*e.g.*, the amount of exerted force and potential collisions). The lack of such feedback can lower the quality of the demonstrations, and consequently of the whole LfD pipeline.

While more complex teleoperation interfaces, such as haptic devices, have been successfully employed [72, 73], in Publication I we opted for kinesthetic teaching, as chosen in [74, 75, 76, 77, 78]. For the recording of contact forces however, we could not rely on the torque sensing capabilities at the manipulator's joints. As the teacher moves the robot during the demonstration, the forces exerted on the environment are produced by the teacher, making the integrated torque sensing capabilities of the robot uninformative. Hence, a dedicated ATI mini 45 Force/Torque sensor was mounted between the robot's flange and tool. With this configuration, shown in Figure 2.1, the teacher grasps the robot above the Force/Torque sensor, allowing the recording of forces applied to the environment through the robot's tool. Finally, the configuration also allows for the simultaneous recording of both trajectories and force profiles. This avoids the adoption of the 2-phase recording scheme presented in [73], where the trajectory and the force profiles are learned separately, possibly introducing errors due to the extra synchronization effort required of the teacher.

From a recording phase, a demonstration is obtained, defined as

$$\left(\boldsymbol{X}_t = \left[\begin{array}{ccccc} \boldsymbol{x}_t^T & \boldsymbol{q}_t^T & \boldsymbol{v}_t^T & \boldsymbol{\omega}_t^T & \boldsymbol{f}_t^T \end{array} \right]^T \right) \text{ with } t = 1, \ldots, L, \tag{2.1}$$



(a) Sensor configuration. **(b)** Kinesthetic teaching.

Figure 2.1. The sensor configuration for recording in-contact tasks adopted in Publication I and related kinesthetic teaching instance, with highlighted robot flange (A), Force/Torque sensor (B), and tool (C). Adapted from Publication I. © 2016 IEEE.

i.e., an L-time-steps long time series of the pose of the robot's end-effector (position $\boldsymbol{x}_t \in \mathbb{R}^3$ and quaternion representation of the rotation $\boldsymbol{q}_t \in \mathbb{H}$, expressed with respect to a fixed reference frame at the base of the robot), its translational and rotational velocities ($\boldsymbol{v}_t$ and $\boldsymbol{\omega}_t$ respectively), and the force readings from the Force/Torque sensor ($\boldsymbol{f}_t \in \mathbb{R}^3$, expressed in the end-effector frame). The demonstrations were recorded at 100 Hz.

While the simultaneous recording of trajectories and forces should allow teachers to provide demonstrations of sufficient quality, we still cannot expect them to demonstrate correctly all aspects of the task at the same time. For example, the teacher could make the robot follow the desired path while exerting the required force at an incorrect speed. Furthermore, a perfect execution may not exist at all for some tasks. Consequently, it may be beneficial to encode the variability of the teacher's demonstrations. Following the reasoning presented in Section 2.1.2, we therefore collected multiple demonstrations and learned a stochastic trajectory representation from them.

2.2.2 Learning Models for In-contact Tasks

Once demonstrations have been collected, an appropriate task model can be trained. We opted for GMR [79], a regression method particularly popular in the LfD field [80]. With GMR, the regression function is not directly modelled but is instead derived from a joint probability density function of the variables of interest. The main advantage of GMR lies in the fact that the computationally intensive density estimation procedure is performed offline, while the computation of the robot's command (*i.e.*, the output of the learned policy) is performed rapidly at run time. This is achieved through the linear transformation and conditioning properties of Multivariate Normal distributions. Thus, popular choices for the joint probability density model are Gaussian Mixture Models (GMMs), Hidden Markov Models (HMMs) [81] with Gaussian observation probabilities,

and HSMMs [82].

Prior to Publication I, the encoding of force profiles from demonstrations had been achieved with DMPs [73, 74, 75]. With such a representation, however, each dimension is encoded separately, and the correlations between pose and force profiles are not learned. Our contribution in Publication I is the inclusion of the force information in the HSMM-GMR statistical model first proposed in [80], allowing for the encoding of the aforementioned correlations between pose and force profiles. With the chosen representation, each state of the HSMM models a different portion of the taught trajectory. HSMMs also offer benefits over models that can be used with GMR, namely GMMs and HMMs. Compared to GMMs, HSMMs directly model the temporal evolution of the system, allowing, for example, self-intersecting and cyclic trajectories. Furthermore, HSMMs can be learned from unaligned demonstrations of different lengths, avoiding the use of sequence alignment methods required by GMMs and DMPs, such as Dynamic Time Warping [83]. Finally, the duration of each state is explicitly modelled, improving on the strictly exponential nature of state duration densities of standard HMMs.

More specifically, a HSMM with N states is parametrized as

$$\lambda = (\boldsymbol{\pi}, T, \mu, \Sigma, \mu^D, \sigma^D), \tag{2.2}$$

where $\boldsymbol{\pi} \in \mathbb{R}^N$ are the starting probabilities and $T \in \mathbb{R}^{N \times N}$ is the transition matrix, whose element t_{ij} represent the probability of moving from state i to state j. Each state has an observation distribution, modelling the distribution of the variables of interest recorded in the demonstrations. Parameters $\mu = \left\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_N\right\}$ and $\Sigma = \{\Sigma_1, \ldots, \Sigma_N\}$ are sets of means and covariance matrices, one for each of the N Multivariate Normal distributions modelling the observation probabilities for each state. Finally, $\mu^D = \left\{\mu_1^D, \ldots, \mu_N^D\right\}$ and $\sigma^D = \left\{\sigma_1^D, \ldots, \sigma_N^D\right\}$ are sets of means and variances, one for each of the N Univariate Normal distributions modelling the duration probabilities of each state.

These parameters are learned from the teacher's demonstrations by a HSMM-specific variant of the Baum-Welch algorithm [84]. The number of states N can be selected as a trade-off between the accuracy and complexity of the model with model selection techniques like the Bayesian Information Criterion (BIC) [85]. Similarly, the number of demonstrations is chosen by taking into account the resources required to produce the demonstrations, for example the time and effort of the teacher and the quality of the trained model.

For our trajectory modelling problem, we characterized the observation $\boldsymbol{X}_t$ at time t as

$$\boldsymbol{X}_t = \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{q}_t \\ \hdashline \boldsymbol{v}_t \\ \boldsymbol{\omega}_t \\ \boldsymbol{f}_t \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}_t^{\mathrm{I}} \\ \hdashline \boldsymbol{X}_t^{\mathrm{O}} \end{bmatrix}, \tag{2.3}$$

where $\boldsymbol{X}_t^{\mathrm{I}}$ is the input of the policy, *i.e.*, the current state, and $\boldsymbol{X}_t^{\mathrm{O}}$ is the output of the policy, *i.e.*, the commands. Following the same reasoning, the mean and the covariance matrix of each observation distribution is characterized as follows

$$\boldsymbol{\mu}_i = \begin{bmatrix} \boldsymbol{\mu}_i^x \\ \boldsymbol{\mu}_i^q \\ \hdashline \boldsymbol{\mu}_i^v \\ \boldsymbol{\mu}_i^\omega \\ \boldsymbol{\mu}_i^f \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_i^{\mathrm{I}} \\ \hdashline \boldsymbol{\mu}_i^{\mathrm{O}} \end{bmatrix}, \quad \Sigma_i = \left[\begin{array}{ccc:ccc} \Sigma_i^x & \Sigma_i^{xq} & & \Sigma_i^{xv} & \Sigma_i^{x\omega} & \Sigma_i^{xf} \\ \Sigma_i^{qx} & \Sigma_i^q & & \Sigma_i^{qv} & \Sigma_i^{q\omega} & \Sigma_i^{qf} \\ \hdashline \Sigma_i^{vx} & \Sigma_i^{vq} & & \Sigma_i^v & \Sigma_i^{v\omega} & \Sigma_i^{vf} \\ \Sigma_i^{\omega x} & \Sigma_i^{\omega q} & & \Sigma_i^{\omega v} & \Sigma_i^\omega & \Sigma_i^{\omega f} \\ \Sigma_i^{fx} & \Sigma_i^{fq} & & \Sigma_i^{fv} & \Sigma_i^{f\omega} & \Sigma_i^f \end{array}\right] = \left[\begin{array}{c:c} \Sigma_i^{\mathrm{I}} & \Sigma_i^{\mathrm{IO}} \\ \hdashline \Sigma_i^{\mathrm{OI}} & \Sigma_i^{\mathrm{O}} \end{array}\right]. \tag{2.4}$$

We can compute through GMR, at each time step t, the probability of observing the command $\boldsymbol{X}_t^{\mathrm{O}}$ given the current state $\boldsymbol{X}_t^{\mathrm{I}}$ as

$$P(\boldsymbol{X}_t^{\mathrm{O}} \mid \boldsymbol{X}_t^{\mathrm{I}}) = \sum_{i=1}^{N} h_{i,t}\, \mathrm{N}(\boldsymbol{X}_t^{\mathrm{O}} | \hat{\boldsymbol{\mu}}_i^{\mathrm{O}}(\boldsymbol{X}_t^{\mathrm{I}}), \hat{\Sigma}_i^{\mathrm{O}}), \tag{2.5}$$

with $\hat{\boldsymbol{\mu}}_i^{\mathrm{O}}(\boldsymbol{X}_t^{\mathrm{I}})$ and $\hat{\Sigma}_i^{\mathrm{O}}$ defined as

$$\begin{aligned} \hat{\boldsymbol{\mu}}_i^{\mathrm{O}}(\boldsymbol{X}_t^{\mathrm{I}}) &= \boldsymbol{\mu}_i^{\mathrm{O}} + \Sigma_i^{\mathrm{OI}}(\Sigma_i^{\mathrm{I}})^{-1}(\boldsymbol{X}_t^{\mathrm{I}} - \boldsymbol{\mu}_i^{\mathrm{I}}), \\ \hat{\Sigma}_i^{\mathrm{O}} &= \Sigma_i^{\mathrm{O}} - \Sigma_i^{\mathrm{OI}}(\Sigma_i^{\mathrm{I}})^{-1}\Sigma_i^{\mathrm{IO}}. \end{aligned} \tag{2.6}$$

The conditional probability of the command $\boldsymbol{X}_t^{\mathrm{O}}$ is modelled as a mixture of Multivariate Normal distributions; we can therefore compute the mean command $\boldsymbol{X}_t^{\mathrm{O}^*}$ as the weighted average of each component's mean as

$$\boldsymbol{X}_t^{\mathrm{O}^*} = \begin{bmatrix} \boldsymbol{v}_t^* \\ \boldsymbol{\omega}_t^* \\ \boldsymbol{f}_t^* \end{bmatrix} = \sum_{i=1}^{N} h_{i,t}\, \hat{\boldsymbol{\mu}}_i^{\mathrm{O}}(\boldsymbol{X}_t^{\mathrm{I}}). \tag{2.7}$$

The weighting term $h_{i,t}$ in Equations 2.5 and 2.7 represents the contribution of each state i to the computation of the current command $\boldsymbol{X}_t^{\mathrm{O}^*}$. For HSMMs, $h_{i,t}$ is the current belief over the states, *i.e.*, a normalized version of the forward variable $\alpha_{i,t}$. More details about the computation of the forward variable $\alpha_{i,t}$ based on the current observation history $\left(\boldsymbol{X}_1^{\mathrm{I}}, \ldots, \boldsymbol{X}_{t-1}^{\mathrm{I}}\right)$ are available in Publication I. With Equation 2.7, we have the commands that allow the robot to follow the demonstrated trajectories.

Finally, regarding the computation of the command signals presented in Publication I, while the force command $\boldsymbol{f}_t^*$ is correctly computed taking into account both the current position and orientation (*i.e.*, the whole input $\boldsymbol{X}_t^{\mathrm{I}}$, as in Equation 2.7), the computation of $\boldsymbol{v}_t^*$ does not take into account the current orientation $\boldsymbol{q}_t$. Similarly, the computation of $\boldsymbol{\omega}_t^*$ does not consider the current position $\boldsymbol{x}_t$. Omitting such terms is equivalent to assuming that the translational velocity does not depend on the current orientation (*i.e.*, assuming that Σ_i^{vq} is a zero matrix). We believe this to be a methodological mistake worth reporting, even though the task executions in the experiments presented in Publication I do not seem to be impacted by it.

2.2.3 Executing In-contact Tasks

For the execution of the learned in-contact tasks, a control strategy that could handle both free space motions and compliant motions was required. We opted for Impedance Control (IC) [86], a control strategy that imposes the dynamic behaviour of a *mass-spring-damper* system at the interface between the manipulator and the environment. More specifically, we used the CIC of the KUKA LWR4+ [87], with control law

$$\boldsymbol{\tau}_{\text{cmd}} = \boldsymbol{J}^T(\text{diag}(\boldsymbol{k}_c)(\boldsymbol{x}_{\text{cmd}} - \boldsymbol{x}_{\text{msr}}) + D(\boldsymbol{d}_c) + \boldsymbol{f}_{\text{cmd}}) + \boldsymbol{\tau}_{\text{dyn}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}), \tag{2.8}$$

with the Jacobian $\boldsymbol{J}$, the Cartesian stiffness parameters $\boldsymbol{k}_c$, the Cartesian damping parameters $\boldsymbol{d}_c$, a superposed feed-forward force term $\boldsymbol{f}_{cmd}$, and the dynamic model $\boldsymbol{\tau}_{\text{dyn}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$. For our in-contact task scenario, the relevant terms of the control law are $\boldsymbol{k}_c$, representing the virtual spring between the commanded pose $\boldsymbol{x}_{\text{cmd}}$ and the measured pose $\boldsymbol{x}_{\text{msr}}$, and $\boldsymbol{f}_{cmd}$. During the execution of the learned strategy, $\boldsymbol{x}_{\text{cmd}}$ is computed from the commands $\boldsymbol{X}_t^{\text{I}}$ of Equation 2.7 by integration as follows:

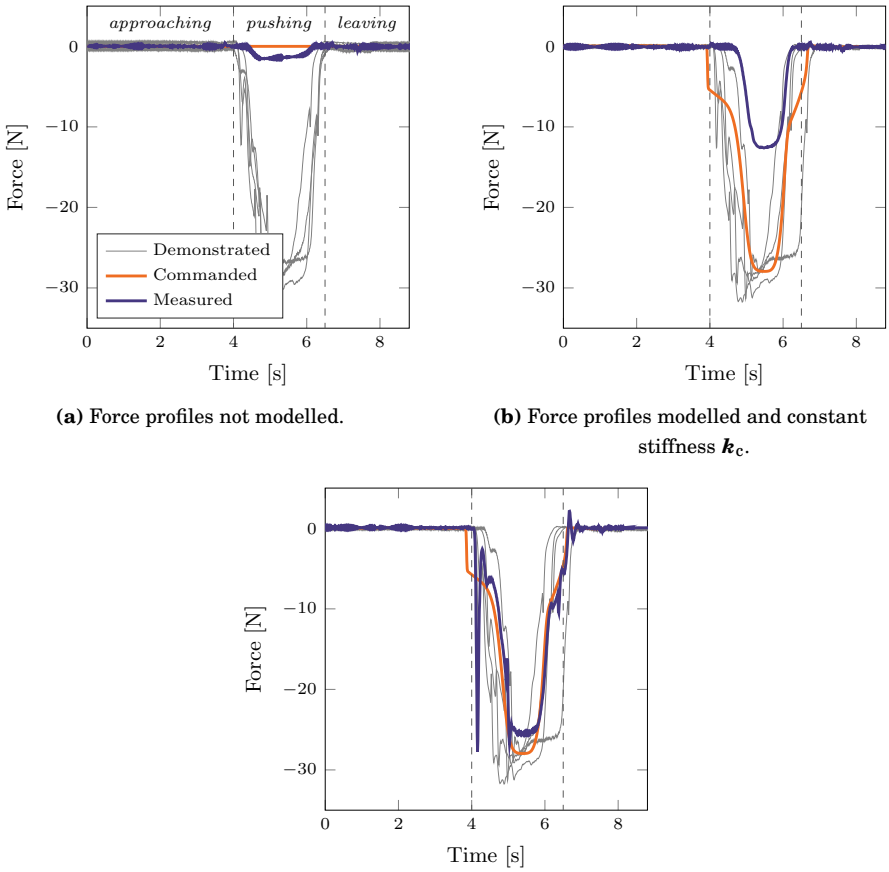
$$\boldsymbol{x}_{\text{cmd}} = \begin{bmatrix} \boldsymbol{x}^*_{\boldsymbol{t+1}} \\ \boldsymbol{q}^*_{\boldsymbol{t+1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x_t} + \tau \boldsymbol{v}^*_{\boldsymbol{t}} \\ e^{\frac{1}{2}\tau \boldsymbol{\omega}^*_{\boldsymbol{t}}} \otimes \boldsymbol{q_t} \end{bmatrix}, \tag{2.9}$$

where τ is the duration of a control time step. Given its feed-forward nature, the force term $\boldsymbol{f}_{cmd}$ is obtained directly from the $\boldsymbol{f}_t^*$ of Equation 2.7.

2.2.4 Results and Discussion

We evaluated the proposed LfD pipeline with two experiments: (1) the pushing of a stiff button (shown in Figure 2.1.b) and (2) the pulling of a door handle. The results from Experiment 1 are summarized here as they concisely present the strengths of the proposed pipeline; the reader is referred to Publication I for the results of Experiment 2.

Figure 2.2.a shows the force profile measured along the tool's pushing direction when contact forces are not modelled. In this case, the robot fails to exert sufficient force to activate the push button, motivating the inclusion of force profiles pursued by our method. Figure 2.2.b shows instead the proposed LfD pipeline, with the linear and the angular components of the controller's stiffness $\boldsymbol{k}_c$ set to a constant value of, respectively, 2000 $\frac{\text{N}}{\text{m}}$ and 200 $\frac{\text{Nm}}{\text{rad}}$. We can see from the commanded force profile how the HSMM-GMR model can learn the desired force profile from the demonstrations. However, the measured force profile does not follow the commanded profile and the robot still fails to activate the push button. This undesirable behaviour is caused by the interaction of the simulated spring term and the feed-forward force term of Equation 2.8: during execution, these two terms can counteract each other and prevent the manipulator from exerting the commanded $\boldsymbol{f}_t^*$.



(a) Force profiles not modelled.

(b) Force profiles modelled and constant stiffness $\boldsymbol{k}_c$.

(c) Force profiles modelled and stiffness $\boldsymbol{k}_c$ selected with proposed mechanism.

Figure 2.2. Force profiles (along the tool's pushing direction) from the push button experiment of Publication I. The exertion of the demonstrated force profiles is accurately achieved only with the proposed stiffness selection mechanism. Adapted from Publication I. © 2016 IEEE.

To achieve the tracking of both force and position profiles during execution, a hybrid control method was adopted in [74, 75], switching between different controllers during the task execution. While effective and simple, this method requires manually specified switching conditions that are possibly subject to noisy sensor readings. Kormushev *et al.* propose, instead, that the desired stiffness be computed from the variability observed in the positional data from the demonstrations [73]. We, instead, propose that the stiffness $\boldsymbol{k}_c$ be adjusted based on the force information encoded in the HSMM. In particular, we compute the probability of each of the states of the HSMM to encode free space motions or compliant motions by modelling the force readings from the demonstrations as a two-components mixture, with the type of components selected experimentally. These probabilities allow the stiffness $\boldsymbol{k}_c$ to be selected during execution,

essentially smoothly switching between the CIC and the feed-forward force controller of Equation 2.8. While manually specified, the components of the mixture are easier to select and more robust than the manually specified thresholds of [74, 75]. More details on the proposed stiffness selection mechanism can be found in Publication I. Figure 2.2.c shows how the commanded force profile is correctly exerted on the environment, allowing the robot to successfully push the button.

While the work of Publication I focused on learning motions at trajectory-level, robots often require a skilful combination of low-level motions to carry out their tasks. As explained in Section 2.1.2, trajectory-level LfD approaches can be integrated with approaches that create high-level representations of the task, such as plans [88, 89, 90].

In the proposed approach, the trajectory part of the demonstrations is recorded in a fixed frame of reference defined at the base of the robot. In the real world problems described in Chapter 1 however, motions often need to be performed with respect to relevant objects placed in the robot's working envelope. Combined with an appropriate system able to track task-relevant objects (*e.g.*, a vision system), the proposed approach can be extended to handle such task-oriented demonstrations by adopting the method proposed in [91].

2.3 Strengths and Weaknesses of Learning from Demonstration

Building on this chapter's presentation of the general LfD pipeline and of our contribution to the specific case of in-contact tasks, this section discusses the strengths and weaknesses of LfD approaches in the light of the dissertation's motivations.

As briefly explained in Chapter 1, LfD approaches have the potential to greatly contribute to the goal of achieving programmability and adaptability for service robots. Primarily, LfD provides non-expert users with an intuitive yet effective way of programming robots. Furthermore, LfD approaches allow the programming of complex tasks that are hard to specify in a declarative way, such as the case of in-contact tasks addressed in Publication I. Another advantage of LfD lies in its data efficiency. Most LfD approaches can learn task models from less than 10 demonstrations, with some techniques able to learn from even a single demonstration. The data efficiency of LfD approaches becomes particularly relevant when comparing them to techniques based on self-learning and exploration, such as RL, that require the robot to act in a possibly suboptimal and unsafe fashion in order to learn. LfD approaches are, however, far from incompatible with exploration-based learning techniques. On the contrary, demonstrations have often been used as the starting point for the exploration process [92, 93, 94, 95], to provide safety boundaries [96, 97], or to learn complex reward functions [98].

However, one of the most evident weakness of LfD approaches lies in the costs imposed by the collection of demonstrations. As demonstrations determine

the quality of the learned models, demonstration interfaces should be carefully designed or selected – a process that is often task-specific, time consuming and expensive, especially when it requires the integration of extra sensors. On a more general level, programming robots only through demonstrations can be restrictive, especially when the learned model presents issues that must be corrected. Most incremental LfD approaches require the teacher to provide extra *corrective demonstrations* to resolve such issues [99]. This is obviously not ideal, as it requires the teacher to demonstrate the entire task even when corrections are needed only in portions of it. We therefore believe that LfD approaches would greatly benefit from allowing more flexible correction modalities, like partial demonstrations [100, 101].

Finally, one of the major weaknesses of LfD approaches is that they require the teacher to provide informative demonstrations, *i.e.*, demonstrations that truly help the training process. This, in turn, requires the teacher to be an expert not only at the taught task but also at teaching a robot – an assumption that rarely holds true in real scenarios. For example, the teacher may not know what requirements the demonstrations must satisfy to be informative or how many demonstrations are required or how diverse they should be. Likewise, the teacher may be unable to identify weaknesses and problems in the current model and consequently fail to provide corrective demonstrations. Moreover, the teacher may not provide all possible successful demonstrations of a task simply because of forgetfulness, resulting in the learning of incomplete models. As discussed in Section 1.1.2, these issues are not specific to LfD approaches but appear in any approach where there are inevitable discrepancies between the robot's model (*e.g.*, the chosen skill representation) and the teacher's model. The true weakness of LfD lies in the fact that teachers are solely responsible for the training process through the demonstrations they provide. In other words, the learner has no mechanism to react to the user's sub-optimal teaching and, consequently, the overall training process is negatively affected. We therefore believe that LfD approaches should be augmented so as to be interactive, with the robot learner taking an active part in the teaching process by requesting specific information, exposing its knowledge gaps and steering the user's teaching efforts. This line of reasoning motivates most of the research of this dissertation and, in particular, the work on AL presented in the next chapter.

3. Active Robot Learning from Humans: the Learning Perspective

The previous chapter introduced the LfD paradigm to robot learning and discussed its advantages and disadvantages. In particular, attention was drawn to how the information in LfD approaches flows only from the teacher to the learner and how this can negatively impact the training process when the user teaches in a suboptimal fashion. One way to address this issue is to have the learner actively participate in the teaching, steering the learning according to the knowledge gaps and inconsistencies of the current model. This idea is the working principle behind AL.

AL is a ML paradigm in which the agent chooses what to learn from [22, 23, 24]. Unlike traditional supervised learning techniques that solely learn from *labelled data*, active learners can select informative samples among *unlabelled* data – referred to as *queries* – and obtain labels from a labelling source – referred to as a *teacher* or *oracle*. By selecting informative queries, active learners can steer the learning process to cover their current knowledge gaps. This search for informativeness allows active learners to improve their models more efficiently (*i.e.*, with less labelled samples) compared to passive learning techniques [22, 102], making AL especially beneficial in learning scenarios where unlabelled samples are plentiful or cheap to obtain but the labelling costs are high.

Robot learning is one such scenario where the collection of labelled data is expensive. For a robot, learning often requires it to operate on the environment (*e.g.*, trying grasps on several objects), thereby increasing training time. Training a robot becomes even more expensive as humans become an integral part of the learning loop, with the learning process requiring people's time and patience. Consequently, AL is a particularly relevant approach, as its learning efficiency reduces the use of such expensive resources.

Motivated by the successes of AL, we applied it to two robot learning problems with humans-in-the-loop. In Publication II, we used AL as an extension of a LfD approach for the teaching of temporal task models. In Publication IV, we presented an application of AL for the EUP of a robot manipulator, where an AL technique was developed to guide novice users in tuning the parameters of robot actions.

Besides proposing a technical solution to the learning problem, in Publication

II we analysed the training process through the HRI lens, observing how different learning strategies influence human teachers and their view of AL robots. In particular, we observed how different types of queries and query selection strategies can be double-edged swords, with aspects that can be detrimental for the teacher. This led to the work presented in Publication III, where we presented a AL strategy that takes into account the order of its query to assist the teacher.

This chapter focuses on Publication II and Publication IV, presenting the design of queries that are human-understandable yet informative, along with the learning methods we employed. Chapter 4 will instead focus on the interaction side of AL explored in Publication II and Publication III.

3.1 Active Learning in a Nutshell

AL is a ML paradigm that allows the learning agent to participate *actively* in the training process. In the Passive Learning (PL) paradigm, the counterpart of AL, the agent learns purely by observing its environment [25]. The environment is therefore assumed to generate the training data necessary for the learning agent. LfD can be considered a form of PL, since the environment, *i.e.*, the teacher, provides the learner with all required training data, *i.e.*, the demonstrations. Conversely, in the AL paradigm, the learner can interact with the environment and perform actions that impact the generation of training data [25]. The most common type of action employed in AL is queries, *i.e.*, direct requests for specific information to the teacher.

Algorithm 1 presents a prototypical AL process where queries are selected from a pool $\mathcal{Q}$ of available queries (*pool-based* scenario), rather than from a stream of queries (*stream-based* scenario) or generated *de novo* by the learner (*query synthesis* scenario) [24]. Until a specified stopping condition is met (often associated with the exhaustion of a limited resource, like a labelling budget or the teacher's time), the learner selects queries from the pool $\mathcal{Q}$ to be made to the teacher. The core idea behind the query selection process is that, given the current understanding of the problem, *i.e.*, the current model θ, some queries better inform the learning process than others. Instead of randomly selecting

Algorithm 1 Prototypical pool-based AL algorithm.

Input: Query pool $\mathcal{Q}$, initial model θ, utility score U, teacher, stopping condition
Output: Trained model θ

1: **repeat**
2: $q^* \leftarrow$ select query from $\mathcal{Q}$ according to U and current model θ
3: $r^* \leftarrow$ make query q^* and obtain answer from teacher
4: integrate $\langle q^*, r^* \rangle$ in model θ
5: **until** the stopping condition is met

queries, it is therefore beneficial to select queries that facilitate learning as efficiently and as economically as possible. Thus, queries are ranked with utility scores based on the concept of information gain, helping the learner answer the question *"What query should I make to maximize my learning opportunities?"* These scores operationalize how much each query is expected to contribute to the improvement of the current model. Query scores are often computed by analysing, for instance, the current labelling uncertainty of elements in $\mathcal{Q}$ (as in *Uncertainty Sampling* [54]) or the expected reduction of labelling errors that making queries would bring to the model (as in *Expected Error Reduction* techniques [103]). As queries are made and the related answers are obtained from the teacher, information from the query-answer pairs is integrated into the model to improve it and inform the next query selections. The ability to reason over the current model and steer the learning process accordingly through the selected queries offers AL agents an advantage over PL agents, which have no control over new labelled data becoming available. This allows AL agents to learn efficiently with less labelled data [22, 102].

Beyond the simple AL pipeline summarized here, AL techniques present myriad complexities and differences. Since the seminal work of Angluin on membership queries [22], AL has been studied in-depth by the ML community and applied to both classification and regression problems, providing structure for the training of a variety of ML models and augmenting other approaches like RL [25, 102] and IRL [104]. This dissertation focuses on robotics applications of AL with humans-in-the-loop. For an extensive review of the ML literature, the reader is referred to [24].

3.2 Active Robot Learning from Human Teachers

In the past two decades, the robot learning community has successfully adopted AL approaches. The main motivation behind the use of AL is its ability to actively gather information in a sample-efficient manner. On a more practical level, AL approaches have essentially been used to solve robotics problems in three ways: (i) in a standalone manner, to solve classification and concept learning problems [21, 44, 45, 51, 52, 105, 106, 107], (ii) as a refinement tool augmenting LfD approaches [46, 108, 109, 110] and IRL [48, 49, 50, 111, 112, 113] techniques, and (iii) to guide the exploration aspect of RL approaches [114, 115].

This section surveys the robot AL literature, focusing on works where humans are part of the learning loop and discussing the design of queries and their integration into various learning methods.

3.2.1 Query Design

Queries and the related answers are the vehicle of information between the learner and the teacher. Their design must satisfy two, often opposing, principles.

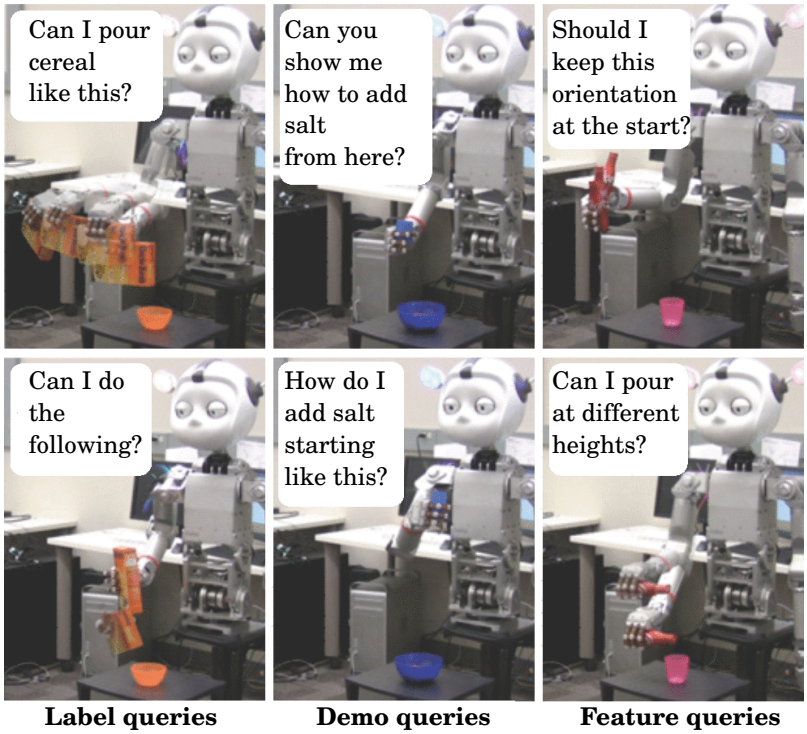


Figure 3.1. Examples of queries for a robot learning manipulation tasks from [46]. © 2012 IEEE.

On one hand, queries must convey the necessary information to tackle the learning task and be compatible with the robot's internal representation of it. On the other hand, queries must be understandable to the human teachers to help them offer their task knowledge as effectively as possible. Careful query design is therefore required to obtain maximum benefit from the combined efforts of learner and teacher.

While ML research often categorizes AL approaches based on *how* and *from where* queries are selected, AL research in robotics focuses more on the nature of a single query, investigating how different query types can be adopted to solve robotics problems. In [46], Cakmak and Thomaz propose a threefold categorization of robot query types: *label queries*, *demonstration queries*, and *feature queries*. Figure 3.1 shows examples of each query type for a manipulation task. An important feature of these queries is the possibility for the robot to use its embodiment to complement queries, referring to query elements such as joint configurations or whole demonstrations that cannot be expressed or summarized in an effective manner with natural language alone.

Label Queries

Label queries, the most commonly used query type in the literature, are requests for labels (or any other variable of interest) for selected unlabelled samples. Best suited to classification problems, label queries follow templates ranging from

"Does sample X belong to class Y?" to *"What class does sample X belong to?"*, with a continuum of variations in between, such as *"Does sample X belong to class Y_1 or Y_2?"* Label queries have mainly been used to solve various classification problems [21, 44, 51, 106, 107, 114] and for learning robot policies, in combination with LfD [15, 108] and IRL approaches [104, 111, 112, 116].

While straightforward to use when dealing with discrete concepts such as discrete actions or class labels, label queries are less suited to regression problems. When the variables of interest are continuous or multi-dimensional, such as the joint configuration of a manipulator, label queries become difficult to process, evaluate, and answer for the human teacher. When such queries are required, robots can use their embodiment to avoid this problem and query the user by *showing* a concept that is complex to express otherwise. In Publication IV, we exploited the robot's embodiment to make queries about continuous variables, such as end-effector velocities and collision thresholds. When the robot's embodiment cannot be leveraged, another solution is to replace the continuous variables in queries with discrete semantic labels, as in the definition of rules of fuzzy control systems [117]. We adopted this solution in Publication II, where we avoided the use of probability values in label queries by overlaying the probability simplex with natural language concepts based on frequency adverbs.

Another design challenge for label queries stems from the difficult interpretation of negative answers. The label queries in Figure 3.1 highlight the problem: if the teacher gives a negative answer, how should the robot interpret it? Was the entire motion wrongly executed or only a fraction of it? How far was it from being correct? To avoid this credit assignment problem [46], query selection strategies can be modified to prioritize questions that are expected to receive a positive answer, accommodating, at the same time, the human tendency of teaching through positive examples [43, 118]. Alternatively, label queries and their expected answers can be augmented to include information that helps the integration of negative feedback into the model [45]. Following this reasoning, we proposed the use of directional answers in Publication IV, allowing the users to provide meaningful feedback if the robot's execution was not as desired.

Demonstration Queries

With demonstration queries, the robot requests a demonstration from the teacher with additional constraints imposed specifically to improve the current model. As shown in Figure 3.1, a robot could request a demonstration with, for instance, a specific starting pose, a different goal pose, or with constraints on the orientation of the end-effector. Having parallels with the traditional AL problem of *Active Class Selection* [119], demonstration queries can also be seen as an intermediate method between label queries and full LfD demonstrations, where the teacher is more constrained than with LfD (the demonstration must respect the imposed constraints) but less restricted than with label queries. Demonstration queries are rarely adopted in the literature, with notable exceptions in [51, 107, 120]. Related to demonstration queries are incremental LfD ap-

proaches, where aspects of the demonstration interface are adjusted during the collection of corrective demonstrations [121, 122, 123]. These LfD approaches, however, alter the demonstration interface mostly to improve the comfort of the teachers rather than to enforce informativeness of the demonstrations.

Feature Queries
With label and demonstration queries, the AL agent learns from instances of the learning target. By contrast, with feature queries, the agent learns directly about the input features of the underlying model [124, 125]. For example, a robot could ask whether the end-effector speed of a robot motion is a good indicator for deciding about its safeness or whether the height above a table is an important feature for pouring motions (as shown in Figure 3.1). The main disadvantage of feature queries, however, is that they require the queried features to be understandable to the user. Furthermore, queried features must be strong indicators for the learning task (*i.e.*, their contribution should be meaningful) for the teacher to provide relevant feedback [46]. These requirements preclude the use of feature queries when automatic feature extraction techniques are used, although human-driven feature selection methods have been recently proposed [126]. As for label queries, robot embodiments can be used to express complex features that would be harder to express otherwise, provided that appropriate methods to showcase such features can be devised.

Comparison-based Queries
In addition to the query types presented in [46], recent work has used *comparison-based queries* to elicit preferences from the teacher. These queries, also known as *rank queries* in the AL literature [127], ask the teacher to express preferences between two [48] or more instances [50, 113] of the learning target. Comparison-based queries are beneficial when it is difficult for the teacher to evaluate or provide feedback on a single isolated instance, as is the case for the IRL scenarios presented in [48, 49, 50, 113]. Furthermore, comparison-based queries can also be used to learn about the relative importance of features, as shown in [49].

3.2.2 Learning from Queries

As introduced in Section 3.1, learning from queries requires two steps: the selection of the query to be made, and the consequent integration of its answer into the model. Query selection strategies are numerous, with their advantages and disadvantages depending on the nature of the learning task at hand. While the reader is referred to the extensive review presented in [24, Chapter 7], it is worth restating here that all such strategies follow the same underlying principle: the AL agent must select its queries in order to learn in the most efficient and cost-effective manner by evaluating the available queries against the current model. In our work, we adopted two query selection strategies: *Uncertainty Sampling* [54] in Publication III and Publication IV, and variations

of the *Expected Error Reduction* technique [103] in Publication II and Publication IV.

Query selection strategies impose requirements on the model being learned. First, the AL agent must be able to reason on the predictions of the current model and possibly analyse the uncertainty related to these predictions. Second, to make the most informative queries at each moment, the AL agent must be able to reason on the latest and most updated model; thus, it is beneficial to update the models after each query. This makes probabilistic models that can be learned incrementally as new observations become available particularly suited to AL applications. Finally, a requirement especially relevant to the case of robots learning from humans is that both the selection phase and the model update must be sufficiently fast to allow smooth interaction. The interaction offers interesting opportunities for designers of AL robots, as computationally expensive operations like query selections can be performed while robots perform other time-consuming actions, like motions and verbal communication.

The next sections build on this analysis of query types and requirements for AL systems and discuss the choices made in Publication II and Publication IV.

3.3 Active Robot Learning for Temporal Task Models

In order to overcome the major weakness of LfD approaches discussed at the beginning of this chapter, in Publication II we proposed a hybrid AL and LfD framework for the learning of temporal task models. Similar to the work presented in [108], the goal was for a robot to model a temporal task, *i.e.*, a sequence of discrete actions performed by the user to achieve a certain goal. Compared to earlier work focusing on classification and concept learning problems [12, 44, 46, 105, 106, 114], the temporal nature of the problem posed extra challenges for query design. The main contributions of Publication II are the design of queries expressed in natural language that are constrained to the context of the demonstration performed by the user, their integration into the learning pipeline along with demonstrations, and the in-depth study of the effects of different query selection strategies on the teacher-learner interaction.

To model user preferences regarding actions and their relative order in the target task, we adopted a MC representation, with each state representing one of the available discrete actions in the action set $\mathcal{A}$. The MC was parametrized as $\theta = \{\boldsymbol{\pi}, T\}$, where $\boldsymbol{\pi} \in \mathbb{R}^{|\mathcal{A}|}$ are the starting probabilities and $T \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}$ is the transition matrix, with $t_{ij} = p(a_s = a_j \mid a_{s-1} = a_i)$ being the probability of performing a_j at time step s given that a_i was performed in the previous step. Once trained, such a model can be used to predict the user's future actions based on their previous action, in order, for instance, to provide user-specific assistance in a collaborative task.

Algorithm 2 summarizes the proposed learning approach. The robot, while observing the teacher demonstrating the task, asks questions expressed in

Algorithm 2 Hybrid AL-LfD framework proposed in Publication II

Input: Action stream $\{a_1, \ldots, a_n\}$, Action set $\mathcal{A}$
Output: User preference model MC

while user provides demonstrations **do**
  initialize a new demonstration D
  while user performs action a **and** $a \neq$ *end action* **do**
    $D \leftarrow$ attach action a to D and passively update model
    $q^* \leftarrow$ select most informative query [see Equation 3.1 and 3.2]
    $r^* \leftarrow$ make query q^* and obtain answer
    MC $\leftarrow$ update with query-answer pair $\langle q^*, r^* \rangle$
  end while
end while

natural language about the ordering and probability of performing certain actions after others. This section discussed the query design, the model update, and the query selection strategies adopted in Publication II.

3.3.1 Frequency and Disambiguation Queries

For the query design, we proposed two types of label queries applicable to the learning of both the $\boldsymbol{\pi}$ and T parameters of the MC: Frequency Queries (FQs) and Disambiguation Queries (DQs). We designed these queries to aid the human teacher in two ways. First, the queries avoid direct references to probabilities or the underlying probabilistic representation of the task. Second, as the teacher answers the robot's queries while demonstrating the task, query generation is constrained to the context of the last performed action.

FQs are label queries that obtain the user's preference about the ordering of an action pair, $\{\mathbf{a_{pre}}, \mathbf{a_{post}}\}$. FQs use a set of frequency adverbs $\mathcal{F}$, such as *never*, *sometimes*, and *always* in the following template

FQ: *"After* $\boxed{\mathbf{a_{pre}}}$*, do you* $\boxed{\textbf{freq}}$ $\boxed{\mathbf{a_{post}}}$*?"*.

For the reasons stated in Section 3.2.1, the use of frequency adverbs replaces the numerical values of the model's parameters, avoiding impractical queries like *"After* $\mathbf{a_{pre}}$*, do you* $\mathbf{a_{post}}$ *with probability equal* 0.9*?"*

DQs are comparison-based queries that obtain the preferences of the user with respect to a pair of actions $\{\mathbf{a_{choice1}}, \mathbf{a_{choice2}}\}$ after the execution of another action $\{\mathbf{a_{pre}}\}$. DQs follow the template

DQ: *"After* $\boxed{\mathbf{a_{pre}}}$*, do you prefer to* $\boxed{\mathbf{a_{choice1}}}$ *or* $\boxed{\mathbf{a_{choice2}}}$*?"*.

DQs expect the user to reply with one of the following answers: "Either of these actions", "$\mathbf{a_{choice1}}$", "$\mathbf{a_{choice2}}$", or "Neither of these actions".

These queries address the learning of the transition matrix T. These templates were adapted to learn also the starting probabilities $\boldsymbol{\pi}$ (*e.g.*, for FQs, *"Do you* **freq** *start with* $\mathbf{a_i}$ *?"*).

As the querying process is interleaved with the teacher's demonstration of the task at hand, the choice of actions that can be placed in the query templates is restricted. More specifically, the queries either target the previous a_{s-1} and the current a_s actions performed by the teacher (FQs and DQs about the past), or at least the current action a_s in the $\mathbf{a_{pre}}$ slot (FQs and DQs about the future). These constraints prevent context switches during the training, avoiding the need for the teacher to consider other steps than the current, previous or next step in order to answer the robot's queries.

3.3.2 Learning from Demonstrations and Answers

Learning of the MC parameters is achieved by two means: the users' demonstrations and their answers to the robot queries. Both the actions belonging to the demonstrations and the query-answer pairs $\langle q, r \rangle$ become available to the robot in an incremental fashion, *i.e.*, one by one, rather than in batches. To allow for incremental learning of the MC parameters, we adopted a Bayesian approach, using a Dirichlet-Categorical model over θ [128, Chapter 3]: Dirichlet distributions $\mathrm{Dir}(\cdot|\boldsymbol{\alpha})$ act as the prior for the starting probabilities $\boldsymbol{\pi}$ and for each row of the transition matrix T, modelled with Categorical distributions $\mathrm{Cat}(\cdot|\boldsymbol{\alpha})$. Additionally, adopting a Bayesian approach allows for the handling of errors possibly present in both the demonstrations and the user's answers to the robot queries, as each observation will impact the model being learned only through the currently available priors. To update the model, each observation is, in fact, combined with the prior distribution to compute the posterior distribution by mean of the empirical counts (*e.g.*, the number of times a certain action a_j is followed by another action a_i). The current estimates of the MC parameters can be obtained as the mean or the mode of the posterior distributions.

This update scheme is immediately applicable to demonstrations: as a demonstration is a sequence of actions, each of its action transitions can be directly used to increase the matching empirical count. Information from the query-answer pairs cannot, however, be directly included in the model. Instead, we must compute the posterior distribution $\mathrm{Dir}(\cdot|\boldsymbol{\alpha}, q, r)$ given the query-answer pair $\langle q, r \rangle$ by bridging the probabilistic nature of the model update with the discrete and linguistic nature of FQs and DQs.

Borrowing ideas from psychological studies on people's perception of probabilities [129, 130], we designed the membership functions $M_{\mathrm{freq}}(p) : \mathbb{S}^1 \to [0,1]$, mapping the linguistic concept of each frequency adverb in $\mathcal{F}$ on the probability simplex. Figure 3.2 shows examples of such membership functions: as an example, highly probable events have high values of membership with $M_{\mathrm{always}}(p)$, while unlikely events (low probability) score low values of $M_{\mathrm{sometimes}}(p)$ and high values of $M_{\mathrm{never}}(p)$. Similarly, we designed the membership functions

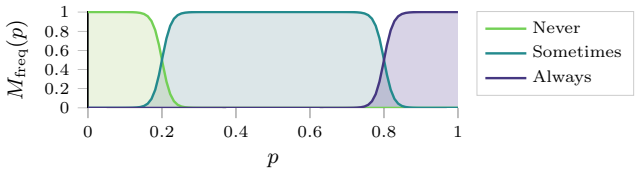
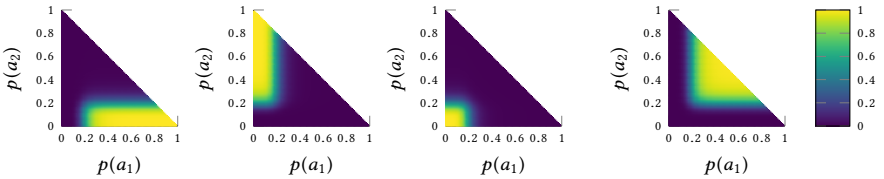


Figure 3.2. Membership Functions $M_{\text{freq}}(p)$ for three frequency adverbs. Adapted from Publication II.



(a) Choice 1. **(b)** Choice 2. **(c)** Neither actions. **(d)** Either actions.

Figure 3.3. Memberships functions $M_{\mathbf{d}}(p(a_1), p(a_2))$, one for each answer to DQs. Adapted from Publication II.

$M_{\mathbf{d}}(p_1, p_2) : \mathbb{S}^2 \to [0,1]$ shown in Figure 3.3 to match the answers to DQs.

To compute the posterior $\text{Dir}(\cdot|\boldsymbol{\alpha}, q^*, r^*)$, we used the membership functions as filters for the current prior distributions. However, since the membership functions are not distributions, no close-form update rule is available. We therefore performed the update by sampling the prior distribution $\text{Dir}(\cdot|\boldsymbol{\alpha})$ relevant to the selected query q^* and weighting the extracted samples with the membership functions, based on the user's answer r^*. The posterior distribution is then obtained by fitting a new distribution on the weighted samples with a weighted version of the Expectation-Maximization (EM) algorithm [131].

3.3.3 Query Selection Strategies

To select the most informative query $q^* \in \mathcal{Q}$, we adapted the method presented in [103] to our Dirichlet-Categorical model and our pool $\mathcal{Q}$ of FQs and DQs. This method evaluates the impact of making each query $q \in \mathcal{Q}$ on the current model by computing the expected reduction of entropy associated with each query, as

$$\begin{aligned} \Delta\mathbb{H}_q &= \overbrace{\mathop{\mathbb{E}}_{r}[\mathbb{H}(\text{Dir}(\cdot|\boldsymbol{\alpha}, q, r))]}^{\text{post-query}} - \overbrace{\mathbb{H}(\text{Dir}(\cdot|\boldsymbol{\alpha}))}^{\text{pre-query}} \\ &= \sum_r p(r|q)\mathbb{H}(\text{Dir}(\cdot|\boldsymbol{\alpha}, q, r)) - \mathbb{H}(\text{Dir}(\cdot|\boldsymbol{\alpha})), \end{aligned} \tag{3.1}$$

where $\text{Dir}(\cdot|\boldsymbol{\alpha})$ is the prior targeted by the query q, $\text{Dir}(\cdot|\boldsymbol{\alpha}, q, r)$ are the posterior distributions for each possible answer r, and $p(r|q)$ is the probability of r being the correct answer to query q. Since the answer r is not known *a priori*, Equation 3.1 computes the expectation over the answer, with the probabilities

$p(r|q)$ estimated using the current model. The query q^* is then selected with

$$q^* = \underset{q}{\operatorname{argmin}} \Delta\mathbb{H}_q. \tag{3.2}$$

3.3.4 Results and Discussion

In Publication II, we experimentally evaluated the proposed **Active** selection strategy in simulation, comparing it to

1. a **Passive** strategy, learning only from demonstrations,
2. a **Random** strategy, learning from demonstrations and by asking queries randomly selected from $\mathcal{Q}$, and
3. a **Threshold** strategy, *i.e.*, a variation of the **Active** method that avoids making queries when $\Delta\mathbb{H}_q$ is lower than a manually tuned threshold ϕ.

As an in-depth analysis of the simulation is available in Publication II, it is sufficient here to state that the proposed **Active** and **Threshold** strategies outperformed the other methods, especially for a low demonstration count, with no significant difference between these two strategies. The **Threshold** strategy, however, consistently asked fewer questions, with 59% fewer queries than the **Active** strategy during the first 10 demonstrations. While the tuning of threshold ϕ is non-trivial, we believe the **Threshold** strategy to be preferable, as it achieves results comparable to those of the **Active** strategy with fewer queries. Chapter 4 further expands on this difference from the HRI perspective, as the reduced number of queries of the **Threshold** strategy played a major role in the user study.

The computational cost of the proposed methods represents their main disadvantage, as Equation 3.1 requires the simulation of the update for all possible query-answer pairs $\langle q, r \rangle$. Consequently, the model update must be repeated a number of times in the order of magnitude of $O(|\mathcal{A}|^2)$ for DQs and $O(|\mathcal{A}||\mathcal{F}|)$ for FQs. Considering that the model update is not available in close form, we cannot expect these methods to scale well, especially for a larger number of actions. Nevertheless, in our simulation experiment and follow-up user study with a set $\mathcal{A}$ of 9 actions and 3 frequency adverbs ($|\mathcal{F}| = 3$), the query selection for the **Active** and **Threshold** strategies took 4.2 ± 0.2 s. This allowed for a smooth interaction, as these computations were performed while the robot was performing other non-computationally intensive but time-consuming actions, such as verbal feedback on the teacher's answers and action choices and various body motions (*e.g.*, nodding, shaking, pointing).

In summary, the framework presented in Publication II successfully integrated LfD and AL approaches for the learning of high-level temporal task models. To fully address the challenges of robot programming by novice users presented in Chapter 1, the framework should integrate its high-level task modelling with a

trajectory-level representation of the motions, necessary for the robot to perform each discrete action. The robot would therefore require a suitable interface to collect trajectory-level demonstrations, such as the passive interfaces presented in Section 2.1.1. In Publication II, the actions that composed the temporal task were known ahead of time, together with labels that verbally described them. To drop this assumption while maintaining the proposed querying scheme, the framework would require two extra components. First, a segmentation mechanism to extract meaningful high-level actions from the trajectory-level demonstrations would be needed [88, 89]. Second, the robot would require interaction protocols to solve a symbol grounding problem [132], in order to negotiate with the human teacher appropriate labels to verbally refer to each action segmented from the trajectory-level demonstrations. We therefore see the bridging of high-level task modelling with the trajectory-level nature of tasks as an interesting and challenging problem for future research.

3.4 Active Learning for Robot EUP

As briefly discussed in Chapter 1, in Publication IV we tackled the tuning of 1-dimensional parameters for robot EUP actions. More specifically, we targeted the tuning of parameters for which kinesthetic teaching does not provide an intuitive or sufficiently reliable interface [42], such as the speed of point-to-point linear motions or the force threshold that triggers the stopping condition of a guarded motion. While kinesthetic teaching is regularly adopted in commercially available EUP frameworks for the specification of the goal pose and via-points of robot motions [36, 37, 38, 39], the parameters targeted in Publication IV are often specified via GUI elements like sliders. To tune these parameters, users often adopt a time consuming, trial-and-error strategy, where the tuned robot action is performed several times while varying the parameter value. This process becomes even more time consuming for novice users who may not know what parameter values to try in order to complete the tuning as quickly as possible (*i.e.*, with the smallest possible number of robot action executions).

To address this problem, in Publication IV we proposed a method to aid the tuning of such parameters by having the robot select which parameter values to evaluate. The main contributions of Publication IV are the framing of the parameter value search as an AL problem, where the robot collects feedback from the user to estimate a range of feasible parameter values. We validated our active tuning approach by integrating it into a plausible EUP framework for a Panda robot manipulator, with a Domain Specific Language (DSL) composed of five parametrized robot actions, such as linear motions, gripper actions and synchronization primitives. This section presents the query design and the query selection strategies proposed in this work.

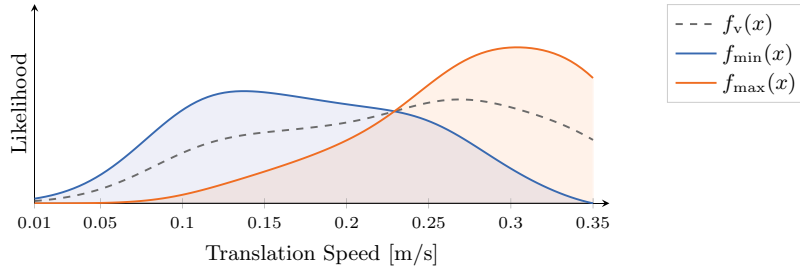


Figure 3.4. Comparison between $f_v(x)$, extracted from expert tuning for the *Translation Speed* parameter of the *Linear Motion* action, and the resulting $f_{min}(x)$ and $f_{max}(x)$ (with $n = 2$). Adapted from Publication IV.

3.4.1 Queries with Directional Answers

The goal of the tuning approach is to estimate, for each action's parameter, a range of feasible values $\hat{\boldsymbol{f}} = [\hat{l}, \hat{u}]$. First, we assume to have a prior distribution modelling the probability of parameter value v to assume a certain value x, with $p(v = x) \sim f_v(x)$. This prior is task-agnostic and can reflect how parameters are usually tuned by experts, or encapsulate safety regulations. Based on $f_v(x)$, we compute two distributions, $f_{min}(x)$ and $f_{max}(x)$, *i.e.*, the distributions of, respectively, the sample minimum and sample maximum, as

$$\begin{aligned} p(l = x) \sim f_{min}(x) &= n(1 - F_v(x))^{n-1} f_v(x), \\ p(u = x) \sim f_{max}(x) &= n F_v(x)^{n-1} f_v(x), \end{aligned} \tag{3.3}$$

with $F_v(x)$ being the cumulative distribution function of $f_v(x)$. Figure 3.4 presents examples of these distributions for the translation speed of a linear motion, with $f_v(x)$ extracted from programs authored by expert users.

With $f_{min}(x)$ and $f_{max}(x)$ acting as prior distributions for the bounds of the feasible parameter range $\hat{\boldsymbol{f}}$, the robot follows Algorithm 3 to select the parameter values to query. The robot evaluates each query $q \in \mathcal{Q}$ (*i.e.*, each possible parameter value) and executes the parametrized action with selected value q^*. Once the query receives an answer r from the user, the learner computes the posteriors of both $f_{min}(x)$ and $f_{max}(x)$. Once the query budget b is spent, the bounds $\hat{l}$ and $\hat{u}$ are computed as the mode of $f_{min}(x)$ and $f_{max}(x)$, respectively.

While in Publication II a query is an actual question expressed in natural language, in Publication IV a query is an action execution with a parameter value selected by the robot. In particular, each action execution is essentially a label query that uses the robot's embodiment, as shown in Figure 3.5.

Given the 1-dimensional nature of the tuned parameters, the answers available to the user are *directional answers*: after each action execution, the user can express whether the selected parameter is *acceptable* or whether it should be *lower* or *higher*. Allowing this kind of answers avoids the problem of negative feedback presented in Section 3.2.1.

Similar to the membership functions presented in Section 3.3.2, directional

Algorithm 3 Active Learning for Feasible Parameter Range Tuning

Input: Query pool $\mathcal{Q}$, $f_{\min}(x)$ and $f_{\max}(x)$, Query budget b
Output: Estimated feasible range $\hat{\boldsymbol{f}} = [\hat{l}, \hat{u}]$

while $b > 0$ **do**
  for all values $q \in \mathcal{Q}$ **do**
    $S_q \leftarrow$ compute query score [see Section 3.4.2]
  end for
  $q^* \leftarrow$ select query based on S_q [see Equation 3.5]
  $r^* \leftarrow$ make selected query q^* and wait for answer
  update $f_{\min}(x)$, $f_{\max}(x)$ given $\langle q^*, r^* \rangle$
  $b \leftarrow b - 1$
end while
$\hat{\boldsymbol{f}} \leftarrow [\text{argmax}_x f_{\min}(x), \text{argmax}_x f_{\max}(x)]$

answers are associated with *filter functions* that update the $f_{\min}(x)$ and the $f_{\max}(x)$ according to the query-answer pair $\langle q, r \rangle$. While more details are available in Publication IV, Figure 3.6 exemplifies the update of $f_{\min}(x)$: after a query q^* is made and the direction answer "lower" is received, the associated filter function $\lambda^-_{\phi,x_0}(x)$ is used to compute the posterior distribution $f_{\min}(x|q^*, r^*)$.

3.4.2 Query Selection Strategies

For the query selection process, we used the Expected Divergence Maximization method (**ExpDiv**), where the divergence between the prior and the expected posterior over the $f_{\min}(x)$ and $f_{\max}(x)$ distributions is used to operationalize the information gain brought by different queries. In particular, a score S_q for each query is computed as the expected divergence between the prior and posterior

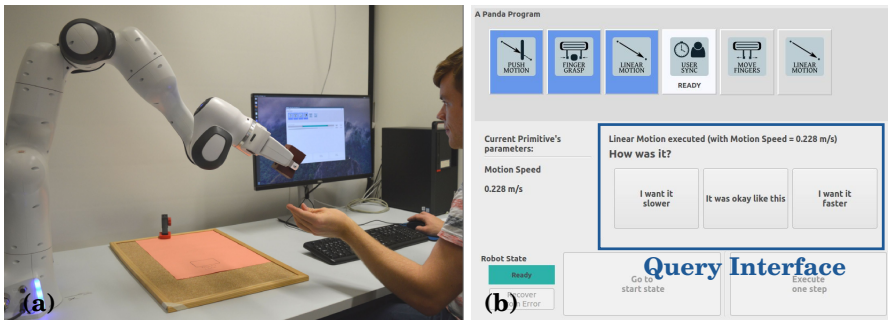


Figure 3.5. Setup (a) for Experiment 2 and 3 of Publication IV, with a Panda robot mounted on a desk and the user tuning robot programs through the Active Tuning Interface (b). Adapted from Publication IV.

for both $f_{\min}(x)$ and $f_{\max}(x)$ as

$$\begin{aligned} S_q^{\cdot} &= \mathop{\mathbb{E}}_{r}[\mathbb{JS}(\overbrace{f_{\cdot}(x|q,r)}^{\text{post-query}}, \overbrace{f_{\cdot}(x)}^{\text{pre-query}})] \\ &= \sum_r p(r|q,f_{\cdot}(x))\mathbb{JS}(f_{\cdot}(x|q,r), f_{\cdot}(x)), \end{aligned} \tag{3.4}$$

where $\mathbb{JS}$ denotes the Jensen-Shannon Divergence [133]. Queries are then selected as

$$q^* = \operatorname*{argmax}_q \{S_q^{\min}, S_q^{\max}\}, \tag{3.5}$$

i.e., the value q^* that is expected to provide the most information to either posterior distribution. The score presented in Equation 3.4 is conceptually similar to that of Equation 3.1 used in Publication II. Both scores allow the robot to reason on the current model by (i) predicting the user's answer r and (ii) evaluating the effect of the query on it. Similar to the active strategies presented in Section 3.3.3, the **ExpDiv** strategy has a complexity of $O(|\mathcal{Q}|^2)$. However, as was the case in Publication II, the time required by the query selection is negligible compared to other actions performed by the robot, *i.e.*, executing parametrized actions.

3.4.3 Results and Discussion

In Experiments 1 and 2 presented in Publication IV, we evaluated **ExpDiv** against a **Random** baseline (with complexity $O(1)$) and **Split**, a simpler strategy that selects q^* by finding the value that "splits in half" either the current $f_{\min}(x)$ or $f_{\max}(x)$ prior (with complexity $O(|\mathcal{Q}|)$). This strategy behaves as *Uncertainty Sampling* [54], where the most uncertain instance is queried regardless of the information gain it is expected to provide. Alternatively, **Split** can be seen as a weighted version of a binary search over the priors. While the reader is again referred to Publication IV for a full analysis of Experiments 1 and 2, in short **ExpDiv** performed better than **Random**, while the differences between **ExpDiv** and **Split** were negligible (despite **Split** being, in general, a less powerful strategy).

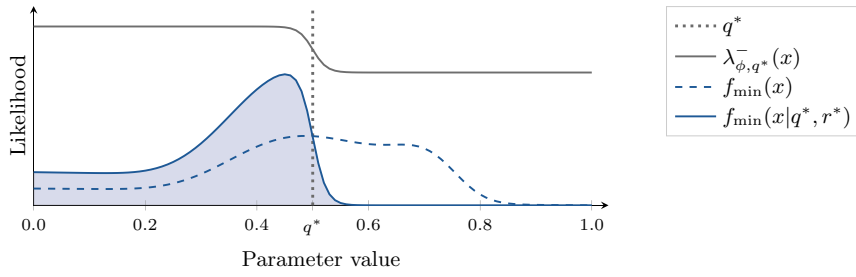


Figure 3.6. The posterior distribution $f_{\min}(x|q^*,r^*)$ is obtained by updating the prior $f_{\min}(x)$ through $\lambda^-_{\phi,x_0}(x)$, after query q^* and answer r^* = "lower". Adapted from Publication IV.

To explore the usability of the proposed active tuning approach, we conducted a user study where eight novice users tuned the parameters of a robot program. We compared the tuning interface shown in Figure 3.5.b, embedding a version of **ExpDiv**, against a standard EUP interface using GUI sliders, for the tuning of a robot handover program. The study showed how the participants were able with the active tuning interface to tune feasible parameter ranges that are closer to the range programmed by experts. Furthermore, the tuning procedure with the active tuning interface was faster than with the passive interface, with an average of eight minutes against the 13 minutes of the passive interface.

While promising, the proposed tuning approach nevertheless has room for improvement. Typically, EUP interfaces allow the user to decide which parameters to tune and whether to tune several of them at the same time. Our active approach, by contrast, tunes each action parameter separately, in sequence, spending on each of them a fixed query budget b. One improvement to this strict tuning scheme would be to have the AL algorithm decide on what parameters the query budget b should be spent within the tuned action. This extra step would require the query selection procedure of Algorithm 3 to simulate the update for each parameter of the current action, increasing, in turn, the computational costs. We nevertheless believe the computational costs to still be reasonable, as the number of 1-dimensional tunable parameters per robot action is usually small by design.[1]

Furthermore, by tuning each parameter separately, our approach does not capture the possible correlations between parameters within the same action. While the simultaneous tuning of multiple parameters is likely to further speed up the tuning process, it would also require the modelling of the relationship between an action's parameters in the prior distributions $f_{\mathrm{v}}(x)$, requiring, in turn, more training data than the current model. Tuning multiple parameters at the same time would also impact the query design. It is unclear whether users would be able to discern the effects of different parameters on the displayed robot action if their values were simultaneously changed. Consequently, it would be interesting to study whether users would still be able to give directional answers or if a simpler query design (*e.g.*, binary label queries such as "was the action execution acceptable?") should be used instead.

The proposed tuning approach operated on a DSL modelled after commercially available EUP frameworks, where parameters that cannot be specified via kinesthetic teaching, such as motion speeds and thresholds, are usually specified through GUI elements. Our approach can however tune parameters beyond the one included in the adopted DSL, as long as appropriate methods to communicate such parameters to the user are provided. As an example, a motion action whose goal pose can be specified in different frames of reference (attached to relevant objects in the environment) could be tuned as long as the robot can effectively refer to these frames. The robot could, for example, directly communicate the

[1] In the proposed DSL, robot actions had either one or two tunable parameters, as described in Table 1 of Publication IV.

selected frame verbally or with pointing gestures, or indirectly communicate it by performing the motion with respect to it (using the embodiment to make the query, as discussed in Section 3.2.1). We therefore believe the design and study of effective methods for AL robots to make more complex queries to be an interesting avenue for future research.

3.5 Discussion

This chapter presented a brief overview of the AL literature for robot applications with humans-in-the-loop, along with the main contributions to the topic from Publication II and Publication IV. While different in terms of applications and adopted methods, both our studies on AL share the same motivation: to allow robots to become an active part of the training process in order to deal with the possibly suboptimal teaching of their human instructors. We focused, in particular, on the problem of designing AL robots with queries that are understandable by humans yet informative for the chosen learning methods, showing how interfacing humans and AL systems often adds extra layers of complexity to the learning pipeline.

While the query designs adopted in Publication II and Publication IV clearly took account of the human nature of the teachers, the query selection strategies presented were only aimed at the typical AL goal, *i.e.*, to efficiently learn with as few questions as possible. These query selection strategies assume teachers to be an ever-present, infallible source of information, often conveniently referred to as *oracles*. Nevertheless, this assumption, surprisingly common in the AL literature, is rarely met in real scenarios. The next chapter addresses this issue and explores the interaction aspect of AL robots, presenting observations from the user study of Publication II and the work of Publication III, where we investigated the impact of different query selection strategies on human teachers and their response times, error rates, and effort required to answer the robot's questions.

4. Active Robot Learning from Humans: the Interaction Perspective

Most research investigates AL in a traditional ML manner, focusing on the learning performance given by, for example, different query types or query selection strategies. An alternative is to investigate the training process of AL as the interaction between two actors: the *learner* and the *teacher*. On an abstract level, the learner is an autonomous system trying to solve a task in a data-driven fashion by making queries to its labelling source, *i.e.*, its teacher. In practice, the nature of active learners varies widely: from the web application scouting one's music tastes with as few questions as possible, to the lawn mower robot asking whether the tulips should be cut down or neatly avoided.

Analysing AL systems from the interaction perspective raises interesting research questions about the role of the teacher in the training. As indicated in the previous chapter, most AL research considers teachers to be *oracles*, *i.e.*, infallible and ever-present labelling sources. Nevertheless, recent work has begun to consider teachers who are not oracles, developing AL approaches that can handle noisy teachers [49], several disagreeing teachers [134] and even teachers who are not always available [109]. Another line of research, mostly conducted in the field of HRI, has investigated, instead, the effects of AL robots on their teachers, with work analysing the impact of queries and selection strategies on the quality of teaching [12, 44, 45, 46, 135] and people's perceptions of AL robots [21, 44, 46].

Following this line of research, in Publication II we conducted a user study to investigate the effects of three query selection strategies on the teacher-learner interaction and on the teachers' perception of AL robots. We observed how different queries and selection strategies can influence the training process and the effort required by teachers. Based on the observations from this study and the available literature [43, 44, 46, 136, 137], we hypothesized that standard AL selection strategies may not be always optimal when interacting with real teachers, *i.e.*, teachers who can make mistakes while answering questions and can become tired or distracted during the training process.

In Publication III, we therefore tested the effects of a traditional query selection strategy on real teachers, studying how the ordering of queries in an information-gathering problem can affect their error rates and response times.

Inspired by the memory retrieval mechanism of the Adaptive Control of Thought–Rational (ACT-R) model [55], we then proposed a non-performance-driven query selection strategy that minimizes the difference between consecutive queries and conducted a user study to compare it to the traditional strategy. This chapter presents observations on the interactive nature of AL robots from the user study of Publication II and, in greater depth, from the work of Publication III.

4.1 Interacting with Active Learning Robots

In Publication II, we conducted a user study (within-subject design) to investigate the three query selection strategies presented in Section 3.3.4, namely the **Active**, **Threshold**, and **Random** strategies. During the interaction, 18 participants interacted with a NAO robot, *Nemo*. The participants taught the robot the steps required for the preparation of sandwiches by (i) demonstrating such steps and (ii) by answering the robot's questions. The teaching, which followed the procedure presented in Algorithm 2, involved the participants showing each action required by the recipe. After each teacher action, the robot asked a question q^*, selected using one of the three aforementioned strategies. The teacher could either reply with one of the answers expected by the query template (see Section 3.3.1) or with "I don't know", triggering no model update.

After receiving the answer r^* to the selected query q^*, the robot provided verbal feedback to the teacher, similar to the non-verbal reactions proposed in [21]. In particular, if the obtained answer r^* was the expected answer, *i.e.*, $r^* = \text{argmax}_r\, p(r|q^*)$, the robot uttered confirmation expressions such as *"I knew it!"* or *"I was expecting that"*. Otherwise, the robot replied with surprised utterances such as *"Oh really?"* and *"Good to know"*.

After each training session with a selection strategy, the participants answered the following 1–7 rating scale questions:

1. How well do you think Nemo learnt the recipe (in percent)? (1 = *0%*, 4 = *50%*, 7 = *100%*)

2. While showing the recipe, was it clear to you if Nemo was learning the recipe? (1 = *Not clear at all*, 7 = *Extremely clear*)

3. Were Nemo's questions bothering or distracting you from your task? (1 = *Extremely distracting*, 7 = *Not bothering at all*)

4. How easy was it to teach Nemo the recipe? (1 = *Extremely difficult*, 7 = *Extremely easy*)

5. How in context were Nemo's questions with respect to your recipe steps? (1 = *Completely out of context*, 7 = *Extremely in context*)

Each question included an optional "Why? Please explain" question. Table 4.1 presents the descriptive statistics for the score of each questionnaire entry. As

the reader can find the statistical analysis of the results in Publication II, this section focuses on the main observations from the user study, separating the effects of different query selection strategies in two categories: effects on the teachers' perception of the robot and effects on the teachers themselves.

4.1.1 Effects on the Teacher's Perception of the Robot

While we expected the participants to identify the **Random** strategy as the worst performing strategy, we did not expect the participants to find differences between the performance of the **Active** and the **Threshold** strategies, as both strategies operate on the same principle. Nevertheless, the **Threshold** strategy was perceived as the best performing approach, with 12 participants preferring it over the **Active** (4 preferences) and the **Random** (2 preferences) strategies.

Entering the study, we expected the participants to assess the robot's learning progress through the selected questions and the feedback to their answers. By exposing current knowledge gaps, the queries act as a transparency mechanism, indirectly exposing the inner working of the selection strategies and their quality [21, 44]. The participants perceived the questions of the **Active** and **Threshold** strategies as *informative* and *clever*, while the queries of the **Random** strategy were described as *random* and *irrelevant*. However, surprisingly few participants mentioned paying attention to the robot's feedback to their answers, questioning the usability of such an indirect transparency mechanism.

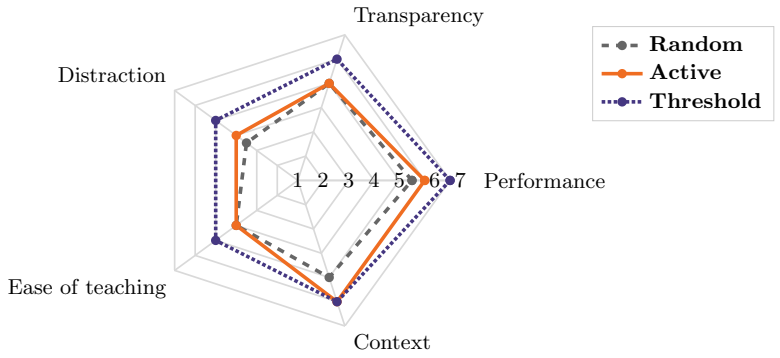
Further analysis of the participants' feedback showed that the feature that resulted in the **Threshold** strategy being the preferred strategy was its parsimonious use of questions. As the **Threshold** strategy asked an average of 29% fewer questions than the **Active** and the **Random** strategies, the participants interpreted the reduction of queries over time as a sign of learning and utilized it to inform their decision about when to stop teaching. Since in this study the participants could not test the robot's knowledge with user-initiated queries as in [21], the reduction in the number of queries made by the **Threshold** strategy became the strongest indicator of progress and performance, thus negatively affecting the participants' perception of the **Active** and **Random** strategies.

The participants also identified patterns in the robot's queries and interpreted them in different ways. For example, the repetition of questions was seen by some participants as a sign of poor learning, while other participants saw it as tool for the robot to consolidate its knowledge. When the robot asked questions about actions that were either yet to be performed by the user or simply unusual, some participants interpreted this as a legitimate learning strategy, allowing the robot to rule out options. These questions, particularly favoured by the **Active** and **Threshold** strategies due to their maximization of information gain, were, however, considered useless by other participants, who often commented that the robot lacked common sense.

The participants' attempts at interpreting the query selection strategies suggest that people naturally want to understand how AL robots learn, and they

Table 4.1. Questionnaire scores (1–7) for the query selection strategies: first quartile (Q1), **median**, third quartile (Q3). The plot graphically compares the median ratings. Adapted from Publication II.

Questionnaire scores	**Random**	**Active**	**Threshold**
Performance (7 = best)	4,**5.5**,6	5,**6**,6	6,**7**,7
Transparency (7 = clearest)	3,**5**,6	5,**5**,6	6,**6**,7
Distraction (7 = least distracting)	3,**3.5**,5	2,**4**,5	4,**5**,5
Ease of teaching (7 = easiest)	2,**4**,5	2,**4**,5	5,**5**,6
Context (7 = most in context)	3,**5**,6	4,**6**,6	5,**6**,6



often do so by applying familiar learning patterns to the behaviours of robots. However, these interpretations seldom align with the actual algorithmic reasoning or principle behind the robot's choices and could cause problems related to over-reliance and mistrust [13]. Chapter 5 builds on these observations and further discusses the need for effective transparency mechanisms.

4.1.2 Effects on Teachers

The effects of the selection strategies on the teacher side of the interaction were observed from the participants' self-reported ease of teaching and distraction, summarized in Table 4.1. The **Threshold** strategy was considered easier to teach compared to the other two strategies, with no differences observed between the **Active** and **Random** strategies. This, again, was a consequence of the ability of the **Threshold** strategy to avoid asking questions, with the participants disliking the constant flow of questions produced by the other two strategies.

While no differences were observed regarding the Distraction score, seven participants expressed a preference for the interaction scheme adopted in the study, with the robot asking questions during the demonstration of the task. An equal number of participants, however, suggested an alternative interaction scheme with temporally separated queries and demonstrations to reduce distraction. The preference for having two temporally distinct phases could also be linked

to the desire of the teachers of AL robots to be in control of the interaction, as observed in [44]. Nevertheless, we believe that, given the temporal nature of the task, separating queries from demonstrations would make answering more complicated for teachers, forcing them to recollect what had previously occurred in order to answer the robot's questions.

The participants answered "I don't know" to only 1.6% of the robot's questions, with all three selection strategies obtaining a high Context score. While the study did not compare different query designs, we believe these results to be a good indicator of the success of the query design presented in Section 3.3.1.

Finally, we analysed the participants' feedback for comments on specific types of queries. The participants remarked on the tendency of the **Active** and **Threshold** strategies (but not the **Random** strategy) to select questions that expected a negative answer and that such queries were unintuitive and hard to answer. This observation aligns with previous research on people's bias toward teaching through positive examples [43, 118].

In summary, the user study of Publication II helped reveal how AL robots, with their queries and selection mechanisms, can influence their human teachers. While self-reported measures from the participants were collected, the nature of the study did not allow for the analysis of other measures, such as error rates, response times, and training performance. Nevertheless, these observations, together with the available literature [43, 44, 45, 46, 137], were instrumental for the work of Publication III, presented in the next section.

4.2 Memory Effort-aware Active Learning

As already stressed in Section 3.1, the main advantage of AL systems lies in their query efficiency, *i.e.*, their ability to select what to learn from and consequently reduce the required number of labelled samples. While AL's query efficiency is often, and reasonably, associated with a reduction of effort for the human teacher [138, 139, 140], this claim has never been supported by a direct analysis of the teacher's workload. At the same time, research in HRI, including the work of Publication II, has observed how human teachers can find the questions of AL robots difficult to answer [43, 44, 46].

Research in the area of cost-effective AL has proposed methods that integrate the concept of labelling cost into several selection strategies [136, 141, 142, 143]. The estimated time required to answer a query is most commonly used as a proxy for the cost of a query, with works incorporating measures of labelling effort for the specific cases of form filling [138] and image annotation [144]. While acknowledging the fact that some queries are harder to answer than others, these works do not investigate the effects of traditional and cost-effective selection strategies on the teacher's workload. In Publication III, we challenged the idea that the query efficiency of AL strategies is linked to a reduction of effort for the teacher. The main contributions of Publication III are the proposal of a

query selection strategy for an information gathering problem that takes into account the flow of queries to support the teacher, and its comparison, by mean of a user study, with a traditional AL strategy. In particular, we analysed the effects of these strategies on teachers' error rates, response times, and workload. This section covers the problem statement, the proposed query selection strategies, and the main results and observations from the user study of Publication III.

4.2.1 Problem Statement

To investigate the effects of different query selection strategies, we targeted a simple information gathering problem where the agent must learn the value of an attribute a for a set $\mathcal{E} = \{e_1, \ldots, e_N\}$ of entities by making queries to a human teacher. A set of categories $\mathcal{C} = \{c_1, \ldots, c_M\}$ is provided to the learner, along with the relevance $w_{c,e}$ of each $c \in \mathcal{C}$ for each $e \in \mathcal{E}$. With this categorical information and the assumption that *entities belonging to the same category are likely to share the same attribute value*, asking a question $q_{e,a}$ about the value of attribute a for entity e reveals more than just the correct attribute value. By modelling the probability of observing attribute a given a category $c \in \mathcal{C}$ as

$$p(a = x|c) \sim f_{c,a}(x|\theta_{c,a}), \tag{4.1}$$

with $f_{c,a}(x|\theta_{c,a})$ being a distribution suiting the nature of a, the teacher's answer r to $q_{e,a}$ can act as an observation for the estimation of $f_{c,a}(x|\theta_{c,a})$. Modelling the probability $p(a = x|c)$ allows the learner to make predictions about the unseen entities through the categories $\mathcal{C}$, computing the probability $p(a = x|e)$ as

$$p(a = x|e) \sim f_{e,a}(x) = \sum_{c \in \mathcal{C}} \bar{w}_{c,e} f_{c,a}(x|\theta_{c,a}), \tag{4.2}$$

with $f_{e,a}(x)$ being a weighted mixture of $f_{c,a}(x)$ and $\bar{w}_{c,e}$ the normalized category relevances. While the interested reader can find more details about the learning procedure in Publication III, it is important here to note how making predictions about the attributes of unseen entities allows the adoption of AL query selection strategies, as explained in Section 3.2.2.

For the simulation experiment and user study of Publication III, we designed the learning scenario around the Animals with Attributes 2 (AwA2) dataset [145]. The dataset contains images of 50 different mammals and their description using 85 semantic attributes. The aforementioned categorical information was, instead, extracted by exploiting the hierarchical representation of WordNet, a lexical database of English [146]. A total of 28 categories were extracted from WordNet starting from the 50 AwA2 entities, forming the Entity-Category Tree $\mathcal{T}$, partially shown in Figure 4.1. In our learning scenario, the agent learns about the attributes of these animals (representing the entity set $\mathcal{E}$) using the extracted categories $\mathcal{C}$ by selecting queries from a query pool $\mathcal{Q}$. Examples of queries the agent can ask are *"Do lions have horns?"* and *"Do giraffes eat meat?"*. Although of little relevance to real world robotics problems,

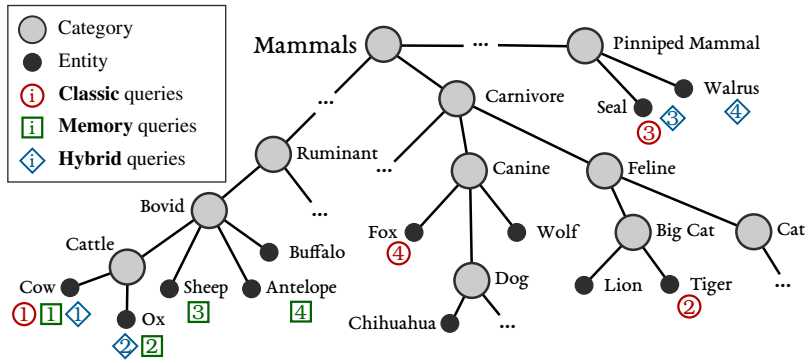


Figure 4.1. Part of the Entity-Category tree $\mathcal{T}$ and representative query flow for each query selection strategy. Adapted from Publication III. © 2019 IEEE.

the AwA2 learning scenario was chosen for three reasons. First, the familiar and relatively easy topic of animal attributes allowed the participants of the user study to comfortably act as teachers. Second, the availability of the ground truth allowed for the analysis of the participants' error rates, a key measure for the study. Finally, the adoption of a well known dataset like AwA2 improved the reproducibility of the work.

4.2.2 Query Selection Strategies

We addressed the aforementioned learning problem with three query selection strategies: a **Classic**, a **Memory**, and a **Hybrid** strategy. The **Classic** strategy used *Uncertainty Sampling* [54], selecting queries based on the entropy of their current prediction as

$$C_q = \mathbb{H}(f_{e,a}(x)). \tag{4.3}$$

In practice, the **Classic** strategy selects the query about which the current model is the most unsure [24, Chapter 2]. Selecting the most uncertain query in $\mathcal{Q}$ causes the **Classic** strategy to scatter its questions as far as possible on the tree $\mathcal{T}$, as shown by its representative query flow in Figure 4.1. In Publication III, we hypothesized that the context switches caused by these efficient queries would increase the effort required by the human teachers, causing them to answer slowly and be prone to errors, consequently hindering the training process.

We therefore proposed the **Memory** strategy, using a query selection that is teacher-aware rather than performance-driven. Inspired by the declarative memory model of ACT-R [55], we designed the **Memory** strategy to select questions that minimize the distance between consecutive queries. This strategy is based on the concept of associative strength between memory chunks, which posits that chunks of memory that are frequently associated require less effort to be retrieved [55, 147, 148]. Since we obviously lacked access to the true associations of chunks in the teacher's memory, we used the structure of $\mathcal{T}$ as

an approximation of it. To select its queries, the **Memory** strategy therefore maximizes

$$M_q = e^{-\delta d(e,p)}, \tag{4.4}$$

where $d(e,p)$ is the distance, defined as the number of edges in $\mathcal{T}$, between the entity e, target of query q, and the entity p, target of the *previous query*, with δ being a scale parameter. The **Memory** strategy therefore groups its queries using the structure of $\mathcal{T}$, as shown in Figure 4.1.

Finally, we proposed a **Hybrid** strategy, a combination of the **Classic** and the **Memory** strategies that maximizes

$$H_q = \sigma C_q + (1-\sigma) M_q, \tag{4.5}$$

with $\sigma \in [0,1]$ controlling the trade-off between the two strategies.

It is important to note that, from an AL perspective, the **Memory** strategy is not optimal, as it trades information gain for a flow of questions that is, at least based on the concept of associative strength, easier to answer. Indeed, the simulation experiment of Publication III demonstrated how the **Classic** strategy, unsurprisingly, outperforms the **Memory** strategy. We refer the reader to Publication III for the analysis of the simulation, focusing here on the results of the user study and on the effects of the three query selection strategies on the error rates and response times of the participants.

4.2.3 User Study

In the user study of Publication III, 26 participants acted as teachers for *Nemo*, a NAO robot embodying the three strategies described in Section 4.2.2 (within-subject design). The robot, shown in Figure 4.2, asked the participants questions which could be answered on a keyboard[1] with "Yes", "No" or "I don't know". In particular, each participant answered questions about six attributes from the AwA2 dataset, with two attributes for each strategy following the scheme of Table 4.2. The training for each attribute lasted 40 seconds, allowing the participants to answer an average of 15 queries.

Our hypotheses entering the user study were that the **Memory** strategy would allow the participants to answer faster and with less errors compared to the **Classic** strategy. Furthermore, we expected the **Classic** strategy to require more mental effort from the teacher compared to the **Memory** strategy. Finally, we expected the **Hybrid** strategy (with σ manually set to 0.8) to obtain intermediate results with respect to the other two strategies. To test these hypotheses, we logged

1. the response times, *i.e.*, the time required to answer a question,

[1] We chose to use a keyboard as the input device instead of more sophisticated interfaces, such as voice recognition, in order to reliably observe the small differences in response time.

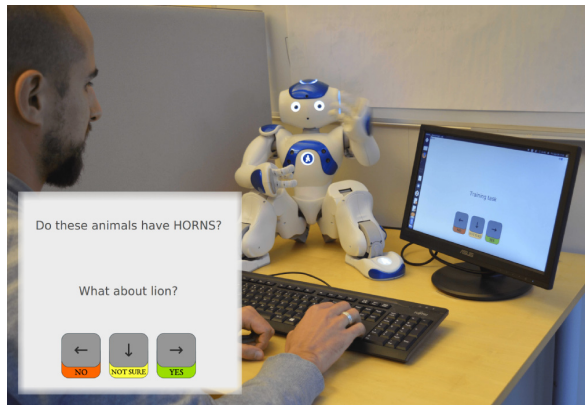


Figure 4.2. Experimental setup and example of question, as shown on the screen. Adapted from Publication III. © 2019 IEEE.

Table 4.2. Study design from Publication III, with the queried AwA2 attributes. Adapted from Publication III. © 2019 IEEE.

	Session 1	**Session 2**	**Session 3**
Visual attributes (physical features)	Do ___ have paws?	Do ___ have horns?	Do ___ have claws?
Non-visual attributes (diet)	Do ___ prefer to eat fish?	Do ___ prefer to eat meat?	Are ___ herbivore?

2. the selected queries and the participants' answers used to compute the participants' error rates.

After the session with each strategy, we administered the RAW NASA Task Load Index (NASA TLX) questionnaire [149] and a session questionnaire with the following 3 Likert statements (1 = *completely disagree*, 7 = *completely agree*):

1. The flow of Nemo's questions felt natural,
2. Nemo's strategy made my job as teacher easy,
3. Nemo's strategy was good for its learning.

Each statement included an optional "Why? Please explain" question.

4.2.4 Results and Discussion

The results of the user study, summarized in Table 4.3, did not completely align with the expectations presented in Section 4.2.3. Surprisingly, we observed the fastest response times with the **Hybrid** strategy, with an average of 0.73 s against the 0.85 s and the 0.90 s of the **Classic** and the **Memory** strategies respectively. The **Hybrid** strategy also allowed the participants to make less

Table 4.3. Results from the user study for each query selection strategy. Mean values of response times, error rates, and prediction percentages. Median of the NASA TLX scores. The reader is referred to Publication III for a complete statistical analysis of the results. Adapted from Publication III. © 2019 IEEE.

	Classic	**Memory**	**Hybrid**
Response time (RT) [s]	0.85	0.90	0.73
RT with visual attributes [s]	0.80	0.86	0.70
RT with non-visual attributes [s]	0.91	0.96	0.78
Error Rate	21.4%	19.5%	11.4%
Prediction Percentage	81.0%	51.4%	74.1%
Prediction Percentage with oracles	89.4%	55.6%	77.5%
NASA TLX scores	**Classic**	**Memory**	**Hybrid**
Mental Demand	6.5	10.5	6.0
Physical Demand	1.0	1.0	1.0
Temporal Demand	6.0	5.5	5.0
Performance	6.5	9.5	5.0
Effort	6.5	8.5	6.0
Frustration	4.0	5.0	5.0

mistakes, with only 11.4% of its questions answered incorrectly. By contrast, with the **Classic** and the **Memory** strategies, the participants made approximately double the number of errors (error rates of 21.4% and 19.5% respectively). However, the high error rate observed for the **Classic** strategy did not prevent it from obtaining the highest prediction percentage,[2] with an average of 81.0% correct predictions. Despite the lower error rate, the **Hybrid** strategy had a prediction rate of 74.1%, followed by the expected poor performance of the **Memory** strategy, with 51.4% correct predictions. To better understand the impact of the participants' errors, Table 4.3 shows also the prediction percentage in the hypothetical case of infallible users. While the **Classic** strategy still performed the best, it was also the strategy that lost the most performance (8.4%) due to participant errors.

While we observed no significant differences for the scores of the session questionnaire and for most of the NASA TLX scores, the participants feedback helped us investigate possible explanations for the response times and error rates. In accordance with the reasoning presented in Section 4.2.2, the participants commented on how the **Classic** strategy seemed to ask random questions and that this made their teaching stressful, unpredictable, and more mentally demanding.

[2]The prediction percentage is the percentage of not queried entities for which the strategy can correctly guess the attribute at the end of the training session.

The participants also recognized that the **Memory** strategy used categories to group its queries, and remarked on how this made the flow of questions natural and easier to answer. A recurring comment about the **Memory** strategy was that participants took advantage of its use of categories to anticipate future questions and related answers. However, this predictability potentially caused the participants to engage in a sort of *autopilot*, answering mechanically the robot's questions, or simply become bored by similar questions, lowering their attention and causing the unexpectedly high error rates and response times. As with the **Memory** strategy, the participants described the questions of the **Hybrid** strategy as easy to answer thanks to the grouping in categories. While some participants also mentioned the ease of anticipating future questions for the **Hybrid** strategy, one participant commented on how the slightly less predictable flow of queries made him more attentive (*"The flow seemed natural, but the slight variation of questions kept me more awake"*), a possible explanation for the lowest error rate and the fastest response time of the **Hybrid** strategy. This observation is in line with the findings of [150], where a robot asking off-topic questions was perceived as easier and more fun to work with by its users.

In summary, the user study of Publication III showed how traditional query selection strategies focused solely on maximizing information gain such as the **Classic** strategy, can make the training process difficult and stressful for human teachers, causing them to make more mistakes and answer at slower paces. The study also showed how the proposed **Memory** strategy, trading performance for ease of teaching, was a poor alternative, as it failed to speed up the training with its simple queries, and moreover frustrated the participants with its poor performance. In other words, the **Memory** strategy minimized the effort required to retrieve information from declarative memory but failed to account for other aspects of the interaction, such as frustration or user attention. This observation supports a design recommendation made in [44]: AL robots should avoid asking uninformative or unnecessary questions, as they could weaken the teacher's trust in the utility of answering them.

While in our study the **Hybrid** strategy overcame the limitations of its two components and yielded overall good results, more research is required on the factors that influence the interaction between AL robots and their human teachers. A better understanding of the training process would allow for new learning mechanisms that adapt to the preferences of the human teachers and their current state (*e.g.*, their tiredness or task expertise). In summary, while the experimental setup of Publication III is of little relevance to the real world robotics problems discussed in Chapter 1, we believe that the presented results can offer useful insights for the design of interactive robots and motivate future research on the actual usability of ML approaches for cases where real users are integral part of the learning loop.

4.3 Discussion

This chapter focused on the interaction aspect of AL robots, presenting the user studies of Publication II and Publication III. After observing how query selection strategies can affect the teacher side of the interaction, Publication III explored the concept of AL robots whose goal is not to maximize information gain but to ease the training process for their human teachers. In the user study, we nevertheless observed how neither of the two extremes, the **Classic** and the **Memory** strategies, benefited the teacher during the training process. These results suggest that a good balance of features (*e.g.*, ease of teaching, performance, and predictability) should be considered when designing AL robots, opening interesting avenues for future research on how to find this balance and identify the aspects of the training process that can perturb it.

In Publication II, we observed how human teachers naturally attempt to understand the decision making of learning robots. The transparency offered by the robot's queries and the feedback to the teachers' answers was however too indirect, resulting in the participants using other features to form their own mental models of the robot's capabilities. Leaving the interpretation of robots' capabilities and limitations to the guesswork of potentially novice users is, however, a dangerous path to follow, as misalignments and discrepancies between the robot's actual capabilities and the users' mental models of those abilities can result in over-reliance and mistrust [13]. The next chapter explores the topic of robot transparency and presents the work of Publication V on the autonomous generation of explanations for robot policies.

5. Robot Transparency through Policy Explanation

As argued in Chapter 1, the teaching of robots by human teachers should be considered a collaborative task, with the accurate sharing of knowledge, beliefs, and suppositions between the involved actors – in a word, *transparency* – being key to successful collaboration. Theodorou *et al.* characterize transparency for robotic systems as a set of mechanisms for reporting reliability and exposing unexpected behaviour and decision making [56]. Lyons describes several facets of robot transparency from an HRI perspective, arguing that transparency should target not only the robot but the interaction as a whole at different levels [13]. The previous chapters already hinted at the topic of transparency by presenting how the inner working of AL robots can be exposed either through their queries, indirectly revealing current knowledge gaps or through purposely devised mechanisms, such as the answer feedback adopted in Publication II. Furthermore, it was shown how human teachers naturally seek to understand learning robots but will possibly do so by using irrelevant aspects of the interaction if inadequate or ineffective transparency mechanisms are offered to them.

While difficult to achieve due to the substantial differences between humans and robots, transparency is of paramount importance, as it can help users trust robots and rely on them only when appropriate [58, 59]. As stricter regulations are imposed on the accountability, safety, and fairness of not only robots but autonomous systems in general [62, 151, 152] and as the popularity of black-box approaches continues to rise, the urgent need for Explainable Artificial Intelligence (XAI) and Interpretable Machine Learning (IML) is becoming clear [30, 60, 61, 63]. In order to increase the transparency of robots for their novice users, in Publication V we proposed a policy explanation method that answers *"why"* questions [153], describing in natural language how the current situation influenced the decision of which action to take. This model-agnostic method aims to provide local explanations that are (i) *robust* to small variations in the policy and (ii) *focused* on the variables that truly impacted the policy's decision. This chapter presents the working principles behind the proposed method, along with the main observations from the user study of Publication V, where we investigated the effect of the generated explanations on the user's understanding of robot policies.

5.1 Focused and Robust Policy Explanations

While the philosophical debate on what constitutes a good explanation is still open [154], in Publication V we focused solely on explanations that answer why a system took a particular action, *i.e.*, what policy the system followed [153, 155]. When generating such explanations, there is a tension between two characteristics: *interpretability* and *completeness* [30]. Interpretability is the ability of an explanation to be understood by humans. Completeness, also referred to as fidelity [156], is instead the ability of an explanation to accurately describe the underlying system. The more an explanation is complete, the larger is the number of situations in which the system's behaviour will be correctly predicted based on the explanation. While both are desirable, these two characteristics are difficult to embed simultaneously in an explanation.[1] Complete explanations can quickly become too complex to be interpretable: listing all the weights of a neural network is an example of a complete explanation that is, however, impossible for humans to understand. Furthermore, explanations that summarize entire policies may also suffer from this problem [157]. Vice versa, skewing explanations towards interpretability may oversimplify the description of the underlying decision making process, hindering the user's ability to build a faithful mental model of it. Publication V uses *local explanations*, *i.e.*, explanations that focus only on a single action or decision instead of the whole policy, producing interpretable explanations without oversimplifying the underlying decision making process.

Given a stochastic policy $\pi(\boldsymbol{x}) : X \to \mathbb{S}^M$ defined on a multidimensional continuous state space X (being a closed subset of $\mathbb{R}^N$) with a discrete action set $\mathcal{A} = \{a_1, \ldots, a_M\}$, a *comprehensive explanation* for action a taken while the system was in state $\boldsymbol{x}$ follows the template

> The system performed $\boxed{\text{action } a}$ because $\boxed{d_0 \text{ was } \gamma_0^*}$ and $\boxed{d_1 \text{ was } \gamma_1^*}$ and ... and $\boxed{d_{N-1} \text{ was } \gamma_{N-1}^*}$,

where d_i is a natural language label of the i-th dimension in X. Following the same reasoning adopted for the design of queries in Section 3.3.1, the numerical value of each dimension d_i of current state $\boldsymbol{x}$ is replaced in the explanation by a natural language descriptor γ_i^*, such as *fast* and *slow*. These descriptors are selected based on membership functions $\epsilon_i(x_i)$ that map the relevance of each descriptor over the dimension d_i. Figure 5.1.b shows three membership functions $\epsilon_0(x_0)$, mapping the relevance of three descriptors (*low*, *medium*, and *high*) over the dimension d_0. In practical terms, comprehensive explanations describe current state $\boldsymbol{x}$ in a human-friendly manner to expose what situation made the system take action a, including every dimension of the state space. For high-dimensional state spaces, comprehensive explanations

[1] A parallel can be drawn with the trade-off between completeness and interpretability of explanations and the design requirements of AL queries described in Section 3.2.1.

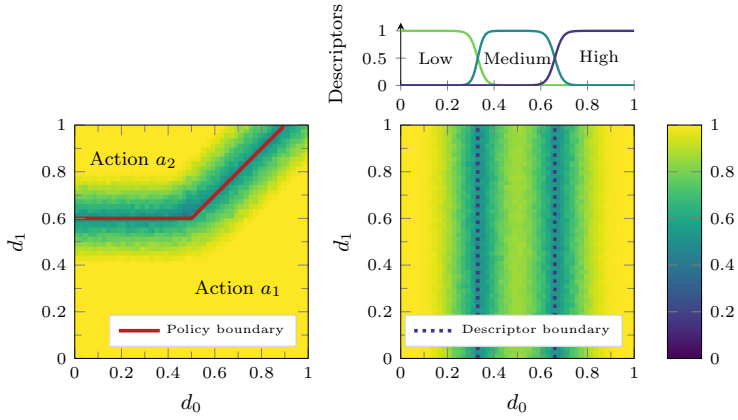
can therefore quickly become long and, consequently, difficult to understand. Moreover, comprehensive explanations can also be overly specific, including dimensions that did not impact the choice of the policy.

To address these limitations, in Publication V we proposed a model-agnostic method that generates local explanations focused on the relevant dimensions of the state space. Instead of requiring policies to be modelled as (or reducible to) Markov Decision Processes (MDPs) [157, 158] or Partially Observable Markov Decision Processes (POMDPs) [159], the proposed method does not rely on a particular policy encoding, requiring only the policy to be evaluated at sampled locations in the state space to determine the relevant dimensions. The proposed method is analogous to Local Interpretable Model-agnostic Explanations (LIME) [156], a model-agnostic method that produces non-verbal explanations for image and text classification models. Methods based on sampling, such as LIME, have nevertheless been shown to be unstable, *i.e.*, they generate different explanations for small changes in the policy's input [160]. Based on four measures computed by sampling the state space, the proposed method can exclude dimensions from the explanation where the policy is locally unstable or the verbal descriptors used to describe it are locally ambiguous.

5.1.1 Dimension Selection

For generating a *focused* and *robust* explanation for a state-action pair $\langle \boldsymbol{x}, a \rangle$, the following cases should be avoided. First, an explanation should not be overly specific, that is it should exclude dimensions that did not impact the action selection (lack of *relevance*). Furthermore, an explanation should exclude a dimension if small changes along such dimension change the policy's output (lack of policy *stability*). Similarly, an explanation should exclude dimensions of current state $\boldsymbol{x}$ that cannot be reliably described with the available descriptors (lack of state *describability*). Finally, explanations about areas of the state space described by a single descriptor that lead to multiple actions should be avoided (lack of *consistency*).

To avoid these four cases, the proposed method computes four measures for each dimension: local measures s_{msr} and d_{msr} to guarantee, respectively, stability and describability, and global measures c_{msr} and r_{msr} to guarantee, respectively, consistency and relevance. Local and global measures differ in the sampling procedure used to compute them. For local measures, v states are uniformly sampled in a hyper-sphere of radius ρ around the explained state $\boldsymbol{x}$. For global measures, the sampling is instead dimension-specific: for each dimension d_i, w values of d_i are sampled while keeping the rest of the dimensions' values at the value of current state $\boldsymbol{x}$. While allowing the explanation of black-box models, the sampling procedure exposes the method to the curse of dimensionality: as the number of dimensions of the state space increases, the number of samples v and w required to obtain good estimates of the four measures increases as well.



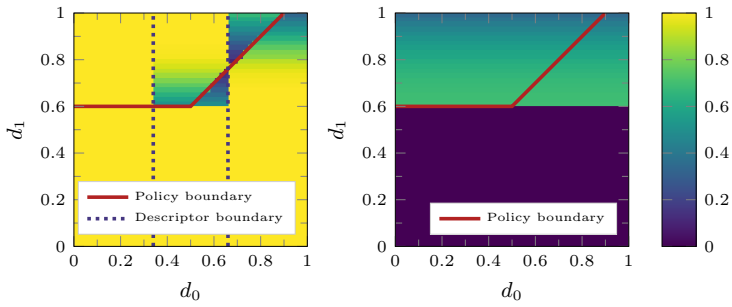
(a) Stability s_{msr} **(b)** Describability d_{msr} along d_0

Figure 5.1. Local measures of stability and describability for dimension d_0. Adapted from Publication V. © 2019 IEEE.

While the reader is referred to Publication V for the details of the computation of the four measures, their working principle is summarized here with the aid of an example: a deterministic policy with two actions defined on a 2-dimensional state space. As shown in Figure 5.1.a, areas of the state space close to the policy boundary are assigned low values of stability score s_{msr}. Similarly, in Figure 5.1.b the describability measure d_{msr} for dimension d_0 is low for states where the available descriptors overlap. It is worth noting how d_{msr} would be low also in areas of the state space where none of the membership functions of the available descriptors are high. In other words, areas of the state space that cannot be described with the available vocabulary of descriptors are assigned a low describability measure d_{msr}.

Figure 5.2.a shows the consistency measure c_{msr} for dimension d_0. For values of d_1 between 0.8 and 1, the descriptor *high* of d_0 describes areas of the state space that lead to different actions. These areas are correctly assigned a low consistency measure c_{msr}. Finally, Figure 5.2.b shows the relevance measure r_{msr} for dimension d_0. For values of d_1 lower than 0.6, any value of d_0 will result in action a_1 being taken, *i.e.*, d_0 is not relevant for the choice of action: this is captured by low values of r_{msr} in that area of the state space.

With the four measures computed for each dimension of the state space, the decision about which dimension to include in the explanation is performed by comparing each measure with suitable thresholds. In Publication V, we excluded a dimension from an explanation if any of its measures was lower than a manually defined threshold $\eta = 0.6$. Figure 5.3 illustrates the inclusion in the explanations of dimensions d_0 and d_1 in different areas of the state space, with four states marked by circled numbers. In state ①, the generated explanation is "The system took action a_2 because d_0 was *medium* and d_1 was *fast*". In state ②, instead, dimension d_0 is excluded because it was not



(a) Consistency c_{msr} along d_0 **(b)** Relevance r_{msr} along d_0

Figure 5.2. Global measures of consistency and relevance for dimension d_0. Adapted from Publication V. © 2019 IEEE.

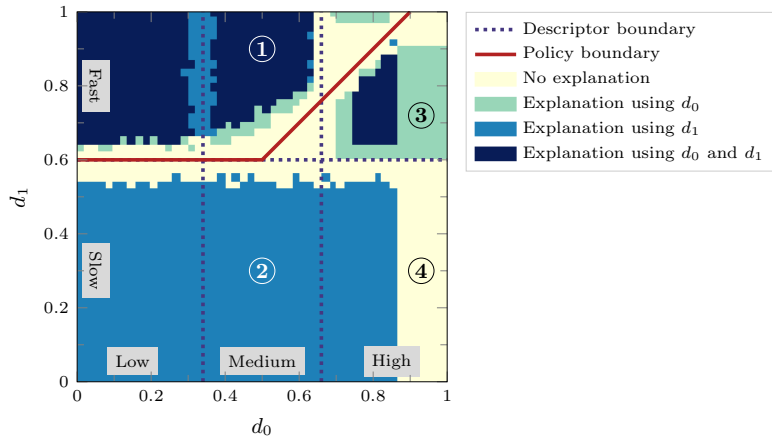


Figure 5.3. Dimensions included in the explanation of different areas of the state space by the proposed method (threshold $\eta = 0.6$). Adapted from Publication V. © 2019 IEEE.

relevant to the action selection, generating "The system took action a_1 because d_1 was *slow*" as an explanation. Similarly, in state ③, d_1 is the excluded dimensions, and the generated explanation is "The system took action a_1 because d_0 was *high*". The method also avoids explanations in proximity to the policy boundary and the descriptors boundaries, avoiding issues related to the lack of robustness [160]. Finally, for state ④, no explanation is generated, as both dimensions d_0 and d_1 are individually deemed irrelevant by the r_{msr} measure, given the particular shape of the policy boundary. This case raises an interesting question: Can combinations of individually irrelevant dimensions be relevant for an explanation? While not covered by the proposed method, we believe this case to be worth further investigation.

While the example above demonstrates the ability of the proposed method to generate explanations for black-box models, it is nevertheless difficult to perform a rigorous evaluation of the method and its numerous parameters. This is because the quality of an explanation can be ultimately evaluated only by

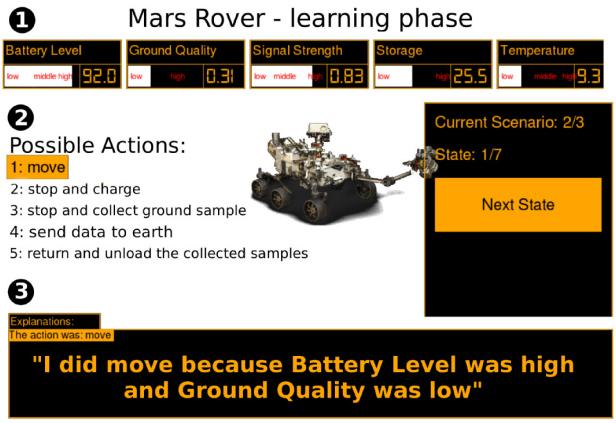


Figure 5.4. User study scenario: a Mars rover exposes to the user (1) its state space $\boldsymbol{x}$ through bar indicators and verbal descriptors, (2) action a selected by its policy, and (3) a **Focused** explanation of the current state-action pair $\langle \boldsymbol{x}, a \rangle$, using only two of the five dimensions of the robot's state space. Adapted from Publication V. © 2019 IEEE

observing how it helps its human recipients, allowing them, for example, to better predict the actions of the explained system. In Publication V, we therefore evaluated the usability of the proposed method with a user study utilizing manually tuned thresholds and sampling parameters, leaving their automatic tuning to future work.

5.1.2 User Study

The user study presented in Publication V investigated the usability of the proposed method, comparing its **Focused** explanations to the **Comprehensive** explanations presented at the beginning of Section 5.1. In this study, 18 participants interacted with two simulated rovers (a Mars and a Moon rover) through the GUI shown in Figure 5.4, learning their policy for a space exploration scenario.[2] The rovers' policy was encoded as a decision tree, with a 5-dimensional state space and five possible actions. For both policies of the two rovers, one dimension was deliberately made irrelevant (*i.e.*, its value never impacted the choice of action). The **Comprehensive** explanations included all dimensions, using the descriptors shown in Figure 5.4. By contrast, the **Focused** explanations included up to two dimensions, selected as presented in Section 5.1.1 (the same threshold, $\eta = 0.6$, was used).

Each participant interacted with both rovers (within-subject design) in two distinct phases: a *learning phase* and a *testing phase*. During the learning phase, a set of 25 state-action pairs was shown to the participants, with each pair complemented with either a **Focused** or a **Comprehensive** explanation.

[2]The scenario was purposely designed to be unusual for the participants, after we observed how, in a pilot study with a faulty vacuum cleaner robot, the participants would disregard the robot's explanations and trust their own understanding of the rather common home appliance.

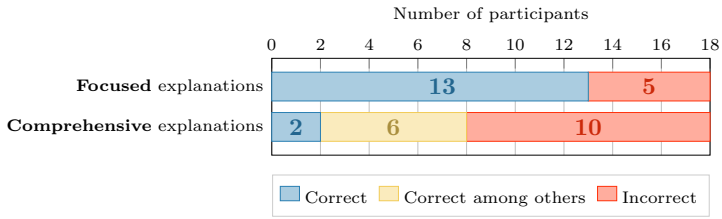


Figure 5.5. Distribution of participants who correctly identified the irrelevant dimension with **Focused** and **Comprehensive** explanations. Adapted from Publication V. © 2019 IEEE.

During the testing phase, the participants were required to select the rover's actions for a set of 11 states using their understanding of the policy obtained from the learning phase or, alternatively, express their inability to do so (answering *"I don't know"*).

The participants' choices during the testing phase were logged and compared to the ground truth provided by the policies, allowing the operationalization of the participants' understanding of the rovers' policies as a percentage of the state-action pairs correctly specified. The participants were also asked to specify which of the dimensions they thought irrelevant, in order to investigate the impact of the two types of explanations on the participants' ability to recognize the policies' irrelevant dimensions. While the interested reader can find a more in-depth analysis of the study in Publication V, this section focuses on these two aspects of the study.

We hypothesized that **Focused** explanations would allow the participants to gain a better understanding of the rovers' policies and therefore obtain a higher percentage of correct actions in the testing phase compared to **Comprehensive** explanations. With **Focused** explanations, the participants answered correctly to 50.5% of the queried states. With **Comprehensive** explanations, the percentage of correct answers was 49.0%, with no statistically significant difference observed between methods. The number of *"I don't know"* answers during the test was, however, extremely low, with only five such answers out of 198 queries states with **Focused** explanations and 12 with **Comprehensive** explanations. While the percentages of correct answers are reasonably high (especially considering the relatively short learning phase), these results indicate that the explanations may have inflated the participants' self-assessed understanding of the policies. This observation suggests that researchers should be aware that explanations can also be a tool of deception, used, intentionally or not, to persuade users that they have a good understanding of the explained system [57, 161].

Figure 5.5 summarizes the results for the participants' ability to identify the irrelevant dimension in the rovers' policies. With the help of **Focused** explanations, 13 out of 18 participants correctly identified the irrelevant dimension. With **Comprehensive** explanations, however, only two participants correctly recognized the irrelevant dimension, with the remainder either selecting the

wrong dimension (10) or listing additional dimensions together with the correct one (6). Thus, while no differences were observed for the percentages of correct actions during the testing phase, the proposed **Focused** explanations allowed the participants to more reliably identify the irrelevant dimension – a result in line with the observations of [158]. This advantage of **Focused** over **Comprehensive** explanations, already noticeable with the rovers' 5-dimensional state spaces, is likely to be of even greater relevance with more complex policies, where **Comprehensive** explanations would be cluttered by a larger number of irrelevant dimensions.

At the end of the study, the participants were given the possibility to describe their concept of an ideal explanation. Four participants described their ideal explanation as short, focused and clear – an observation in line with human bias for simpler explanations [30, 161]. Three participants stressed instead the importance of the order used to present the information in an explanation. These participants' preferences were met by the proposed **Focused** explanations, with the four measures used to sort the relevant dimensions and omit the irrelevant ones. Finally, while both **Focused** and **Comprehensive** explanations expose the state-action pairing of the policy, six participants expected semantic information about the logic used to encode the policy itself. Using the explanation shown in Figure 5.4 as an example, the explanation would need to be augmented to "I moved because the Battery Level was high, *and I therefore don't risk running out of battery power* and the Ground Quality is low *and my goal is to collect high quality samples*". Including such information poses serious challenges to the automatic generation of explanations. In [158], Elizalde *et al.* proposed augmenting explanations with information extracted from a hand-coded knowledge base of relations between variables, components, and procedures of the explained system. However, such detailed semantic information may not be always available, especially if the policy is learned from data or exploration. Nonetheless, it would be interesting to study how the goals pursued during training (for example, the maximised objective function) could be explained to the user. For example, an RL agent could augment explanations of its behaviour by reporting which terms of the reward function were the most relevant for the decision in question.

5.2 Discussion

After introducing the concept of transparency for autonomous systems and emphasizing its importance in the light of the present popularity of black-box models, this chapter presented the method proposed in Publication V for the generation of focused and robust local explanations. While in this work explanation generation was motivated by the need for novice users to understand and collaborate with their robots, explanations can also be a tool for the designers of black-box models, helping them debug their models [156, 162], trace the influ-

ence of training data on models' decisions [163], assess algorithmic fairness [164], and provide accountability [165]. The intended use and target audience of an explanation should therefore inform the design of explanation methods, especially regarding the trade-off between completeness and interpretability. In Publication V, local explanations were preferred over global explanations because of their ability to produce concise explanations by locally approximating the policy. The main issue with local explanation methods lies, however, in the difficult definition of what is local [151], circumvented in the proposed method by manually tuning the parameters of the sampling process. A promising alternative to local explanations are counterfactual explanations [151, 166], *i.e.*, explanations that, instead of exposing the relevant dimensions behind a decision, present an actionable perturbation of the dimensions that causes the policy to output a different decision (*e.g.*, *"The system took a_1 because d_1 was 30. If d_1 was 45, however, it would have taken a_2"*). Similarly, the intended use and target audience should influence the choice of medium used to convey the explanation. While this chapter focused on explanations expressed in natural language, such as those adopted in Publication V and related works [158, 157], explanations can also be visual, highlighting what areas or features of an input image influenced the model's decision [156, 167, 168]. As for the design of AL queries discussed in Section 3.2.1, the embodiment of robots can also be leveraged as an explanation medium to efficiently communicate robots' goals, capabilities [169, 170], and inabilities [20] to their users.

While the user study of Publication V focused on the amount and quality of the information included in an explanation, the proposed method can also detect the robustness of an explanation and abstain from explaining when small changes in the state space change the policy output or when no adequate descriptor is available. However, simply avoiding explaining is unlikely to be sufficient for most use cases. At the very least, ways to expose the reasons behind a *non-explanation* should be devised. The four measures presented in Section 5.1.1 could be used to explain to the user why no explanation was generated (*e.g.*, *"I did not explain because I cannot describe the current situation with my vocabulary"*), potentially triggering recovery actions by the user [171, 172]. While avoiding explanations can potentially help users form a faithful mental model of the policy, this ability is nevertheless likely to heavily influence the users' trust in the explained system. We therefore believe that research should continue to investigate the efficacy of explanation methods in close relation with their effect on their human recipients.

6. Conclusions

Pursuing the goal of providing programmability and adaptability to robots, this dissertation presented a suite of interactive methods aimed at robots that learn from human teachers. First, the LfD approach of Publication I was presented for the programming of in-contact tasks, enabling, through the use of demonstrations collected via kinesthetic teaching, the encoding of force profiles that are otherwise difficult to specify in declarative terms. When discussing the weaknesses of LfD approaches, the unidirectionality of the flow of information during the training process was identified as the most relevant for the case of novice users. To allow robots to participate in the training process in a more meaningful way, we therefore applied the AL paradigm to two robot learning scenarios: for the learning of temporal task models in Publication II and for the tuning of action parameters in a EUP framework in Publication IV. A crucial trade-off was explored in the query design of both works, with queries needing to be understandable and answerable by human teachers while allowing the training of the underlying model. Given the close interaction required between AL robots and their human teachers, the usability of the proposed methods was further evaluated with user studies. Comparing several query selection strategies, we observed how traditional strategies aiming solely at learning performance can negatively impact human teachers, raising their error rates and response times. The memory effort-aware strategy proposed in Publication III represents a first attempt at including teacher-related variables in the selection of queries. While the proposed strategy did not yield the expected results, it revealed how other aspects of the teacher-learner interaction, such as attention and boredom, have major effects on user perception and the usability of AL robots and, thus, should be further studied. Analysis of the users' perception of learning robots revealed the importance of transparency mechanisms for the success of the training process. The research in Publication V partially addressed this transparency issue, contributing to the rapidly expanding literature on XAI with a model-agnostic policy explanation method that generates robust and focused explanations expressed in natural language.

Within the development of AL robots, this dissertation placed great emphasis on the design of queries. In contrast to classification problems where the

standard query template *"Does this sample belong to that class?"* can be often effortlessly applied, the nature of the problems addressed here raised interesting challenges for query design. To handle the probabilistic and temporal nature of the model learned in Publication II, novel query types were proposed, together with the use of membership functions connecting the user's answers expressed in natural language with the underlying learning process. Similarly, using the robot's embodiment to ask questions as employed in Publication IV avoided exposing the numerical values of the tuned parameters directly to the user. While the presented solutions are problem-specific, we believe the principles observed in the design of these queries to be relevant not only for AL applications but, in general, for interactive learning methods that must balance between usability and performance.

Even though applying the aforementioned principles will produce learning methods that can be used effectively by novice users, the strengths and weaknesses of each learning paradigm will remain. We therefore believe that robots should support multiple learning paradigms when possible in order to leverage the strengths of each paradigm and offer teachers the most suitable channel at any one time. While the work of Publication II successfully integrated the LfD and AL paradigms, the structure of the learning process did not allow the participants to choose their preferred teaching method. Studying the integration of different input modalities while allowing the teacher to choose which modality to utilize would answer interesting questions about the efficacy of different paradigms either at distinct stages of the training process or in relation to the expertise of teachers. It would be also interesting to study the extent to which users adopt human teaching strategies, such as scaffolding and attention direction, and how these relate to the teaching tools made available to them.

This dissertation presented several studies where novice users taught AL robots by answering different types of questions. To properly study the usability of different query selection strategies and query designs, the interaction between robots and end-users was purposely limited to the training session. Thus, the study participants did not interact with the robot *after* the training session and, as a result, did not reap the benefits of their teaching by, for example, working alongside the robot or supervising its activities. We believe that, in a more realistic scenario, users would engage with robots in a more iterative manner, with training sessions interleaved with exploratory deployment sessions to test the current capabilities of the robot. Studying robot learning in both its training and deployment aspects would provide a more faithful depiction of the problem, leading to a deeper understanding of the interaction aspects already explored in Publication II and Publication III.

Adopting this holistic view of robot learning would allow researchers to study not only whether novice users can teach robots but also whether they can evaluate robot capabilities. Moreover, this would reveal the tools that robots should offer to assist such evaluation, informing, in turn, the development of transparency mechanisms that allow end-users to debug their learning robots. As

novice users cannot be expected to program robots in a traditional manner, robots should not be required to effectively explain the intricacies of their learning methods to users with no understanding of ML. Explanation methods like the one presented in Publication V could, however, be used to expose what is currently being learned, aiding the answering of questions such as “Is my robot making progress?”, “Can my robot do this at the moment?”, and “When should I stop training my robot?” Similarly to the training phase, a strong case can be made for the testing phase being bidirectional. Robot-initiated transparency mechanisms, such as explanations, expressions of inability, and requests of help, should be paired with user-initiated mechanisms that allow, for example, users to ask their robots questions or test particular skills.

This dissertation discussed interactive robot learning with humans-in-the-loop, with a focus on the interaction established between its actors. While this work was motivated by the fact that robots can not be programmed to face every situation *out-of-the-box*, it is nevertheless clear that the robots of the future can not realistically learn everything after deployment. Just as it is unreasonable to expect the average computer user to be able or willing to train their email-filter to reliably detect spam, we can not expect robots to learn entirely from their end-users how to grasp objects or navigate in buildings. We believe, however, that service robots should be equipped with a suite of basic capabilities that can be adapted and incrementally refined, after deployment, by their end-users, in order to be valuable and cost-effective. This dissertation, with the proposed learning methods and their evaluation with human teachers, contributes to this challenging long-term goal.

References

[1] Karel Capek. *R.U.R. (Rossum's Universal Robots)*. 1921.

[2] Mark E. Rosheim. *Robot Evolution: the Development of Anthrobotics*. John Wiley & Sons, 1994.

[3] Tomas Lozano-Perez. Robot programming. *Proceedings of the IEEE*, 71(7):821–841, 1983.

[4] Sonia Chernova and Andrea L. Thomaz. Robot learning from human teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(3):1–121, 2014.

[5] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469 – 483, 2009.

[6] Giacomo Rizzolatti, Leonardo Fogassi, and Vittorio Gallese. Neurophysiological mechanisms underlying the understanding and imitation of action. *Nature Reviews Neuroscience*, 2(9):661–670, 2001.

[7] Chrystopher L. Nehaniv and Kerstin Dautenhahn. Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications. In *Interdisciplinary approaches to robot learning*, pages 136–161. World Scientific, 2000.

[8] Katharina Muelling, Jens Kober, and Jan Peters. Learning table tennis with a mixture of motor primitives. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pages 411–416. IEEE, 2010.

[9] Alex X. Lee, Henry Lu, Abhishek Gupta, Sergey Levine, and Pieter Abbeel. Learning force-based manipulation of deformable objects from multiple demonstrations. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 177–184. IEEE, 2015.

[10] Kristen Stubbs, Pamela J. Hinds, and David Wettergreen. Autonomy and common ground in human-robot interaction: A field study. *IEEE Intelligent Systems*, 22(2):42–50, 2007.

[11] Sidney Strauss and Margalit Ziv. Teaching is a natural cognitive ability for humans. *Mind, Brain, and Education*, 6(4):186–196, 2012.

[12] Maya Cakmak and Andrea L. Thomaz. Eliciting good teaching from humans for machine learners. *Artificial Intelligence*, 217:198–215, 2014.

[13] Joseph B. Lyons. Being Transparent about Transparency: A Model for Human-Robot Interaction. In *2013 AAAI Spring Symposium Series*, 03 2013.

[14] Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, pages 663–670, 2000.

[15] Sonia Chernova and Manuela Veloso. Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research*, 34(1):1, 2009.

[16] Sonia Chernova and Manuela Veloso. Multi-thresholded approach to demonstration selection for interactive robot learning. In *Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction*, HRI '08, pages 225–232, New York, NY, USA, 2008. ACM.

[17] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018.

[18] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, New York, NY, USA, 2004. ACM.

[19] Cory J. Hayes, Maryam Moosaei, and Laurel D. Riek. Exploring implicit human responses to robot mistakes in a learning from demonstration task. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 246–252. IEEE, 2016.

[20] Minae Kwon, Sandy H. Huang, and Anca D. Dragan. Expressing robot incapability. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '18, page 87–95, New York, NY, USA, 2018. ACM.

[21] Crystal Chao, Maya Cakmak, and Andrea L. Thomaz. Transparent active learning for robots. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction*, HRI '10, pages 317–324. IEEE, 2010.

[22] Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[23] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4(1):129–145, 1996.

[24] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[25] Sebastian Thrun. Exploration in active learning. In *The Handbook of Brain Theory and Neural Networks*, pages 381–384. MIT Press, 1998.

[26] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, Andrew J. Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 7(1-2):1–179, 2018.

[27] Sylvain Calinon and Dongheui Lee. Learning control. In *Humanoid Robotics: a Reference*. Springer, 2018.

[28] Harish Ravichandar, Athanasios S. Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1), 2020.

[29] Jenna Burrell. How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big Data & Society*, 3(1):2053951715622512, 2016.

[30] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89. IEEE, 2018.

[31] Geoffrey Biggs and Bruce Macdonald. A survey of robot programming systems. In *Proceedings of the Australasian Conference on Robotics and Automation (ACRA)*. ARAA, 2003.

[32] Henry Lieberman, Fabio Paternò, Markus Klann, and Volker Wulf. End-user development: An emerging paradigm. In *End user development*, pages 1–8. Springer, 2006.

[33] Emilia I. Barakova, Jan C. C. Gillesen, Bibi E. B. M. Huskens, and Tino Lourens. End-user programming architecture facilitates the uptake of robots in social therapies. *Robotics and Autonomous Systems*, 61(7):704–713, 2013.

[34] Justin Huang, Tessa Lau, and Maya Cakmak. Design and evaluation of a rapid programming system for service robots. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 295–302. IEEE, 2016.

[35] Franz Steinmetz, Annika Wollschläger, and Roman Weitschat. Razer-a human-robot interface for visual task-level programming and intuitive skill parameterization. *IEEE Robotics and Automation Letters*, 3(3):1362–1369, 2018.

[36] Casper Schou, Jens S. Damgaard, Simon Bøgh, and Ole Madsen. Human-robot interface for instructing industrial tasks using kinesthetic teaching. In *IEEE ISR 2013*, pages 1–6. IEEE, 2013.

[37] Sonya Alexandrova, Zachary Tatlock, and Maya Cakmak. Roboflow: A flow-based visual programming language for mobile manipulation tasks. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5537–5544. IEEE, 2015.

[38] Maj Stenmark, Mathias Haage, and Elin Anna Topp. Simplified programming of re-usable skills on a safe industrial robot: Prototype and evaluation. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '17, page 463–472, New York, NY, USA, 2017. ACM.

[39] Franz Steinmetz and Roman Weitschat. Skill parametrization approaches and skill architecture for human-robot interaction. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 280–285. IEEE, 2016.

[40] Javi F. Gorostiza and Miguel A. Salichs. End-user programming of a social robot by dialog. *Robotics and Autonomous Systems*, 59(12):1102–1114, 2011.

[41] Felix Duvallet, Thomas Kollar, and Anthony Stentz. Imitation learning for natural language direction following through unknown environments. In *2013 IEEE International Conference on Robotics and Automation*, pages 1047–1053. IEEE, 2013.

[42] Baris Akgun, Maya Cakmak, Karl Jiang, and Andrea L. Thomaz. Keyframe-based learning from demonstration. *International Journal of Social Robotics*, 4(4):343–355, 2012.

[43] Saleema Amershi, Maya Cakmak, William B. Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, 2014.

[44] Maya Cakmak, Crystal Chao, and Andrea L. Thomaz. Designing interactions for robot active learners. *IEEE Transactions on Autonomous Mental Development*, 2(2):108–118, 2010.

[45] Stephanie Rosenthal, Manuela Veloso, and Anind K. Dey. Acquiring accurate human responses to robots' questions. *International Journal of Social Robotics*, 4(2):117–129, 2012.

[46] Maya Cakmak and Andrea L. Thomaz. Designing robot learners that ask good questions. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 17–24. ACM, 2012.

[47] Victor Gonzalez-Pacheco, Maria Malfaz, Jose C. Castillo, Alvaro Castro-Gonzalez, Fernando Alonso-Martín, and Miguel A. Salichs. How much should a robot trust the user feedback? analyzing the impact of verbal answers in active learning. In *International Conference on Social Robotics*, pages 190–199. Springer, 2016.

[48] Dorsa Sadigh, Anca D. Dragan, Shankar S. Sastry, and Sanjit A. Seshia. Active preference-based learning of reward functions. In *Robotics: Science and Systems*, 2017.

[49] Chandrayee Basu, Mukesh Singhal, and Anca D. Dragan. Learning from richer human guidance: Augmenting comparison-based learning with feature queries. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 132–140. ACM, 2018.

[50] Erdem Biyik and Dorsa Sadigh. Batch active preference-based learning of reward functions. In *Conference on Robot Learning*, pages 519–528, 2018.

[51] Kalesha Bullard, Yannick Schroecker, and Sonia Chernova. Active learning within constrained environments through imitation of an expert questioner. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 2045–2052. AAAI Press, 2019.

[52] Victor Gonzalez-Pacheco, Maria Malfaz, Alvaro Castro-Gonzalez, Jose C. Castillo, Fernando Alonso-Martín, and Miguel A. Salichs. Analyzing the Impact of Different Feature Queries in Active Learning for Social Robots. *International Journal of Social Robotics*, 10(2):251–264, 2018.

[53] Jimmy Baraglia, Maya Cakmak, Yukie Nagai, Rajesh Rao, and Minoru Asada. Initiative in robot assistance during collaborative task execution. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 67–74. IEEE, 2016.

[54] David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning Proceedings 1994*, pages 148–156. Elsevier, 1994.

[55] John R. Anderson, Dan Bothell, Christian Lebiere, and Michael Matessa. An Integrated Theory of List Memory. *Journal of Memory and Language*, 38(4):341–380, 05 1998.

[56] Andreas Theodorou, Robert H. Wortham, and Joanna J. Bryson. Why is my robot behaving like that? Designing transparency for real time inspection of autonomous robots. In *AISB Workshop on Principles of Robotics*, University of Sheffield, 04 2016. University of Bath.

[57] Adrian Weller. Transparency: Motivations and challenges. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 23–40. Springer, 2019.

[58] Taemie Kim and Pamela Hinds. Who should I blame? effects of autonomy and transparency on attributions in human-robot interaction. In *ROMAN 2006-The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pages 80–85. IEEE, 2006.

[59] Munjal Desai, Poornima Kaniarasu, Mikhail Medvedev, Aaron Steinfeld, and Holly Yanco. Impact of robot failures and feedback on real-time trust. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 251–258. IEEE, 2013.

[60] Andreas Holzinger. From machine learning to explainable AI. In *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, pages 55–66. IEEE, 2018.

[61] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[62] Heike Felzmann, Eduard Fosch-Villaronga, Christoph Lutz, and Aurelia Tamo-Larrieux. Robots and transparency: The multiple dimensions of transparency in the context of robot technologies. *IEEE Robotics & Automation Magazine*, 26(2):71–78, 2019.

[63] Sule Anjomshoae, Amro Najjar, Davide Calvaresi, and Kary Främling. Explainable agents and robots: Results from a systematic literature review. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1078–1088. IFAAMAS, 2019.

[64] Kerstin Fischer, Franziska Kirstein, Lars Christian Jensen, Norbert Krüger, Kamil Kukliński, Maria Vanessa aus der Wieschen, and Thiusius Rajeeth Savarimuthu. A comparison of types of robot control for programming by demonstration. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 213–220. IEEE, 2016.

[65] Chrystopher L. Nehaniv and Kerstin Dautenhahn. The correspondence problem. *Imitation in animals and artifacts*, 41, 2002.

[66] Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.

[67] Alexandros Paraschos, Christian Daniel, Jan Peters, and Gerhard Neumann. Using probabilistic movement primitives in robotics. *Autonomous Robots*, 42(3):529–551, 2018.

[68] Rainer Bischoff, Johannes Kurth, Günter Schreiber, Ralf Koeppe, Alin Albu-Schäffer, Alexander Beyer, Oliver Eiberger, Sami Haddadin, Andreas Stemmer, Gerhard Grunwald, and Gerhard Hirzinger. The KUKA-DLR lightweight robot arm - a new reference platform for robotics research and manufacturing. In *41st International Symposium on Robotics (ISR)*, pages 1–8, 2010.

[69] Tu-Hoa Pham, Abderrahmane Kheddar, Ammar Qammaz, and Antonis A. Argyros. Towards force sensing from vision: Observing hand-object interactions to infer manipulation forces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2810–2819, 2015.

[70] Wenzhen Yuan, Siyuan Dong, and Edward H. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.

[71] Giovanni Sutanto, Zhe Su, Stefan Schaal, and Franziska Meier. Learning sensor feedback models from demonstrations via phase-modulated neural networks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1142–1149. IEEE, 2018.

[72] Sylvain Calinon, Paul Evrard, Elena Gribovskaya, Aude Billard, and Abderrahmane Kheddar. Learning collaborative manipulation tasks by demonstration using a haptic interface. In *2009 International Conference on Advanced Robotics*, pages 1–6. IEEE, 2009.

[73] Petar Kormushev, Sylvain Calinon, and Darwin G. Caldwell. Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input. *Advanced Robotics*, 25(5):581–603, 2011.

[74] Franz Steinmetz, Alberto Montebelli, and Ville Kyrki. Simultaneous kinesthetic teaching of positional and force requirements for sequential in-contact tasks. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 202–209. IEEE, 2015.

[75] Alberto Montebelli, Franz Steinmetz, and Ville Kyrki. On handing down our tools to robots: Single-phase kinesthetic teaching for dynamic in-contact tasks. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5634. IEEE, 2015.

[76] Leonel Rozo, Sylvain Calinon, Darwin G. Caldwell, Pablo Jiménez, and Carme Torras. Learning collaborative impedance-based robot behaviors. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[77] Leonel Rozo, Sylvain Calinon, Darwin G. Caldwell, Pablo Jiménez, and Carme Torras. Learning physical collaborative robot behaviors from human demonstrations. *IEEE Transactions on Robotics*, 32(3):513–527, 2016.

[78] Loris Roveda, Giacomo Pallucca, Nicola Pedrocchi, Francesco Braghin, and Lorenzo Molinari Tosatti. Iterative learning procedure with reinforcement for high-accuracy force tracking in robotized tasks. *IEEE Transactions on Industrial Informatics*, 14(4):1753–1763, 2017.

[79] Hsi Guang Sung. *Gaussian mixture regression and classification*. PhD thesis, Rice University, 2004.

[80] Sylvain Calinon, Florent D'halluin, Eric L. Sauser, Darwin G. Caldwell, and Aude G. Billard. Learning and reproduction of gestures by imitation. *IEEE Robotics & Automation Magazine*, 17(2):44–54, 2010.

[81] Lawrence Rabiner and Biing-Hwang Juang. An introduction to Hidden Markov Models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[82] Shun-Zheng Yu. Hidden semi-markov models. *Artificial intelligence*, 174(2):215–243, 2010.

[83] Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 359–370. AAAI Press, 1994.

[84] Shun-Zheng Yu and Hisashi Kobayashi. Practical implementation of an efficient forward-backward algorithm for an explicit-duration hidden markov model. *IEEE Transactions on Signal Processing*, 54(5):1947–1951, 2006.

[85] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[86] Neville Hogan. Impedance control of industrial robots. *Robotics and Computer-Integrated Manufacturing*, 1(1):97–113, 1984.

[87] Günter Schreiber, Andreas Stemmer, and Rainer Bischoff. The fast research interface for the kuka lightweight robot. In *IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications – How to Modify and Enhance Commercial Controllers (ICRA)*, pages 15–21. IEEE, 2010.

[88] Scott Niekum, Sarah Osentoski, George Konidaris, and Andrew G. Barto. Learning and generalization of complex tasks from unstructured demonstrations. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5239–5246. IEEE, 2012.

[89] George Konidaris, Leslie Pack Kaelbling, and Tomas Lozano-Perez. From skills to symbols: Learning symbolic representations for abstract high-level planning. *Journal of Artificial Intelligence Research*, 61:215–289, 2018.

[90] Yeping Wang, Gopika Ajaykumar, and Chien-Ming Huang. See what I see: Enabling user-centric robotic assistance using first-person demonstrations. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pages 639–648, 2020.

[91] Sylvain Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent service robotics*, 9(1):1–29, 2016.

[92] Matthew E. Taylor, Halit Bener Suay, and Sonia Chernova. Integrating reinforcement learning with human demonstrations of varying ability. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '11, pages 617–624. IFAAMAS, 2011.

[93] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 627–635, 2011.

[94] Beomjoon Kim, Amir-massoud Farahmand, Joelle Pineau, and Doina Precup. Learning from limited demonstrations. In *Advances in Neural Information Processing Systems*, pages 2859–2867, 2013.

[95] Murtaza Hazara and Ville Kyrki. Reinforcement learning for improving imitated in-contact skills. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 194–201. IEEE, 2016.

[96] Pieter Abbeel, Adam Coates, and Andrew Y. Ng. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, 29(13):1608–1639, 2010.

[97] Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

[98] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st International Conference on Machine Learning*, pages 1–8, 2004.

[99] Sylvain Calinon and Aude Billard. Incremental learning of gestures by imitation in a humanoid robot. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 255–262. IEEE, 2007.

[100] Amir Ghalamzan, Chris Paxton, Gregory D. Hager, and Luca Bascetta. An incremental approach to learning generalizable robot tasks from human demonstration. In *2015 International Conference on Robotics and Automation (ICRA)*, pages 5616–5621. IEEE, 2015.

[101] Weitian Wang, Rui Li, Yi Chen, Max Z. Diekel, and Yunyi Jia. Facilitating human–robot collaborative tasks by teaching-learning-collaboration from human demonstrations. *IEEE Transactions on Automation Science and Engineering*, 16(2):640–653, 2018.

[102] Sebastian B. Thrun. Efficient exploration in reinforcement learning. Technical report, Carnegie Mellon University, 1992.

[103] Nicholas Roy and Andrew Mccallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of 18th International Conference on Machine Learning*, 2001.

[104] Manuel Lopes, Francisco Melo, and Luis Montesano. Active learning for reward estimation in inverse reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 31–46. Springer, 2009.

[105] Victor Gonzalez-Pacheco, Almudena Sanz, Maria Malfaz, and Miguel A. Salichs. Using novelty detection in HRI: Enabling robots to detect new poses and actively ask for their labels. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 1110–1115. IEEE, 2014.

[106] Joachim de Greeff and Tony Belpaeme. Why robots should be social: Enhancing machine learning through social human-robot interaction. *PLoS one*, 10(9):e0138061, 2015.

[107] Kalesha Bullard, Andrea L. Thomaz, and Sonia Chernova. Towards Intelligent Arbitration of Diverse Active Learning Queries. In *Intelligent Robots and Systems (IROS), 2018 IEEE/RSJ International Conference on*, pages 6049–6056. IEEE, 2018.

[108] Bradley Hayes and Brian Scassellati. Discovering task constraints through observation and active learning. In *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*, pages 4442–4449. IEEE, 2014.

[109] Elaine Schaertl Short, Adam Allevato, and Andrea L. Thomaz. Sail: Simulation-informed active in-the-wild learning. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 468–477. IEEE, 2019.

[110] Nemanja Rakicevic and Petar Kormushev. Active learning via informed search in movement parameter space for efficient robot task learning and transfer. *Autonomous Robots*, pages 1–19, 2019.

[111] Yuchen Cui and Scott Niekum. Active reward learning from critiques. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6907–6914. IEEE, 2018.

[112] Daniel S. Brown, Yuchen Cui, and Scott Niekum. Risk-aware active inverse reinforcement learning. In *Conference on Robot Learning*, pages 362–372, 2018.

[113] Chandrayee Basu, Erdem Biyik, Zhixun He, Mukesh Singhal, and Dorsa Sadigh. Active learning of reward dynamics from hierarchical queries. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2019.

[114] Johannes Kulick, Marc Toussaint, Tobias Lang, and Manuel Lopes. Active learning for teaching a robot grounded relational symbols. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1451–1457, 2013.

[115] Taylor Kessler Faulkner, Reymundo A. Gutierrez, Elaine Schaertl Short, Guy Hoffman, and Andrea L. Thomaz. Active attention-modified policy shaping. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 728–736, 2019.

[116] Manuel Lopes, Thomas Cederborg, and Pierre-Yves Oudeyer. Simultaneous acquisition of task and feedback models. In *Development and Learning (ICDL), 2011 IEEE International Conference on*, volume 2, pages 1–7. IEEE, 2011.

[117] Lotfi A. Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[118] Andrea L. Thomaz and Cynthia Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6-7):716–737, 2008.

[119] Rachel Lomasky, Carla E. Brodley, Matthew Aernecke, David Walt, and Mark Friedl. Active class selection. In *European Conference on Machine Learning*, pages 640–647. Springer, 2007.

[120] Elena Gribovskaya, Florent d'Halluin, and Aude Billard. An active learning interface for bootstrapping robot's generalization abilities in learning from demonstration. In *RSS Workshop Towards Closing the Loop: Active Learning for Robotics*, volume 62, 2010.

[121] Dongheui Lee and Christian Ott. Incremental kinesthetic teaching of motion primitives using the motion refinement tube. *Autonomous Robots*, 31(2-3):115–131, 2011.

[122] Matteo Saveriano, Sang-ik An, and Dongheui Lee. Incremental kinesthetic teaching of end-effector and null-space motion primitives. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3570–3575. IEEE, 2015.

[123] Martin Tykal, Alberto Montebelli, and Ville Kyrki. Incrementally assisted kinesthetic teaching for programming by demonstration. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 205–212. IEEE, 2016.

[124] Hema Raghavan, Omid Madani, and Rosie Jones. Active learning with feedback on features and instances. *Journal of Machine Learning Research*, 7:1655–1686, 2006.

[125] Gregory Druck, Burr Settles, and Andrew McCallum. Active learning by labeling features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 81–90. Association for Computational Linguistics, 2009.

[126] Kalesha Bullard, Sonia Chernova, and Andrea L. Thomaz. Human-driven feature selection for a robot learning classification tasks from demonstration. In *Robotics and Automation (ICRA), 2018 IEEE International Conference on*. IEEE, 2018.

[127] Nir Ailon. An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity. *Journal of Machine Learning Research*, 13(Jan):137–164, 2012.

[128] Kevin P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[129] Roy S. Lilly. The qualification of evaluative adjectives by frequency adverbs. *Journal of Verbal Learning and Verbal Behavior*, 7(2):333–336, 1968.

[130] Bernard M. Bass, Wayne F. Cascio, and Edward J. O'connor. Magnitude estimations of expressions of frequency and amount. *Journal of Applied Psychology*, 59(3):313, 1974.

[131] Thomas Minka. Estimating a Dirichlet distribution. Technical report, Massachusett Institute of Technology, 2000.

[132] Silvia Coradeschi, Amy Loutfi, and Britta Wrede. A short review of symbol grounding in robotic and intelligent systems. *KI-Künstliche Intelligenz*, 27(2):129–136, 2013.

[133] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[134] Ofer Dekel, Claudio Gentile, and Karthik Sridharan. Selective sampling and active learning from single and multiple teachers. *Journal of Machine Learning Research*, 13(Sep):2655–2697, 2012.

[135] Maya Cakmak and Andrea L. Thomaz. Optimality of human teachers for robot learners. In *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*, pages 64–69. IEEE, 2010.

[136] Burr Settles, Mark Craven, and Lewis Friedland. Active learning with real annotation costs. In *Proceedings of the NIPS Workshop on Cost-sensitive Learning*, pages 1–10, 2008.

[137] Aaron Bestick, Ravi Pandya, Ruzena Bajcsy, and Anca D. Dragan. Learning human ergornomic preferences for handovers. In *Robotics and Automation (ICRA), 2018 IEEE International Conference on*, pages 3257–3264. IEEE, 2018.

[138] Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *AAAI*, volume 5, pages 746–751, 2005.

[139] Edwin Lughofer. Hybrid active learning for reducing the annotation effort of operators in classification systems. *Pattern Recognition*, 45(2):884–896, 02 2012.

[140] Pedram Daee, Tomi Peltola, Marta Soare, and Samuel Kaski. Knowledge elicitation via sequential probabilistic inference for high-dimensional prediction. *Machine Learning*, 106(9-10):1599–1620, 10 2017.

[141] Burr Settles. From Theories to Queries: Active Learning in Practice. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, pages 1–18, 04 2010.

[142] Ashish Kapoor, Eric Horvitz, and Sumit Basu. Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *IJCAI*, volume 7, pages 877–882, 2007.

[143] Pinar Donmez and Jaime G. Carbonell. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM conference on Information and Knowledge Management*, pages 619–628, 2008.

[144] Sudheendra Vijayanarasimhan and Kristen Grauman. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2262–2269. IEEE, 2009.

[145] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 07 2018.

[146] George A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[147] John R. Anderson. *How can the human mind occur in the physical universe?* Oxford University Press, 2009.

[148] Seongsik Jo, Rohae Myung, and Daesub Yoon. Quantitative prediction of mental workload with the ACT-R cognitive architecture. *International Journal of Industrial Ergonomics*, 42(4):359–370, 07 2012.

[149] Sandra G. Hart. Nasa-task load index (NASA-TLX); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society annual meeting*, volume 50, pages 904–908. Sage Publications, 2006.

[150] Jesse Thomason, Aishwarya Padmakumar, Jivko Sinapov, Justin Hart, Peter Stone, and Raymond J. Mooney. Opportunistic active learning for grounding natural language descriptions. In *Conference on Robot Learning*, pages 67–76, 2017.

[151] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology*, 31:841, 2017.

[152] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a “right to explanation”. *AI magazine*, 38(3):50–57, 2017.

[153] Brian Y. Lim, Anind K. Dey, and Daniel Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2119–2128, 2009.

[154] Sylvain Bromberger. *On what we know we don't know: Explanation, theory, linguistics, and how questions shape them*. University of Chicago Press, 1992.

[155] Brian Y. Lim and Anind K. Dey. Assessing demand for intelligibility in context-aware applications. In *Proceedings of the 11th International Conference on Ubiquitous computing*, pages 195–204, 2009.

[156] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[157] Bradley Hayes and Julie A. Shah. Improving robot controller transparency through autonomous policy explanation. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 303–312. ACM, 2017.

[158] Francisco Elizalde, Enrique Sucar, Julieta Noguez, and Alberto Reyes. Generating explanations based on markov decision processes. In *Mexican International Conference on Artificial Intelligence*, pages 51–62. Springer, 2009.

[159] Ning Wang, David V. Pynadath, and Susan G. Hill. Trust calibration within a human-robot team: Comparing automatically generated explanations. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 109–116. IEEE, 2016.

[160] David Alvarez-Melis and Tommi S. Jaakkola. On the robustness of interpretability methods. In *2018 ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*. ICML, 2018.

[161] Bernease Herman. The promise and peril of human evaluation for model interpretability. In *NIPS Symposium on Interpretable Machine Learning*, 2017.

[162] Saleema Amershi, Max Chickering, Steven M. Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 337–346, 2015.

[163] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894. JMLR. org, 2017.

[164] Junzhe Zhang and Elias Bareinboim. Fairness in decision-making—the causal explanation formula. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[165] Ashraf Abdul, Jo Vermeulen, Danding Wang, Brian Y. Lim, and Mohan Kankanhalli. Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda. In *Proceedings of the 2018 CHI Conference on Human factors in Computing systems*, pages 1–18, 2018.

[166] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617, 2020.

[167] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *European Conference on Computer Vision*, pages 3–19. Springer, 2016.

[168] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *European Conference on Computer Vision*, pages 269–286. Springer, 2018.

[169] Anca D. Dragan, Kenton C. Lee, and Siddhartha S. Srinivasa. Legibility and predictability of robot motion. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 301–308. IEEE, 2013.

[170] Anca D. Dragan, Shira Bauman, Jodi Forlizzi, and Siddhartha S. Srinivasa. Effects of robot motion on human-robot collaboration. In *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 51–58. IEEE, 2015.

[171] Ross A. Knepper, Stefanie Tellex, Adrian Li, Nicholas Roy, and Daniela Rus. Recovering from failure by asking for help. *Autonomous Robots*, 39(3):347–362, 2015.

[172] Stephanie Rosenthal, Manuela Veloso, and Anind K. Dey. Is someone in this office available to help me? *Journal of Intelligent & Robotic Systems*, 66(1-2):205–221, 2012.

Errata

Publication I

At the end of Section 3.A, the correct definition of the quaternion representation of the rotational velocity is $\boldsymbol{q}_\omega \equiv e^{\frac{1}{2}\tau\dot{\boldsymbol{q}}}$. The correct definition was used in the implementation.

Aalto-DD 145/2020

Robots are being adopted in an increasing number of new application areas, such as health care, logistics, and domestic services. Far from the structured environments of industrial settings, interaction between robots and humans will often become necessary and potentially beneficial. Simultaneously, the target audience of robots will grow to include users who lack the technical skills needed to program robots in the traditional manner.

This dissertation proposes interactive learning methods based on Active Learning (AL) and Learning from Demonstration (LfD) that allow robots to learn by interacting with humans-in-the-loop. Furthermore, the dissertation pays particular attention to the Human-Robot Interaction (HRI) aspect of robot learning, investigating how the aforementioned methods influence and are influenced by the interactive nature of the training process.

ISBN 978-952-64-0054-9 (printed)
ISBN 978-952-64-0055-6 (pdf)
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Electrical Engineering
Department of Electrical Engineering and Automation
www.aalto.fi

BUSINESS + ECONOMY

ART + DESIGN + ARCHITECTURE

SCIENCE + TECHNOLOGY

CROSSOVER

DOCTORAL DISSERTATIONS