Interactive Machine Learning for Assistive Robots

Koert, Dorothea

(2020)

DOI (TUprints): https://doi.org/10.25534/tuprints-00014184

Publikationstyp: Dissertation

Fachbereich: 20 Fachbereich Informatik

Quelle des Originals: https://tuprints.ulb.tu-darmstadt.de/14184

Interactive Machine Learning for Assistive Robots

Interaktives Maschinelles Lernen für Assistenzroboter
Zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
Genehmigte Dissertation von Dorothea Carola Koert aus Hanau
Tag der Einreichung: 06. Januar 2020, Tag der Prüfung: 11. Februar 2020

1. Gutachten: Prof. Dr. Jan Peters
2. Gutachten: Prof. Dr. Heni Ben Amor
Darmstadt – D 17

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Informatik
Intelligente Autonome Systeme

Interactive Machine Learning for Assistive Robots
Interaktives Maschinelles Lernen für Assistenzroboter

Genehmigte Dissertation von Dorothea Carola Koert

1. Gutachten: Prof. Dr. Jan Peters
2. Gutachten: Prof. Dr. Heni Ben Amor

Tag der Einreichung: 06. Januar 2020
Tag der Prüfung: 11. Februar 2020

Darmstadt – D 17

Bitte zitieren Sie dieses Dokument als:
URN: urn:nbn:de:tuda-tuprints-141845
URL: http://tuprints.ulb.tu-darmstadt.de/14184

In memory of my grandfather -
a true engineer until the end

Erklärung zur Dissertation

Hiermit versichere ich, die vorliegende Dissertation ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. In der abgegebenen Thesis stimmen die schriftliche und elektronische Fassung überein.

Darmstadt, 01.Dezember 2020

D.Koert

Abstract

Intelligent assistive robots can potentially support elderly persons and caregivers in their everyday lives and facilitate a closer man and machine collaboration as an essential part of the yet to come 5-th industrial revolution. In contrast to classical robotic applications where robots were mostly designed for repetitive tasks, assistive robots will face a variety of different tasks in close contact with everyday users. In particular, it is difficult to foresee the variety of applications beforehand since they depend on a person's individual needs and preferences. This renders preprogramming of all tasks for assistive robots difficult and gives need to explore methods of how robots can learn new tasks at hand during deployment time. Learning from and during direct interaction with humans provides hereby a potentially powerful tool for an assistive robot to acquire new skills and incorporate prior human knowledge during the exploration of novel tasks. Such an interactive learning process can not only help the robot to acquire new skills or profit from human prior knowledge but also facilitates the participation of inexperienced users or coworkers which can lead to a higher acceptance of the robot. However, while on the one hand human presence and assistance can be beneficial during the learning process, on the other hand, close contact with inexperienced users also imposes challenges. In shared workspaces or in close contact with everyday users a robot should be able to adapt learned skills to achieve as little disturbance of humans as possible. It becomes also important to evaluate human preferences about such adaptation strategies, their understanding of interactive learning processes and different ways for human input into learning. To come closer to the goal of intelligent assistive robots is therefore important to develop novel interactive learning methods and evaluate them in different robotic applications.

This thesis focusses on three main challenges related to the development of assistive intelligent robots and their interaction with everyday users. The different parts of the thesis contribute not only novel theoretical methods but additionally also evaluations on different robotic tasks with users, that had zero or only little prior experience with robots.

The first challenge is to enable robots to learn cooperative skills from a potentially

open-ended stream of human demonstrations in an incremental fashion. While learning new skills from human demonstrations has already been exploited in the literature it remains challenging to learn skill libraries from incrementally incoming demonstrations and when the total number of skills is not known beforehand. Therefore, in the first part of the thesis, we introduce an approach for online and incremental learning of a library for collaborative skills. Here, we follow a Mixture of Experts based approach and incrementally learn a library of collaborative skills and a gating model from coupled human-robot trajectories. Once trained, the gating model can decide which skill to choose as an appropriate response to a human motion, based on prior demonstrations and activate the corresponding robot skill. In contrast to existing batch learning methods, our method does not require the total number of skills to be known a priori and can learn new skills as well as update existing skills from multiple human demonstrations. The cooperative skills are represented as Probabilistic Interaction Primitives which can capture variance and inherent correlations in the demonstrations. We evaluate our method with different human subjects in a task where a robot assists the subjects in making a salad. We also evaluate hereby how learned skills transfer between different subjects.

Second, intelligent assistive robots should be able to adapt learned skills to humans when working in close contact or shared workspaces. For Probabilistic Movement Primitives (ProMPs), which were chosen as a skill representation in this thesis, such methods for online adaptation were missing in the literature so far. Hereby, it is in particular important to also evaluate the perceived level of safety and comfort of humans according to different adaptation strategies. To this end, we present two methods for online adaptation of learned skills in a shared workspace setting. Here, we introduce two novel online adaptation methods for ProMPs, namely spatial deformation and temporal scaling. Spatial deformation avoids collisions by dynamically changing the shape of the movement primitive, while at the same time staying close to the demonstrated motions. In temporal scaling, we adapt the ProMP's velocity profile to avoid time-dependent collisions. To achieve intention aware adaptation in shared workspaces we combine both methods with a goal-directed prediction model for human motions. This prediction model can also be learned online from human motions. We conducted experiments for both novel adaptation methods in comparison to non-adaptive behavior with inexperienced users and evaluated influences on task performance as well as subjective metrics such as comfort and perceived level of safety.

The third challenge that we consider in this thesis is how a library of learned skills can be used in practice to solve sequential robotic tasks. While hereby reinforcement

learning offers a powerful tool for reward-driven learning and self-improvement, in real robotic applications it often suffers from costly and time-consuming sample collection. Here, human input might be beneficial to speed up and guide the learning. Therefore, it is important to enable and compare different ways how human input can be incorporated in reinforcement learning algorithms. In this thesis, we present an approach, which incorporates multiple forms of human input into reinforcement learning for sequential tasks. Since depending on the task human input might not always be correct, we additionally introduce the concept of self-confidence for the robot, such that it becomes able to question human input. We evaluate which input channels humans prefer during interaction and how well they accept suggestions or rejections of the robot if the robot becomes confident in its own decisions.

To summarize, the different parts of the thesis contribute to the development of intelligent assistive robots that can learn from imitating humans, adapt the learned skills dynamically to humans in shared workspaces and profit and learn from human input during self-driven learning of how to sequence skills into more complex tasks. The three main contributions to the state of the art are hereby: First, a novel approach to incrementally learn a library for collaborative skills when the total number of skills is not known a priori. Second, two novel methods for online adaptation of ProMPs and their combination with a goal-directed prediction model to enable intention aware online adaptation in shared workspaces. And third, an approach that combines multiple forms of human input with a reinforcement learning algorithm and a novel concept of self-confidence to learn and improve the sequencing of skills into more complex tasks.

Zusammenfassung

Intelligente Assistenzroboter können potentiell ältere Menschen und Pflegepersonal in ihrem Alltag unterstützen sowie eine engere Zusammenarbeit von Mensch und Maschinen als wesentlicher Bestandteil der noch bevorstehenden 5. industriellen Revolution ermöglichen. Im Gegensatz zu klassischen Anwendungsbereichen, in denen Roboter meist für rein repetitive Aufgaben konzipiert wurden, werden Assistenzroboter eine Vielzahl von unterschiedlichen Aufgaben in engem Kontakt mit den alltäglichen Nutzern bewältigen müssen. Insbesondere ist es schwierig, die Vielfalt der Anwendungen im Voraus zu kennen, da sie von den individuellen Bedürfnissen und Vorlieben eines Menschen abhängen. Dies erschwert die Vorprogrammierung aller Aufgaben für Assistenzroboter und macht es notwendig, Methoden zu erforschen, wie Roboter während der Einsatzzeit neue Aufgaben erlernen können. Das Lernen von und während der direkten Interaktion mit Menschen stellt dabei ein potentiell mächtiges Werkzeug für einen Assistenzroboter dar, um neue Fähigkeiten zu erwerben und menschliches Vorwissen bei der Erforschung neuer Aufgaben einzubeziehen. Ein solcher interaktiver Lernprozess kann Robotern nicht nur helfen, sich neue Fähigkeiten anzueignen oder von menschlichem Vorwissen zu profitieren, sondern erleichtert auch die Inklusion unerfahrener Benutzer oder Mitarbeiter in den Lernprozess, was zu einer höheren Akzeptanz des Roboters führen kann. Doch während einerseits die menschliche Anwesenheit und Unterstützung während des Lernprozesses von Vorteil sein kann, stellt der enge Kontakt mit unerfahrenen Benutzern auch eine Herausforderung dar. In gemeinsamen Arbeitsbereichen oder in engem Kontakt mit alltäglichen Nutzern sollte ein Roboter in der Lage sein, gelernte Fähigkeiten so anzupassen, dass der Mensch möglichst wenig gestört wird. Es ist dabei wichtig, die Präferenzen von Menschen bezüglich solcher Anpassungsstrategien, ihr Verständnis interaktiver Lernprozesse und verschiedene Möglichkeiten für menschlichen Input zum Lernen zu evaluieren. Um dem Ziel intelligenter Assistenzroboter näher zu kommen, müssen daher, neuartige interaktive Lernmethoden entwickelt und in verschiedenen Roboteranwendungen getestet werden.

Diese Arbeit konzentriert sich auf drei Hauptherausforderungen, die mit der Entwicklung von assistiven intelligenten Robotern und deren Interaktion mit alltäglichen Nutzern zusammenhängen. Die verschiedenen Teile der Arbeit tragen nicht nur neue theoretische Methoden bei, sondern zusätzlich auch Experimente zu verschiedenen Roboteraufgaben mit Nutzern, die keine oder nur wenig Erfahrung mit Robotern hatten.

Die erste Herausforderung besteht darin, Roboter in die Lage zu versetzen, kooperative Fähigkeiten aus menschlichen Demonstrationen auf inkrementelle Weise zu erlernen. Während das Erlernen neuer Fähigkeiten aus menschlichen Demonstrationen in der Literatur bereits ausgenutzt wurde, bleibt es eine Herausforderung, Fähigkeitsbibliotheken aus inkrementell eingehenden Demonstrationen zu lernen, wenn die Gesamtzahl der Fähigkeiten vorher nicht bekannt ist. Daher stellen wir im ersten Teil dieser Thesis einen Ansatz für das online- und inkrementelle Lernen einer Bibliothek für kollaborative Fähigkeiten vor. Hier folgen wir einem Mixture-of-Experts basierten Ansatz und lernen inkrementell eine Bibliothek für kollaborative Fähigkeiten und ein Gating-Modell aus gekoppelten Mensch-Roboter-Trajektorien. Einmal trainiert, kann das Gating-Modell anhand vorheriger Demonstrationen entscheiden, welche Fähigkeit als angemessene Reaktion auf eine menschliche Bewegung gewählt werden soll und die entsprechende Roboterfähigkeit aktivieren. Im Gegensatz zu bestehenden Batch-Lernmethoden erfordert unsere Methode nicht, dass die Gesamtzahl der Fertigkeiten a priori bekannt ist und kann sowohl neue Fertigkeiten erlernen als auch bestehende Fertigkeiten aus mehreren menschlichen Demonstrationen aktualisieren. Die kooperativen Fähigkeiten werden als Probabilistische Interaktionsprimitive dargestellt, die Varianz und inhärente Zusammenhänge in den Demonstrationen erfassen können. Wir evaluieren unsere Methode mit verschiedenen menschlichen Probanden in einer Aufgabe, bei der ein Roboter den Probanden bei der Zubereitung eines Salats assistiert. Wir untersuchen hierbei auch, inwieweit die erlernten Fähigkeiten zwischen verschiedenen Menschen übertragen werden können.

Zweitens, sollten intelligente Assistenzroboter in der Lage sein, gelernte Fähigkeiten an den Menschen anzupassen, wenn sie in engem Kontakt oder in gemeinsamen Arbeitsbereichen arbeiten. Für Probabilistische Bewegungsprimitive (Probabilistic Movement Primitives, ProMPs), die in dieser Arbeit als Fertigkeitsdarstellung gewählt wurden, fehlten solche Methoden zur Online-Anpassung bisher in der Literatur. Dabei ist es insbesondere wichtig, auch das wahrgenommene Sicherheits- und Komfortniveau des Menschen nach verschiedenen Anpassungsstrategien zu bewerten. Zu diesem Zweck stellen wir zwei Methoden zur Online-Adaption von gelernten Fähigkeiten in einem gemeinsamen Arbeitsumfeld vor. Hier stellen wir zwei neuartige Online-Anpassungsmethoden für ProMPs vor, nämlich die räumliche

Deformation und die zeitliche Skalierung. Die räumliche Deformation vermeidet Kollisionen, indem sie die Form des Bewegungsprimitivs dynamisch verändert, während sie gleichzeitig nahe an den demonstrierten Bewegungen bleibt. Bei der zeitlichen Skalierung passen wir das Geschwindigkeitsprofil des ProMPs an, um zeitabhängige Kollisionen zu vermeiden. Um eine intentionale Anpassung in gemeinsamen Arbeitsräumen zu erreichen, kombinieren wir beide Methoden mit einem zielgerichteten Vorhersagemodell für menschliche Bewegungen. Dieses Vorhersagemodell kann auch online aus menschlichen Bewegungen gelernt werden. Wir führen dabei Experimente für beide neuartigen Adaptionsmethoden im Vergleich zu nicht-adaptivem Verhalten mit unerfahrenen Benutzern durch und evaluieren Einflüsse auf die Aufgabenausführung sowie subjektive Metriken wie Komfort und wahrgenommenes Sicherheitsniveau.

Die dritte Herausforderung, die wir in dieser Arbeit betrachten, ist, wie eine Bibliothek gelernter Fähigkeiten in der Praxis zur Lösung sequentieller Roboteraufgaben eingesetzt werden kann. Während hierbei Reinforcement Learning ein mächtiges Werkzeug für belohnungsorientiertes Lernen und Selbstverbesserung bietet, leidet es in realen Roboteranwendungen oft unter einer kostspieligen und zeitaufwendigen Probensammlung. Hier kann menschlicher Input von Vorteil sein, um das Lernen zu beschleunigen und zu leiten. Daher ist es wichtig, verschiedene Möglichkeiten zu ermöglichen und zu vergleichen, wie menschliche Eingaben in Reinforcement Learning-Algorithmen integriert werden können. In dieser Arbeit stellen wir einen Ansatz vor, der verschiedene Formen des menschlichen Inputs in das Reinforcement Learning für sequentielle Aufgaben einbezieht. Da der menschliche Input je nach Aufgabenstellung nicht immer korrekt sein kann, führen wir zusätzlich das Konzept des Selbstvertrauens des Roboters ein, so dass er in der Lage ist, den menschlichen Input zu hinterfragen. Wir untersuchen, welche Eingabekanäle Menschen während der Interaktion bevorzugen und wie gut sie Vorschläge oder Ablehnungen des Roboters akzeptieren, wenn der Roboter selbstbewusst in seinen eigenen Entscheidungen wird.

Zusammengefasst tragen die verschiedenen Teile der Arbeit zur Entwicklung intelligenter Assistenzroboter bei, die durch Nachahmung von Menschen lernen können, die gelernten Fähigkeiten in gemeinsamen Arbeitsbereichen dynamisch an den Menschen anpassen und beim selbstgesteuerten Lernen von der Abfolge der Fähigkeiten zu komplexeren Aufgaben und von menschlichem Input profitieren können. Die drei wichtigsten Beiträge zum Stand der Technik sind hiermit: Erstens, ein neuartiger Ansatz zum inkrementellen Lernen einer Bibliothek für kollaborative Fähigkeiten, wenn die Gesamtzahl der Fähigkeiten nicht a priori bekannt ist. Zweitens, zwei neue Methoden zur Online-Adaption von ProMPs und deren Kombination mit einem zielgerichteten Bewegungs Vorhersagemodell, um eine intentionsbasierte Online-Adaption in gemeinsamen Arbeitsbereichen zu ermöglichen. Und drittens, ein Ansatz, der mehrere Formen menschlichen Inputs mit einem Reinforcement Learning-Algorithmus und einem neuartigen Konzept von Selbstvertrauen des Roboters kombiniert, die Abfolge von Fähigkeiten in komplexeren Aufgaben zu erlernen und zu verbessern.

Acknowledgments

During the work on this thesis, I have received a lot of support and I am very grateful to everyone, who helped me to achieve this goal. In particular, I want to thank

- my supervisor Prof. Jan Peters, for giving me valuable input and advice in the right moments and enough freedom to work on my own ideas and topics throughout this thesis,
- Prof. Heni Ben Amor, for being my external committee member and Prof. Oskar von Stryk, Prof. Frank Jäkel and Prof. Carsten Binnig for becoming members of my committee,
- all colleagues, collaborators, and students I had the pleasure to meet and work with during my Ph.D., in particular, Joni, Guilherme, Marco, Rudi, Filipe, Oleg, Daniel, Svenja, Gregor, Samuele, Michael, Stefan, Nour, Yi, Albert, Janine, Susi, Vildan and Carlo,
- all the great people I met before my Ph.D. during my time at Team Hector - all the things I learned in this time started my passion for robots and made me the roboticist that I am,
- Guilherme, for his support, guidance and great tips for writing during my Master's thesis and at the beginning of my Ph.D. and Prof. Constantin Rothkopf for valuable discussions and scientific advice to the second part of this thesis,
- Joni, for all the helpful discussions, scientific and personal advice throughout the last years, for his great sense of humor and for helping me not to stop believing in myself and research in the hardest times,
- Dirk, for continuously providing encouragement, motivation, support and valuable ideas during my work for the KoBo project and for being an inspiration in his working attitude and personality,

- Susi, for being the best colleague and teammate I could have wished for, for making sure I get enough food, light, and air in the basement, for having sometimes more faith in me than I have in myself and for motivating and supporting me when I needed it the most.

Finally, I am also very grateful for my friends, parents, grandparents and my brother and sister who are always there for me, helped me to reach this point in life and provided the most valuable counterpart to work in the last years. Most of all I deeply thank Valerij for beeing my light at the end of all the long lab sessions, for always giving me strength, love, and support in good and bad times and for reminding me about what's important in life at the end of the day - and the lion baby inside me and its strength to live through the last months together with me.

Research in this thesis was supported by the German Federal Ministry of Education and Research (BMBF) project 16SV7984 (KOBO34).

Contents

1. Introduction **3**
1.1. Motivation . . . 3
1.1.1. Learning to Imitate . . . 6
1.1.2. Learning to Adapt . . . 7
1.1.3. Learning to Improve . . . 9
1.1.4. Research Questions . . . 10
1.2. Main Contributions . . . 11
1.2.1. Incremental Learning of an Open-Ended Collaborative Skill Library . . . 12
1.2.2. Learning Intention Aware Online Adaptation of Movement Primitives . . . 12
1.2.3. Multi-Channel Interactive Reinforcement Learning for Sequential Tasks . . . 13
1.3. Necessary Background . . . 14
1.3.1. Definition for Skills and Tasks . . . 14
1.3.2. Probabilistic Movement Primitives . . . 15
1.4. Outline . . . 17

2. Incremental Learning of an Open-Ended Collaborative Skill Library **19**
2.1. Related Work . . . 21
2.2. Incremental Interaction Primitives . . . 23
2.2.1. Batch Learning for Mixture of Interaction ProMPs . . . 23
2.2.2. Online Open-Ended Mixture of Interaction ProMPs . . . 24
2.2.3. A Skill Library for Collaborative Tasks . . . 29
2.3. Experimental Evaluation . . . 30
2.3.1. 2D trajectory data . . . 30
2.3.2. Learning a Cooperative Skills with a Robotic Arm . . . 33
2.3.3. Limitations . . . 44
2.4. Conclusion . . . 44

3. Learning Intention Aware Online Adaptation of Movement Primitives 47

3.1. Related Work 49
3.2. Intention Aware Online ProMP Adaptation 51
3.2.1. Online Spatial Deformation of ProMPs 51
3.2.2. Online Temporal Scaling of ProMPs 53
3.2.3. Probabilistic Model for Human Trajectory Prediction 55
3.3. Experimental Evaluation 60
3.3.1. Experimental Setup 60
3.3.2. Learning Motion Goals 60
3.3.3. Intention Aware Online Adaptation of ProMPs 62
3.3.4. Limitations 66
3.4. Conclusion 67

4. Multi-Channel Interactive Reinforcement Learning for Sequential Tasks 69

4.1. Related Work 71
4.2. Multi-Channel Interactive Reinforcement Learning 73
4.2.1. Human Input Channels 74
4.2.2. Reinforcement Learning Module 77
4.2.3. Human-Advice-Module 77
4.2.4. Action Selection 78
4.2.5. Self-Confidence 78
4.2.6. Component Implementation 79
4.3. Experimental Evaluation 84
4.3.1. Robotic Kitchen Task 84
4.3.2. Robotic Sorting Task 95
4.3.3. Discussion 103
4.4. Conclusion 104

5. Conclusion 107

5.1. Summary 107
5.2. Future Work 109

A. Publication List 113

A.1. Journal Articles 113
A.2. Articles in Conference Proceedings 113
A.3. Workshop Papers 114

B. Curiculum Vitae 115

Notation

This table includes the basic notation used in this thesis. Due to the vast amount of variables, symbols are defined in the specific section where they are used. If a symbol is overloaded across chapters, the correct meaning should be apparent from the context.

Notation	Description
x	Scalar value
$\dot{x}$	Time derivative
$\boldsymbol{x} = [x_0, .., x_i, ..., x_N]$	Vector of elements x_i
x_i	Element i of vector $\boldsymbol{x}$
$\boldsymbol{X}$	Matrix $\boldsymbol{X}$with elements X_{ij}
$\boldsymbol{X}^{-1}$	Inverse of the matrix $\boldsymbol{X}$
$[]^{\mathrm{T}}$	Transposition
$\boldsymbol{I}$	Identity matrix
$p(x)$	Distribution over x
$p(x\|y)$	Distribution over x given y

Abbreviations

Acronym	Description
EM	Expectation Maximization
DMP	Dynamic Movement Primitive
GMM	Gaussian Mixture Models
HRI	Human Robot Interaction
KL	Kullback-Leibler
LfD	Learning from Demonstration
MDP	Markov Decision Process
ProMP	Probabilistic Movement Primitive
RL	Reinforcement Learning
ROS	Robot Operating System

1. Introduction

The computer is incredibly fast, accurate, and stupid. Man is unbelievably slow, inaccurate, and brilliant. The marriage of the two is a force beyond calculation.
(L. Cherne)

1.1. Motivation

Intelligent assistive robots have the potential to become one of the core technologies of industrial applications as well as helpful and assistive partners in everyday life. While robots have been used in the car manufacturing industry already since the 1960s, only in the last decades robotic technology has reached out to a larger variety of industrial fields such as logistics, medical and food industries. In particular, the future use of more cooperative robots and bringing back the human factor into the industrial automation processes will be one of the facilitated key points in the future industry 5.0 revolution (Nahavandi, 2019; Østergaard, 2018). Hereby, intelligent assistive robot-coworkers should enable close cooperation between humans and robots and synergetic utilization of the workforce of humans and machines.

Besides the need for intelligent robot assistants in the industrial sector, another promising application area for assistive robots is in the healthcare sector. In the last years many job offers for elderly care and nursing positions could not be filled in Germany (Bundesagentur für Arbeit, 2019), while Germany (Allmendinger and Ebner, 2006) and Europe (Doblhammer and Ziegler, 2006) are facing a strong demographic change resulting in more elderly people who require assistance in the next years and less young people to fill the open positions. Intelligent assistive robots can hereby potentially assist caregivers as well as elderly people in their everyday life activities. In elderly homes, such assistive robots can potentially reduce the load of repetitive tasks for nursing staff, and as personal assistants for the elderly, they could facilitate a longer independent living.

However, despite their great potential, which is summarized in Figure 1.1, intelligent assistive robots still need further development to reach the stage of being a

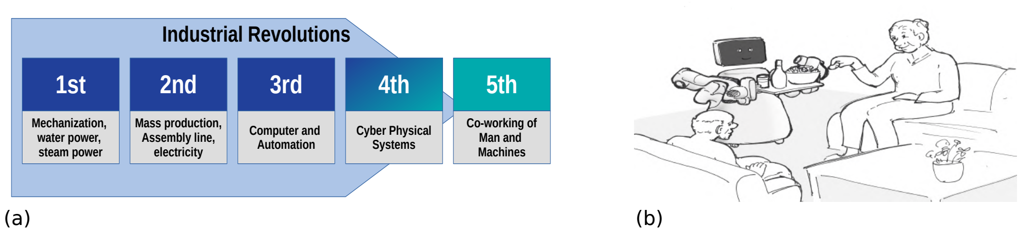

Industrial Revolutions
1st
Mechanization, water power, steam power
2nd
Mass production, Assembly line, electricity
3rd
Computer and Automation
4th
Cyber Physical Systems
5th
Co-working of Man and Machines
(a)
(b)

Figure 1.1.: Intelligent assistive robots can potentially be beneficial in the industrial and the healthcare sector. (a) The 5-th industrial revolution will include closer human-machine collaboration and facilitates to bring the human factor back into the industrial manufacturing process (Nahavandi, 2019). Hereby, intelligent robotic coworkers can potentially enable synergetic collaboration between man and machines in shared workspaces. (b) In Germany and all over Europe the demographic change results in an increasingly elderly population while at the same time there is a lack of caregivers and nursing staff. Intelligent assistive robots provide a potential solution to this challenge and could assist caregivers and elderly persons in everyday life activities.[1]

real benefit in such challenging tasks in shared human-robot environments (Hayes and Scassellati, 2013; Tsarouchi et al., 2016). Although in recent years robotic solutions have been used in more and more application areas, the applications often focus on tasks where the robots do not face the challenges of close interaction with everyday users. Additionally, their development and deployment are commonly still restricted to the expertise of roboticists and programming experts.

While on the one hand close contact with humans provides challenges in terms of adaptation to and prediction of human behaviors and intentions, on the other hand, humans can also be actually helpful to the robots. In particular, if robots become able to continuously learn and profit from the broad knowledge and skills of humans this can be a big chance for them to continuously learn and enlarge their capabilities. However, to enable robots to quickly adapt to new tasks at hand and to work in close contact with humans new methods are needed that allow also inexperienced users to teach new tasks to a robotic assistant. Additionally, assistive robots need to be able to adapt learned tasks to humans when working in shared

[1] Kobo serving snacks scenario of the BMBF project 16SV7984 (KOBO34), Illustration from Tamara Loibl, Courtesy of the Franka Emika GmbH

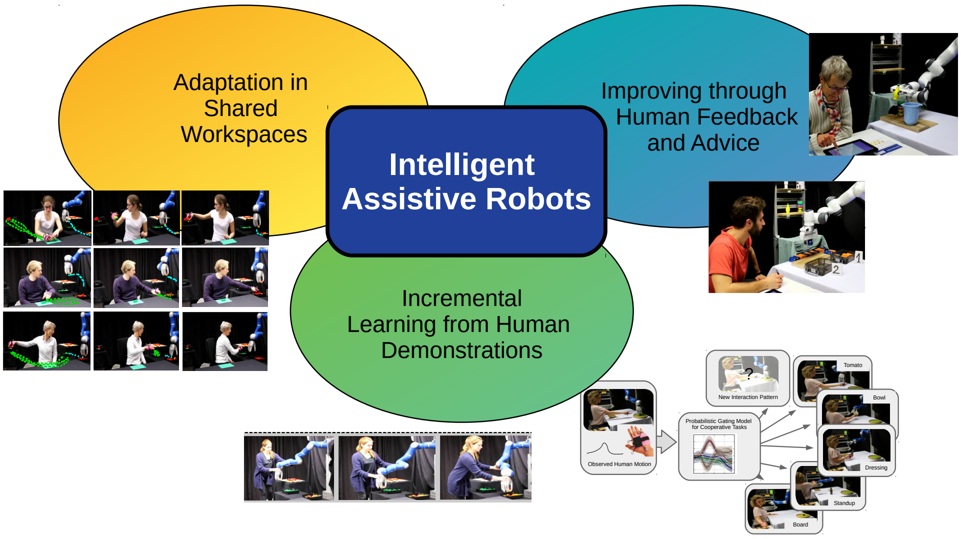

Adaptation in Shared Workspaces
Improving through Human Feedback and Advice
Intelligent Assistive Robots
Incremental Learning from Human Demonstrations
New Interaction Pattern
Tomato
Bowl
Probabilistic Gating Model for Cooperative Tasks
Observed Human Motion
Dressing
Standup
Board

Figure 1.2.: This thesis contributes to the development of intelligent assistive robots in three main areas: First, incremental skill learning from human demonstrations can enable also non-expert users to teach new tasks to a robot. Second, the ability to adapt learned skills to a human can improve coworking in workspaces shared by humans and robots. And third, by incorporating human feedback and advice in Reinforcement Learning a robot can learn and self-improve in sequencing already learned skills to achieve more complex tasks.

workspaces as well as the ability to improve their behavior based on human input.

Incorporating human demonstrations, feedback, and advice into training and improvement of robotic tasks can also help to reduce the initially required amount of training data since, in robotics, such a reduced amount of training data is usually crucial to enable practice transfer of Machine Learning methods.
This thesis contributes hereby in particular to the topics incremental skill learning from human demonstrations, online adaptation of skills in shared workspaces and interactive Reinforcement Learning for sequential robotic tasks as visualized in Figure 1.2. We believe that the contributions of the thesis can hereby help to get closer to the goal of intelligent assistive robots that should be able to learn new skills from human demonstrations, adapt their learned skills to human co-workers

and leverage human feedback and advice to learn and improve the sequencing of single skills. The following sections present a short summary of prior works from the related research fields, namely *Learning to Imitate* in Section 1.1.1, *Learning to Adapt* in Section 1.1.2 and *Learning to Improve* in Section 1.1.3. Subsequently, Section 1.1.4 states the particular research questions of this thesis.

1.1.1. Learning to Imitate

Learning new tasks by imitating a demonstrator offers a way for a robot to quickly reduce the search space for an optimal solution of an unknown task (Atkeson and Schaal, 1997; Billard et al., 2008; Schaal, 1999; Hussein et al., 2017; Osa et al., 2018) and at the same time is a natural and bioinspired teaching mechanism for humans (Meltzoff and Moore, 1977). The first concepts for imitation learning and programming by demonstration date back to the 1980s (Lozano-Perez, 1983; Levas and Selfridge, 1984). In these early approaches often demonstrations were acquired from teleoperated control of the robots which is nowadays often replaced by more user-friendly means such as direct guidance of the robot, namely kinesthetic teaching (Inamura et al., 2006; Weiss et al., 2009; Akgun and Subramanian, 2011) or learning from external observations using vision and motion capturing systems (Kuniyoshi, 1989; Jenkins and Mataric, 2002; Ijspeert et al., 2002; Chernova and Thomaz, 2014). Considering how demonstrations are recorded and how the embodiment of the learner matches the body of the teacher, a categorization into record mapping and embodiment mapping of imitation learning approaches has been proposed in Argall et al. (2009). The record mapping refers hereby to the way how the demonstrations are recorded and what particular information (e.g. joint or task space trajectories) is provided to the learner. The embodiment mapping refers to the concept of how to map the body of the teacher to the embodiment of the learner. The teacher can hereby, either demonstrate desired behavior directly on the learner, e.g. through teleoperation or kinesthetic teaching of the robot, or the robot can observe motions of the teacher which give need to also tackle to correspondence problem.

While Learning from Demonstrations (LfD) offers an often intuitive way also for non-expert users to teach new tasks to a robot, it is also often prone to the assumptions of good demonstrations, thus depends on the demonstration abilities of the demonstrator. Another challenge can hereby also be to learn from demonstrations of potentially different demonstrators, that might vary in their concrete task execution. Moreover, pure imitation, thus only 'replay' of the observed motions is often problematic as soon as the task setting or the environment change slightly between

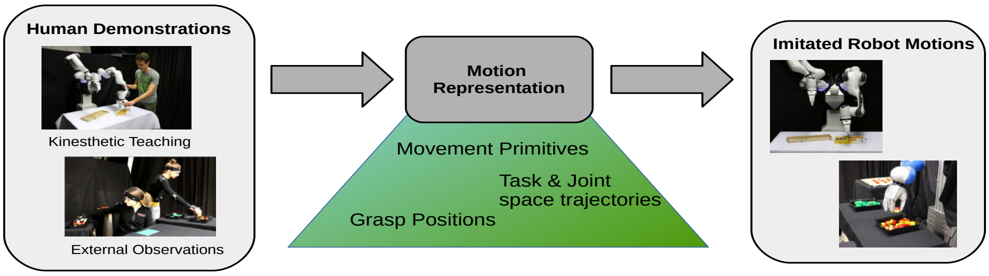


Figure 1.3.: Learning by imitating human demonstrations offers the possibility for assistive robots to learn new tasks from non-robotic experts. The demonstrations can hereby be given in different forms e.g. by direct guidance (i.e. kinesthetic teaching) or over external observations from e.g. cameras of motion tracking systems. Movement primitives offer hereby a beneficial and modular representation of such human demonstrations.

training and execution time. The concept of learning movement primitives (Ijspeert et al., 2013; Calinon et al., 2007; Paraschos et al., 2018) offers hereby a modular representation of the demonstrations for imitation learning of robot skills. Figure 1.3 visualizes an exemplary procedure on how to get from human demonstrations to imitating robot motions.

However, given that future assistive robots will most likely face a large variety of different tasks, it is desirable to not only learn movement primitives in a batch-wise fashion, but to be able to incrementally build a skill library over time from multiple human demonstrations.

1.1.2. Learning to Adapt

As mentioned in Section 1.1.1, pure reproduction of demonstrated behavior is often not sufficient. In particular, changing environmental conditions or varying object locations give need for generalization capabilities of learned motions after training time. To this end, the use of Machine Learning to extract repetitive movement patterns and skills from demonstrations was already suggested in 1994 by Muench et al. (1994). The more recent concept of motion primitives(Ijspeert et al., 2013; Calinon et al., 2007; Paraschos et al., 2018) provides a decomposable task structure as well as possibilities for temporal modulation and adaptation of robot motions

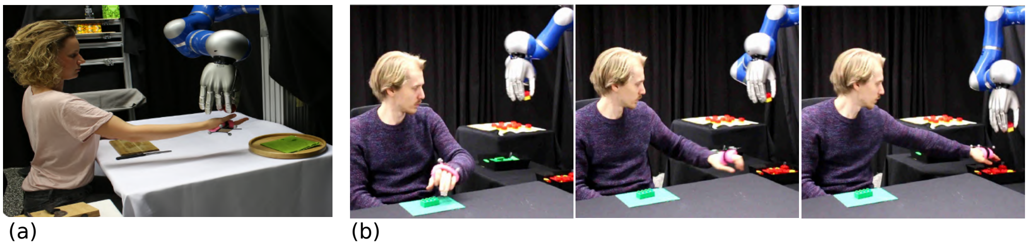


Figure 1.4.: Future assistive robots require the ability to adapt learned skills to humans (a) In cooperative tasks robots need to adapt their motions to match the human counterpart (Koert et al., 2018) (b) When performing independent tasks in shared workspaces with humans, robots should adapt their motions to achieve little disturbance of their human co-workers (Koert et al., 2019a).

to changing goal locations or via points (Paraschos et al., 2018). The main idea of motion primitives is hereby to learn generalizable task representations from only a few human demonstrated trajectories. The modular structure of the motion primitives enables thereby also composition and transfer of subtasks to different robot behaviors. The most commonly used trajectory-based representation are hereby Dynamic Movement Primitives (DMPs), which represents a dynamical system that can be modulated by a non-linear forcing function (Ijspeert et al., 2003, 2013). An extension of DMPs to support sequencing was also proposed in (Kulvicius et al., 2011). Probabilistic representations for movement primitives that also consider variance in the demonstrations have been introduced in the literature to handle and represent variance in the demonstrations (Paraschos et al., 2018; Calinon et al., 2007).

While movement primitives already provide beneficial adaptation capabilities to static changes in the environment or object states, shared workspaces between robots and humans give additional rise to tackle challenges in adaptation of robot skills to humans (Vasic and Billard, 2013; Kulić and Croft, 2005; Lasota and Shah, 2015; Koppenborg et al., 2017; Dragan et al., 2013). Hereby, safe coexistence in shared workspaces (Vasic and Billard, 2013; Kulić and Croft, 2005) as well as predictability and legibility (Dragan et al., 2013) of robot motions should be objectives to the adaptation. Two types of situations that require such adaptation of learned skills to humans, that should be in particular highlighted here, are cooperative tasks, such as handovers, and independent tasks with workspace overlap. Figure 1.4 shows examples for such situations. While in cooperative tasks, as depicted in Figure 1.4 (a) it is crucial that a robot can adapt its motions to match the human counterpart, during independent motions, as shown in Figure 1.4 (b), in shared workspaces it should mainly be achieved that the robot's task execution does not disturb or hinder the humans task execution. In order to enable robots with the ability of such dynamic adaptation of their skills to human co-workers, additionally means for prediction of human motion patterns and intentions need to be considered (Ikeda et al., 2013; Ravichandar and Dani, 2015; Bai et al., 2015; Luo et al., 2018; Dermy et al., 2017; Mainprice et al., 2015).

1.1.3. Learning to Improve

While there are a number of tasks where robots can directly imitate human demonstrations to achieve sufficient task performance, in more complex tasks a robot might not be able to directly learn only from human demonstrations. In particular, if a robot has already learned a library of single skills it is still challenging how to sequence the skills correctly to achieve more complex tasks. Means for self-improvement and the ability to learn complex tasks from the combination of basic skills are therefore highly desirable for assistive robots. Hereby, the concept of Reinforcement Learning (Sutton et al., 1998) provides a powerful tool for self-improvement of robotic agents. In Reinforcement Learning (RL), an agent interacts with the environment and learns independently with the goal of maximizing received rewards (Sutton and Barto, 2018). Such experience-based and reward-driven learning has also been studied with infants (Hämmerer and Eppinger, 2012) and in the past years has found more and more applications also in robotics (Kober et al., 2013; Kormushev et al., 2013).

However, while Reinforcement Learning has achieved in particular impressive results in simulated environments (Mnih et al., 2013, 2015) in real robotic applications Reinforcement Learning is often challenging due to the high sample complexity and the time and hardware costs of collecting them. Incorporating human feedback and advice into Reinforcement Learning systems offers hereby opportunities to speed up the learning process. Human input can in particular help to reduce the state space (Suay and Chernova, 2011), guide the exploration (Kuhlmann et al., 2004; Moreno et al., 2004) or provide a more informative reward structure (Thomaz et al., 2005; Knox and Stone, 2008).

However, such an interactive learning process, as illustrated in Figure 1.5, which involves human users rises new challenges regarding the design of novel interactive Reinforcement Learning algorithms and the best means of intuitive and optimal forms for human input from a human-centered point of view (Knox and Stone,

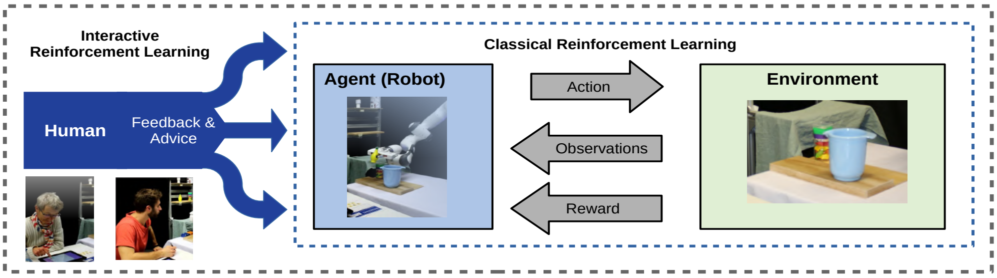


Figure 1.5.: In classical Reinforcement Learning an agent learns from its experiences in the environment and associated rewards. Interactive Reinforcement Learning includes human feedback and advice, in order to guide the agent's exploration and speed up the learning process, which is, in particular, beneficial for robotic applications.

2011; Loftin et al., 2014; Argall et al., 2008; Griffith et al., 2013; Abel et al., 2017; Grizou et al., 2013). In particular, human feedback and advice may often differ from assumptions of the classic RL-setting since input during the learning process may be subject to a positive bias (Thomaz and Breazeal, 2008; Knox and Stone, 2012), strongly vary between people or also simply be wrong sometimes (Amershi et al., 2014). It was also reported in the literature that humans do not only want to give positive or negative feedback but also actively propose actions (Kaochar et al., 2011; Thomaz and Breazeal, 2008), which motivates interactive Reinforcement Learning approaches that offer multiple input channels to users (Abel et al., 2017). Since many of such interactive Reinforcement Learning algorithms are only evaluated in simulated environments and with sometimes also only simulated human input it is thereby an important research direction to evaluate the use of human input in Reinforcement Learning in experiments with users on real robotic applications.

1.1.4. Research Questions

In order to contribute to the development of assistive robots which should be able to learn from human demonstrations, adapt learned skills to humans and improve themselves through human feedback, this thesis investigates the following research questions.

Incremental Learning of a Library for Cooperative Skills
In the first part of the thesis, we focus on a setting where an assistive robot is

supposed to learn a variety of different skills from a potentially open-ended stream of unlabeled human demonstrations. Hereby, we aim to answer

- How can an assistive robot learn and incrementally update a cooperative skill-library from human demonstrations, in an unsupervised setting?
- How do the learned cooperative skills transfer between different humans?

Adaptation to Humans in Shared-Workspaces
In shared workspaces, there is a need for online adaptation of learned skills to human-coworkers. Important related research questions are hereby

- How can an assistive robot adapt its learned skills to human co-workers in a shared human-robot workspace?
- How do humans perceive different ways of robot skill adaptation in such situations and what kind of adaptation do they prefer?

Interactive Reinforcement Learning for Sequencing of Skills
Even if a skill library is available to an assistive robot, there is still the need to learn how to sequence single skills to achieve complex tasks. While Reinforcement Learning provides a tool for independent learning and self-improvement in such situations, it is desirable to incorporate human input to accelerate and guide the learning process. Important research questions to answer are hereby

- How can an assistive robot include multiple forms of human feedback and advice during learning of sequential tasks and which types of input are preferred by users?
- Can the robot recognize if users give potentially wrong input, and how do humans react when the robot starts to question their input?

1.2. Main Contributions

This section summarizes the main contributions of the thesis. In particular, this thesis contributes to the areas of learning collaborative skill libraries from unlabeled human demonstrations, learning intention aware adaptation of robot skills in shared workspaces and including human feedback and advice in reinforcement learning for sequential tasks.

1.2.1. Incremental Learning of an Open-Ended Collaborative Skill Library

Intelligent assistive robots can potentially contribute to maintaining an elderly person's independence by supporting everyday life activities. However, the number of different and personalized activities to be supported renders pre-programming of all respective robot behaviors prohibitively difficult. Instead, to cope with a continuous and potentially open-ended stream of cooperative tasks, new collaborative robot behaviors need to be continuously learned and updated from demonstrations. To this end, in Chapter 3 we introduce an online learning method to incrementally build a cooperative skill library of probabilistic interaction primitives. The resulting model chooses a corresponding robot response to a human movement where the human intention is extracted from previously demonstrated movements. While existing batch learning methods for movement primitives usually learn such skill libraries only once for a pre-defined number of different skills, our approach enables extending the skill library in an open-ended and online fashion from new incoming demonstrations. The proposed approach is evaluated on a low-dimensional benchmark task and in a collaborative scenario with a 7DoF robot, where we also investigate the generalization of learned skills between different subjects.

1.2.2. Learning Intention Aware Online Adaptation of Movement Primitives

In order to operate close to non-experts, future robots require both an intuitive form of instruction accessible to laymen and the ability to react appropriately to a human co-worker. Instruction by imitation learning with Probabilistic Movement Primitives (ProMPs) allows capturing tasks by learning robot trajectories from demonstrations including the motion variability. However, appropriate responses to human co-workers during the execution of the learned movements are crucial for fluent task execution, perceived safety, and subjective comfort. To facilitate such appropriate responsive behaviors in human- robot interaction, the robot needs to be able to react to its human workspace co-inhabitant online during the execution of the ProMPs. In Chapter 4 we, therefore, introduce an approach for learning of intention aware adaptation of movement primitives. Hereby, we learn a goal-based intention prediction model from human motions. Using this probabilistic model, we introduce intention-aware online adaptation to ProMPs. We compare two different novel approaches: First, online spatial deformation, which avoids collisions by changing the shape of the ProMP trajectories dynamically during execution while staying close to the demonstrated motions and second, online temporal scaling, which adapts the velocity profile of a ProMP to avoid time-dependent collisions. We evaluate both approaches in experiments with non-expert users. The subjects reported a higher level of perceived safety and felt less disturbed during intention aware adaptation, in particular during spatial deformation, compared to a non-adaptive behavior of the robot.

1.2.3. Multi-Channel Interactive Reinforcement Learning for Sequential Tasks

The ability to learn new tasks by sequencing already known skills is an important requirement for future robots. Hereby, reinforcement learning offers a powerful tool to learn and self-improve how to combine skills for sequential tasks. However, in real robotic applications, the cost of sample collection and exploration prevent the application of reinforcement learning for a variety of tasks. To overcome these limitations, human input during reinforcement can be beneficial to speed up the learning, guide the exploration and prevent the choice of disastrous actions. Nevertheless, there is a lack of experimental evaluations of multi-channel interactive reinforcement learning systems on robotic tasks with inexperienced human users, in particular for cases where human input might partially be wrong.

Therefore, in Chapter 5, we present an approach that incorporates multiple human input channels for interactive reinforcement learning in an unified framework and evaluate it on two robotic tasks with 20 inexperienced human subjects. To enable the robot to also handle potentially incorrect human input we incorporate a novel concept for self-confidence, which allows the robot to question human input after an initial learning phase. The second robotic task is hereby specifically designed to investigate if this self-confidence can enable the robot to achieve learning progress even if the human input is partially incorrect. Hereby, we additionally evaluate how humans react to suggestions of the robot, once the robot notices human input might be wrong.

Our experimental evaluations show that our approach can successfully incorporate human input to accelerate the learning process in both robotic tasks even if it is partially wrong in the second task. However, not all humans were willing to accept the robot's suggestions or rejections of their input by the robot, in particular when they did not fully understand the learning process and the reasons behind the robot's suggestions. We believe findings from this experimental evaluation can be in particular beneficial for the future design of algorithms and interfaces for interactive reinforcement learning applications with inexperienced users.

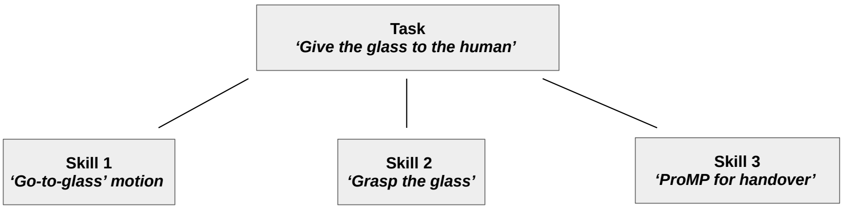


Figure 1.6.: We define a hierarchical structure where a more complex behavior, i.e. a task, is composed out of different skills. Skills can hereby be either preprogrammed basic motions of the robot or learned e.g from human demonstrations.

1.3. Necessary Background

In this section, we provide definitions for the terms skill and task to clarify how these terms will be used throughout the thesis. Additionally, we give a short summary of the foundations of Probabilistic Movement Primitives, which will be used as a skill representation in this thesis.

1.3.1. Definition for Skills and Tasks

Since there is no unified definition of the terms skill for a robot in the literature we consider it important to clarify how we will use this term throught the thesis. Following similar hierarchical concepts for task definition in the literature (Zoliner et al., 2005; Konidaris et al., 2012), we define a *skill* as a basic part of a more complex behavior. One main property of a skill is hereby that it is a modular building block for a complex behavior and might be reusable across different tasks. In order to achieve a more complex behavior, that we will refer to as *task* there is the need to combine the available skills. Figure 1.6 illustrates an example of a task and the decomposed skill structure. We assume that skills can either be preprogrammed such as go-to-point motions, fixed gripper or hand positions, or can be learned e.g. from human demonstrations. In this thesis learned skills are represented by movement primitives. Probabilistic Movement Primitives as a representation for learning skills from demnstrations are discussed in the next section.

1.3.2. Probabilistic Movement Primitives

Probabilistic Movement Primitives (ProMPs) (Paraschos et al., 2018) represent demonstrated movements in the form of distributions over trajectories. In order to obtain this distribution, each trajectory $\boldsymbol{\tau}$ is first approximated by a linear combination of basis functions ϕ. Hereby, it is assumed that the trajectories are sampled from the conditional distribution

$$p(\boldsymbol{\tau}|\boldsymbol{w}) = \prod_t \mathcal{N}(\boldsymbol{\tau}_t|\boldsymbol{\phi}_t^{\mathrm{T}}\boldsymbol{w}, \boldsymbol{\Sigma}_{\mathrm{o}}), \tag{1.1}$$

where $\boldsymbol{w}$ denotes the weight vector and $\boldsymbol{\Sigma}_{\mathrm{o}}$ the observation noise.
More precisely, a joint or task space position $\boldsymbol{\tau}_t$ at time step t can be represented as

$$\tau_t = \boldsymbol{\phi}_t^{\mathrm{T}}\boldsymbol{w} + \sigma_{\mathrm{o}}, \tag{1.2}$$

where $\boldsymbol{\phi}_t$ contains N basis functions ϕ evaluated at time step t, $\boldsymbol{w}$ is a weight vector, σ_{o} is a zero-mean Gaussian noise and $[]^{\mathrm{T}}$ denotes transposition of the vector. The choice of basis functions ϕ depends on the type of demonstrated movements. Throughout this thesis we use radial basis functions. The weight vector $\boldsymbol{w}$ for each demonstrated trajectory is computed with Ridge Regression

$$\boldsymbol{w} = (\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}} + \lambda \boldsymbol{I})^{-1}\boldsymbol{\Phi}\boldsymbol{\tau}, \tag{1.3}$$

where $\boldsymbol{\Phi} = [\boldsymbol{\phi}_0, ..., \boldsymbol{\phi}_t, ..., \boldsymbol{\phi}_T]$ contains all basis functions at at the respective timesteps, λ is a small factor e.g. 1e-6 and $\boldsymbol{I}$ denotes an identity matrix. For multiple recorded demonstrated trajectories, and their respective weight vector, a distribution over the weight vectors $p(\boldsymbol{w})$ can then be obtained with Maximum Likelihood Estimation. Since the number N of basis functions is usually much lower than the number of time steps of recorded trajectories, the distribution $p(\boldsymbol{w})$ can be seen as a compact representation of the demonstrated movements, which accounts for variability in the execution. By choosing $p(\boldsymbol{w})$ to be a Gaussian,

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{w}}, \boldsymbol{\Sigma}_{\boldsymbol{w}}) \tag{1.4}$$

we ensure the distribution over trajectories can be computed in closed form by integrating out the weight vectors $\boldsymbol{w}$. This probabilistic representation offers a number of convenient properties. In particular, ProMPs offer a representation that allows for operations from probability theory to specify goal or via-points, correlate different degrees of freedom via conditioning and combine different primitives through blending (Paraschos et al., 2018).

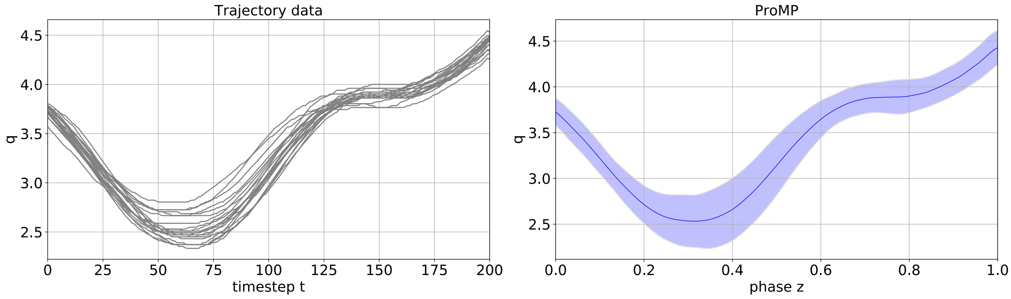


Figure 1.7.: ProMPs offer a probabilistic representation for distributions over trajectories. (a) shows multiple example demonstrated trajectories, (b) shows the resulting ProMP with the mean and two times standard deviation plotted over the phase of execution.

Figure 1.7 shows an example distribution over trajectories on the left and the obtained ProMP on the right. Hereby, it should be noted that while the trajectory data is plotted over the number of timesteps, the ProMP is visualized over the phase, that is a concept of normalized time.

1.4. Outline

Subsequent to the introduction this thesis is structured in five main chapters. Figure 1.8 visualizes this structure of the thesis. The presented research has hereby led to a number of publications. Here, we provide an overview of the content of the following chapters and how they are linked to the different publications.

Chapter 2 introduces our approach for incremental learning of a skill library for collaborative tasks. Hereby, we use Incremental Gaussian Mixture Models to learn cooperative skills, from unlabeled demonstrations in an open-ended fashion. The chapter is based on Koert et al. (2018, 2019b).

In **Chapter 3**, we present two novel methods for online adaptation of Probabilistic Movement Primitives to human co-workers. In particular, our approach incorporates a goal-directed model to predict human motions and evaluates the two proposed adaptation strategies in experiments with human subjects. The content of the chapter is hereby based on Koert et al. (2019a).

Chapter 4 introduces our approach to learn sequential tasks using interactive reinforcement learning. Hereby, we investigate the benefits of different input channels for human users and a novel concept for self-confidience that enables the robot to question human input if it controdicts the robots learned policy after an initial training time. The content of this chapter is based on Koert et al. (2020).

Chapter 5 provides a summary and conclusion of the thesis. Additionally, the chapter gives an outlook on potentially interesting future research directions that emerge out of the topics of this thesis, namely skill transfer, preference based adaptation, predicting human behavior and intentions, multimodal communication and interaction channels, explainability and transperency and experiments with assistive robots in the wild.

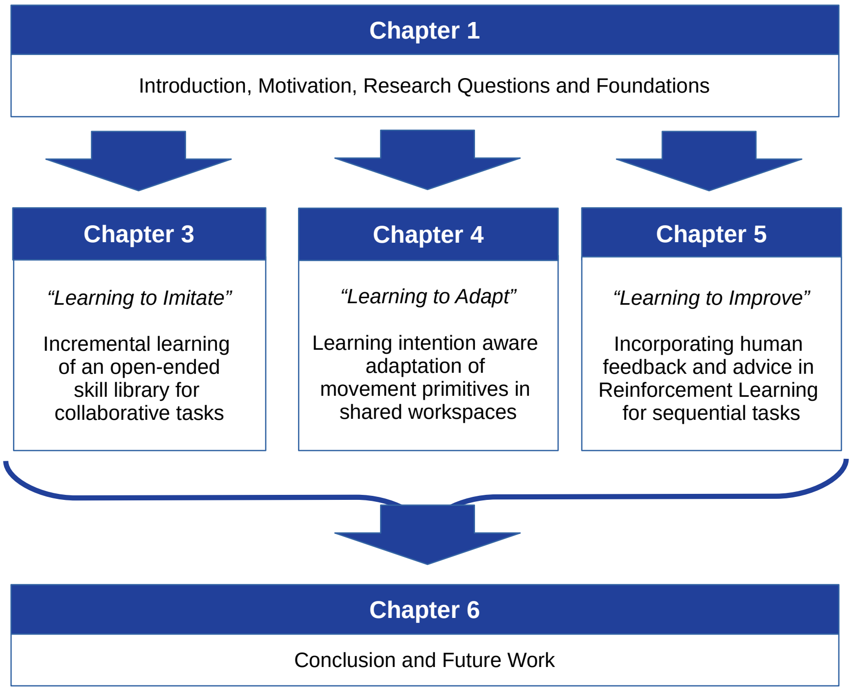

Chapter 1
Introduction, Motivation, Research Questions and Foundations
Chapter 3
"Learning to Imitate"
Incremental learning of an open-ended skill library for collaborative tasks
Chapter 4
"Learning to Adapt"
Learning intention aware adaptation of movement primitives in shared workspaces
Chapter 5
"Learning to Improve"
Incorporating human feedback and advice in Reinforcement Learning for sequential tasks
Chapter 6
Conclusion and Future Work

Figure 1.8.: Outline of this thesis

2. Incremental Learning of an Open-Ended Collaborative Skill Library

The expected demographic change is an urgent and prevailing challenge for society. An increasing number of elderly people need assistance in their daily lives while only few caregivers are available (Linz and Stula, 2010). In order to address this challenge, the development of technical solutions for elderly assistance is essential. In particular, assistive robots can potentially support elderly people in their daily lives and thus help to maintain their personal independence. Such cooperative robot assistants need the ability to support multiple different tasks and additionally should be able to individually adapt to a user's needs. This disqualifies pre-programming of all possible tasks. Instead, an intuitive way to teach multiple personalized skills to a robot is needed, which can be realized by Learning from Demonstrations (LfD) (Schaal, 1999).

Since human motions exhibit a high variability (Rosenbaum, 2009), a probabilistic approach for cooperative skill learning from demonstrations is required, which takes variations in human motions during demonstrations and during execution time into account. In addition, to be able to learn cooperative skills from multiple human demonstrations and representing them probabilistically, it is not just crucial to detect multiple different skills but particularly to update an existing skill library with completely new cooperative skills and refining the existing skills with new demonstrations.

This chapter presents a novel online learning method for building a collaborative skill library, which enables open-ended learning of new skills and refinement of existing skills. Figure 2.1 summarizes the approach. An incremental mixture model of Probabilistic Movement Primitives is proposed for online learning of collaborative skills from demonstrations. Cooperative skills are represented as Interaction ProMPs (Amor et al., 2014; Maeda et al., 2014) which capture correlations between human and robot movements as well as the inherent variance.

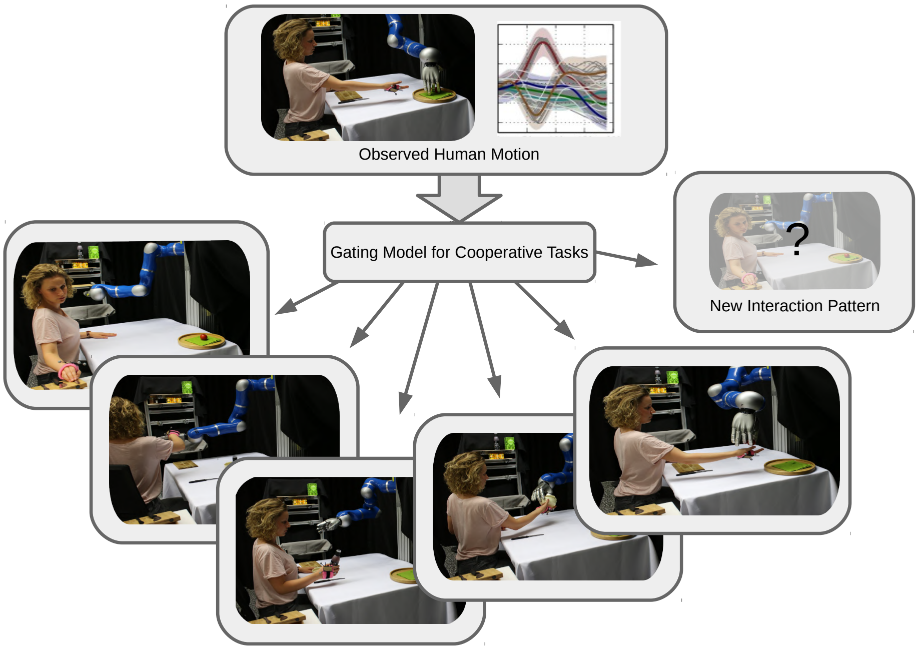


Figure 2.1.: Intelligent robot assistants should learn multiple personalized cooperative skills from a continuous and open-ended stream of new demonstrations. To this end, we propose a novel approach for online and open-ended learning of a mixture model of Probabilistic Interaction Primitives. In particular, our approach updates existing cooperative skills from new demonstrations and extends the collaborative skill library for new skills when needed. Hereby, our model chooses a robot response to an observed human motion based on prior demonstrations while considering variance in the demonstrations as well as coupling between human and robot motions.

Although these movement primitives have already been used for learning multiple different cooperative skills (Ewerton et al., 2015), to the best of our knowledge, none of the present approaches is able to iteratively learn multiple Interaction ProMPs online and in an open-ended way. However, for personalized robot assistants it is crucial to open-endedly learn new tasks and continuously update existing cooperative skills with new demonstrations. In particular, in such an open-ended scenario the total number of cooperative skills cannot be known beforehand and thus needs to be extended during the learning process.

In contrast to prior work on learning a Mixture of Interaction primitives (Ewerton et al., 2015), our new approach does not rely solely on demonstrations which are available at the first training time but can integrate new demonstrations and skills over multiple training sessions. In addition to the experimental validation with single subjects, we evaluate our method on learning a library across multiple subjects. To enable successful incremental learning of a skill library on data of multiple subjects we introduce aging out and consolidation of model components and an automated normalization routine based on personal workspace and start position of the subjects' motions.

The rest of this chapter is structured as follows: Section 2.1 discusses related work. Section 2.2 provides an overview on the existing approach of Batch Learning for a Mixture of Interaction Primitives and introduces our novel approach for Online Open-Ended Learning of a Mixture of Interaction Primitives. In Section 2.3, we evaluate this new approach on 2D trajectory data of hand written letters and in a collaborative robotic scenario. In the robotic scenario, we additionally investigate how the learned skills transfer between different subjects. Finally, we conclude with Section 2.4 and discuss ideas for future work.

2.1. Related Work

Learning cooperative tasks between humans and robots from demonstration is a popular approach as it enables also non-expert users to teach personalized skills to robots (Argall et al., 2009; Billard et al., 2008; Pastor et al., 2009; Vogt et al., 2017; Maeda et al., 2017b). While a number of existing learning from demonstration literature focusses on task representation for a single agent for collaborative tasks it is also desirable to address coupled skill representations e.g. for human and robot motions (Amor et al., 2013; Vogt et al., 2015; Nikolaidis et al., 2015). In particular, the concept of movement primitives offers hereby a lower dimensional representation of trajectories and a modular framework that does not only reproduce demonstrated behaviors but potentially also generalizes to new situations (Ijspeert et al., 2013; Calinon et al., 2007; Paraschos et al., 2018; Vogt et al., 2017; Amor et al., 2013; Huang et al., 2019) To capture also variability in non-deterministic human motion demonstrations it is desirable to not only model demonstrations with single trajectories (Ijspeert et al., 2013) but to also capture the variability using a probabilistic approach and distributions over trajectories (Calinon et al., 2007; Paraschos et al., 2018; Vogt et al., 2017). Probabilistic Interaction Movement

Primitives (Interaction ProMPs) (Amor et al., 2014; Maeda et al., 2014, 2017b) offer hereby a probabilistic representation to model inherent correlations in the movements of two actors, such as human and robot, from coupled demonstrated trajectories while showing beneficial interpolation capabilities compared to other movement primitive approaches (Paraschos et al., 2018).

However, to achieve a personalized cooperative robot it is desirable to learn multiple cooperative tasks and decide on their activation depending on the context or on the human intention (Pérez-D'Arpino and Shah, 2015; Lee et al., 2010; Konidaris et al., 2012). To this end, an approach that deploys Gaussian Mixture Models (GMMs) and Expectation Maximization to learn multiple Interaction ProMPs from unlabeled demonstrations has been introduced (Ewerton et al., 2015). This approach considers batch data, i.e. assuming the availability of all data during training. This limits its application to settings where the number of tasks does not change after training and no new demonstration trajectories need to be integrated. Moreover, such batch learning prevents scalability as computation time and memory requirements become infeasible for large skill libraries or datasets (Calinon and Billard, 2007). Various approaches outside the Human-Robot Interaction (HRI) scope have addressed these problems.

Initially, the machine learning communities have proposed incremental learning approaches for Gaussian Mixture Models. Some approaches propose updating a GMM with complete new model component datasets (Ahmed and Xing, 2008) or assume the incoming data points to be time-coherent (Arandjelovic and Cipolla, 2006). Incremental Gaussian Mixture Model learning introduced a way to continuously learn a GMM from an incoming data stream while not fixing the number of total components beforehand (Engel and Heinen, 2010; Pinto and Engel, 2015). Another two-level approach introduces methods for splitting and merging of GMM components (Declercq and Piater, 2008). Updating of robotic movement representations online from new demonstrations has also been used for incremental learning of extensions of GMMs for gesture imitation (Calinon and Billard, 2007), updating Gaussian Processes from demonstrations and thereby reducing the movement variance (Maeda et al., 2017a) or incremental updating of task-parameterized Gaussian Mixture Models (Hoyos et al., 2016). While all these works focus on updating multiple existing movement representations, in a long-term setting adding new tasks is also important. Approaches that also add new components when needed have been proposed in the context of online updating of task-parameterized semi-tied hidden semi-Markov models for manipulation tasks (Havoutis et al., 2016), learning full-body movements (Kulić et al., 2012), a bootstrapping cycle for automatic extraction of primitives from complex trajectories (Lemme et al., 2014) or robot table tennis (Muelling et al., 2010). However, while we draw inspiration from the aforementioned related work, in an HRI scenario it is additionally desirable to consider the inherent coupling and variance in human and robot motions in the demonstrations.

2.2. Incremental Interaction Primitives

In this section, we present our approach to continuously learn and update multiple cooperative skills from demonstrations. Here, demonstrations are given in the form of coupled human and robot trajectories $\boldsymbol{d}^n = \{\boldsymbol{\tau}_h^n, \boldsymbol{\tau}_r^n\}$, where $\boldsymbol{\tau}_h^n$ can e.g. be a sequence of human wrist positions and $\boldsymbol{\tau}_r^n$ can e.g. be a sequence of robot joint positions. To learn multiple cooperative skills from these demonstrations in an online open-ended fashion we introduce a model that is inspired by the Mixture of Experts architecture (Jacobs et al., 1991) and consists of two intertwined parts. On the one hand, we use the human trajectories from the demonstrations to train and update a gating model, which will later be used to decide between different cooperative skills. In addition, we train probabilistic models to generate appropriate robot response trajectories. We deploy Interaction ProMPs (Amor et al., 2014; Maeda et al., 2014), as they are able to capture the inherent correlation in robot and human motions from the demonstrations. Figure 2.2 summarizes our approach to train this mixture model in an online and open-ended fashion.
In the following, we briefly describe the previously proposed batch-based, stationary Mixture of Interaction ProMPs in Section 2.2.1. Next, we present our novel approach to learn a mixture model of Probabilistic Movement Primitives in an online and open-ended fashion in Section 2.2.2. Finally, in Section 2.2.3 we show how the obtained library of multiple Interaction ProMPs and the corresponding gating model can be deployed in an HRI scenario.

2.2.1. Batch Learning for Mixture of Interaction ProMPs

An Interaction ProMP (Amor et al., 2014; Maeda et al., 2014) is a ProMP that uses a distribution over the trajectories of at least two interacting agents. The demonstrations are now given in the form of a stacked vector for the observed and the controlled agent $\boldsymbol{q} = [\boldsymbol{q}^o, \boldsymbol{q}^c]^T$, where $\boldsymbol{q}^o$ denotes the demonstrated trajectories for the observed agent and $\boldsymbol{q}^c$ denotes the demonstrated trajectories of the controlled agent. Respectively, the weight vector is also represented in an augmented form $\bar{\boldsymbol{w}} = [\boldsymbol{w}_o^T, \boldsymbol{w}_c^T]^T$. Given a set of demonstrations, a distribution over multiple stacked weight vectors can be obtained just as previously described such that

$$p(\bar{\boldsymbol{w}}) = \mathcal{N}(\boldsymbol{\mu}_{\bar{\boldsymbol{w}}}, \boldsymbol{\Sigma}_{\bar{\boldsymbol{w}}}). \tag{2.1}$$

Given a sequence $\mathcal{D}$ of positions of the observed agent (e.g. human), Interaction ProMPs (Amor et al., 2014; Maeda et al., 2014) provide methods to infer a corresponding (most likely) trajectory of the controlled agent (robot).

The previously proposed batch learning for Mixture of Interaction ProMPs (Ewerton et al., 2015) is an extension to Interaction ProMPs that allows to learn several different interaction patterns from unlabeled demonstrations by applying Gaussian Mixture Models (GMMs), where each mixture component represents one interaction pattern. The Mixture of Interaction Primitives is hereby learned from batch data and the number of components needs to be fixed beforehand. In the case of K different interaction patterns, the distribution over the weight vectors $\bar{\boldsymbol{w}}$ is

$$\begin{aligned} p(\bar{\boldsymbol{w}}) &= \sum_{k=1}^{K} p(k) p(\bar{\boldsymbol{w}}|k) \\ &= \sum_{k=1}^{K} \alpha_k \, \mathcal{N}(\bar{\boldsymbol{w}}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \end{aligned} \tag{2.2}$$

where α_k is the k-th mixture weight that can be prior (if not learned) or posterior (if learned from given data), $\boldsymbol{\mu}_k$ is the mean and $\boldsymbol{\Sigma}_k$ the covariance matrix of the k-th component. The parameters of the GMM are hereby learned in the weight space using the Expectation Maximization (EM) algorithm. Since this approach assumes that all data is available at the learning time the number of components K remains fixed after learning and the model cannot easily integrate new demonstrations. This means that if new demonstrations come in the GMM can not be updated without the need for full recomputation nor can new interaction patterns be included in the model if they occur.

2.2.2. Online Open-Ended Mixture of Interaction ProMPs

We propose a new method to achieve online learning of cooperative skills in an open-ended fashion. Hereby, demonstrations are given in the form of robot and human trajectories $\{\boldsymbol{\tau}_r, \boldsymbol{\tau}_h\}$. First, we compute a corresponding representation with weight vectors as introduced in Section 2.2.1. Here, we consider that the human trajectory is of dimensions $D_h \times T$ where D_h is the degree of freedom of the observations (e.g. in case of observing the wrist position $D_h = 3$). T denotes

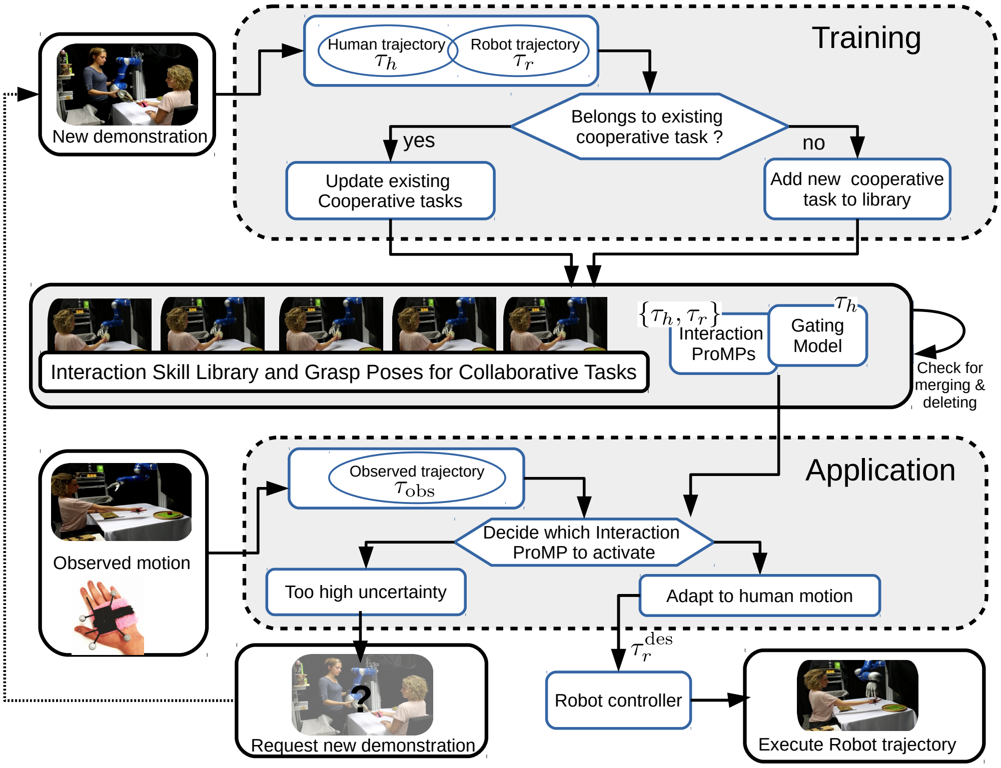

Training
New demonstration
Human trajectory τ_h
Robot trajectory τ_r
Belongs to existing cooperative task ?
yes
no
Update existing Cooperative tasks
Add new cooperative task to library
Interaction Skill Library and Grasp Poses for Collaborative Tasks
$\{\tau_h, \tau_r\}$
Interaction ProMPs
τ_h
Gating Model
Check for merging & deleting
Application
Observed motion
Observed trajectory τ_{obs}
Decide which Interaction ProMP to activate
Too high uncertainty
Adapt to human motion
τ_r^{des}
Request new demonstration
Robot controller
Execute Robot trajectory

Figure 2.2.: We introduce a novel approach for online and open-ended learning of a mixture model for cooperative skills. During training, demonstrations are given in the form of trajectories of a human demonstrator τ_h and corresponding trajectories of a robot arm τ_r, that are obtained via motion capturing and kinesthetic teaching. From these demonstrations, we update or extend the skill library that consists of a gating model and multiple corresponding Interaction ProMPs. During runtime, the gating model decides on the activation of particular Interaction ProMPs that we subsequently adapt to the variance in the observed motion. If the gating model is too uncertain about the activation of Interaction ProMPs the robot can request more demonstrations.

the number of time steps and the robot trajectory is of dimensions $D_r \times T$ where D_r is the degrees of freedom of the robot (e.g. $D_r = 7$ in case of a 7DoF robot arm). For N basis functions ϕ we compute the matrix $\mathbf{\Phi} = [\phi_0, ...\phi_t, ..., \phi_T]$ with dimension $N \times T$. In this work, Gaussian basis functions evenly spaced along the time axis are an appropriate choice due to the stroke-based movements.

We then compute the weight vectors as a lower dimensional representation of the trajectories where we first compute the weight vectors for each dimension $\tilde{\boldsymbol{w}}$ as

$$\left[\tilde{\boldsymbol{w}}_1^h, ..., \tilde{\boldsymbol{w}}_{D_h}^h, \tilde{\boldsymbol{w}}_1^r, ..., \tilde{\boldsymbol{w}}_{D_r}^r\right]^{\mathrm{T}} = (\mathbf{\Phi}\mathbf{\Phi}^T + \beta \boldsymbol{I})^{-1}\mathbf{\Phi}\left[\boldsymbol{\tau}_h, \boldsymbol{\tau}_r\right]^{\mathrm{T}} \tag{2.3}$$

where β is a factor for Ridge Regression and $\boldsymbol{I}$ is an identity matrix. In experimental evaluation, we found that normalizing the trajectory data within a fixed range before transforming it into the weight space yields overall better results. Subsequently, we compute the stacked weight vectors

$$\boldsymbol{w}_h = [\tilde{\boldsymbol{w}}_1^h, ..., \tilde{\boldsymbol{w}}_{D_h}^h] \quad \text{and} \qquad \boldsymbol{w}_r = [\tilde{\boldsymbol{w}}_1^r, ..., \tilde{\boldsymbol{w}}_{D_r}^r]. \tag{2.4}$$

From these demonstrations, now represented in form of $\{\boldsymbol{w}^r, \boldsymbol{w}^h\}$, we learn the two intertwined parts of our model: The gating model that decides on the cooperative skills based on human motions and multiple corresponding Interaction ProMPs that can subsequently generate a corresponding robot response. For the gating model we train a Gaussian Mixture Model (GMM) only on the weights of the human trajectories $\boldsymbol{w}^h$, as at runtime only the human motion will be observed when the system needs to decide on the particular cooperative skill and the response of the robot.

In parallel to the gating model, the corresponding Interaction ProMPs are trained with the augmented weight vector $\bar{\boldsymbol{w}}$ of human and robot trajectories to model the correlations in the motions. We assume that new training data needs to be integrated continuously and that we do not know beforehand the number of different collaborative skills that might be shown to the robot during long-term training. To this end, we use Incremental Gaussian Mixture Models (Engel and Heinen, 2010) to achieve the continuous integration of new demonstrations. Here, we update the gating model and the parameters of the Interaction ProMPs in an Expectation Maximization fashion.

In the Expectation step, we compute the responsibilities λ_{kn} of the existing cooperative skill k for a new demonstration $\{\boldsymbol{w}_n^h, \boldsymbol{w}_n^r\}$, that is the probability of a new demonstration to belong to an already known cooperative skill

$$\lambda_{kn} := p(k|\boldsymbol{w}_n^h) = \frac{p(k)p(\boldsymbol{w}_n^h|k)}{p(\boldsymbol{w}_n^h)} \\ = \frac{\alpha_k \mathcal{N}(\boldsymbol{w}_n^h|\boldsymbol{\mu}_k^g, \boldsymbol{\Sigma}_k^g)}{\sum_{j=1}^K \alpha_j \mathcal{N}(\boldsymbol{w}_n^h|\boldsymbol{\mu}_j^g, \boldsymbol{\Sigma}_j^g)}, \tag{2.5}$$

where $\boldsymbol{\mu}_k^g$ and $\boldsymbol{\Sigma}_k^g$ are respectively the mean and covariance matrix of the k-th component of the gating model and α_k are the mixture component weights.
In the Maximization step, we use the responsibilities to recursively update the parameters of the gating model as well as the parameters of the already learned Interaction ProMPs. For each already learned Interaction ProMP k we first compute

$$v_k = v_k + 1, \qquad s_k = s_k + \lambda_{kn}, \\ \gamma_k = \frac{\lambda_{kn}}{s_k}, \qquad \tilde{\gamma}_k = \gamma_k + \exp\left(-s_k\right)\lambda_{kn}, \tag{2.6}$$

where v_k is the age of the k-th component and s_k represents the amount of trajectories the component already modeled well. We then update the parameters of the gating model

$$\boldsymbol{\mu}_k^g = \boldsymbol{\mu}_k^g + \gamma_k(\boldsymbol{w}_n^h - \boldsymbol{\mu}_k^g), \\ \boldsymbol{C}_k^g = (1-\tilde{\gamma}_k)\boldsymbol{C}_k^g + \tilde{\gamma}_k(\boldsymbol{w}_n^h - \boldsymbol{\mu}_k^g)(\boldsymbol{w}_n^h - \boldsymbol{\mu}_k^g)^T - (\tilde{\gamma}_k - \gamma_k)(\boldsymbol{\mu}_k^g - \boldsymbol{\mu}_k^{g,\text{old}})(\boldsymbol{\mu}_k^g - \boldsymbol{\mu}_k^{g,\text{old}})^T, \\ \alpha_k = \frac{s_k}{\sum_{j=1}^K s_j} \tag{2.7}$$

where $\boldsymbol{\mu}_k^{g,\text{old}}$ denotes the mean of the gating model before the update. The formulas correspond to the formulas in the incremental GMM (Engel and Heinen, 2010), except that we introduce $\tilde{\gamma}_k$ to achieve that during the first demonstrations the covariance is shifted faster away from the (possibly wrong) initialization. Additionally, we compute the updated parameters of the corresponding Interaction ProMPs

$$\boldsymbol{\mu}_k^e = \boldsymbol{\mu}_k^e + \gamma(\bar{\boldsymbol{w}}_n - \boldsymbol{\mu}_k^e), \tag{2.8}$$
$$\boldsymbol{C}_k^e = (1-\tilde{\gamma}_k)\boldsymbol{C}_k^e + \tilde{\gamma}_k(\bar{\boldsymbol{w}}_n - \boldsymbol{\mu}_k^e)(\bar{\boldsymbol{w}}_n - \boldsymbol{\mu}_k^e)^T - (\tilde{\gamma}_k - \gamma_k)(\boldsymbol{\mu}_k^e - \boldsymbol{\mu}_k^{e,\text{old}})(\boldsymbol{\mu}_k^e - \boldsymbol{\mu}_k^{e,\text{old}})^T,$$

where $\boldsymbol{\mu}_k^e$ is the mean of the k-th Interaction ProMP and $\boldsymbol{\Sigma}_k^e$ is the covariance matrix of the k-th Interaction ProMP. Whenever $p(\boldsymbol{w}_n^h|k)$ is below a threshold T_{nov}

for all existing K components we initialize a new component with

$$\begin{aligned} \boldsymbol{\mu}^g_{K+1} =& \boldsymbol{w}^h_n, & \boldsymbol{\Sigma}^g_{K+1} &= \boldsymbol{\Sigma}^g_{\text{init}}, \\ \boldsymbol{\mu}^e_{K+1} =& \boldsymbol{w}_n, & \boldsymbol{\Sigma}^e_{K+1} &= \boldsymbol{\Sigma}^e_{\text{init}}, \\ v_{K+1} =& 1, & s_{K+1} &= 1, \end{aligned} \tag{2.9}$$

where $\boldsymbol{\Sigma}^g_{\text{init}}$ and $\boldsymbol{\Sigma}^e_{\text{init}}$ denote the initial covariance matrix of the gating model and the experts that can e.g. be initialized as identity matrix. If a component has reached a certain age $v_k > v^{\text{merge}}_{\text{min}}$ we check also for merging of components to ensure that no unnecessary components are maintained. Therefore, we compute the probability of the mean of a cluster j to belong to a cluster i as $p(\boldsymbol{\mu}_j|i) = \mathcal{N}(\boldsymbol{\mu}_j|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ and decide on merging if

$$p(\boldsymbol{\mu}_j|i) > T^k_{\text{merge}}. \tag{2.10}$$

The threshold for merging is hereby initialized equally with T_{merge} for all new components but increases as a component absorbs more demonstrations, such that already well consolidated components are not so easily merged.

$$T^k_{\text{merge}} = T_{\text{merge}} s^{\beta}_k, \tag{2.11}$$

where β is a factor that controls how fast component gets consolidated. Once we determined candidates i, j for merging we recompute the joined mean and covariance

$$\begin{aligned} \boldsymbol{\mu}_{ij} =& \frac{s_i \boldsymbol{\mu}_i + s_j \boldsymbol{\mu}_j}{s_i + s_j}, \\ \boldsymbol{\Sigma}_{ij} =& \frac{s_i^2 \boldsymbol{\Sigma}_i + s_j^2 \boldsymbol{\Sigma}_j + (s_i \boldsymbol{\mu}_i + s_j \boldsymbol{\mu}_j) s_i \boldsymbol{\mu}_i + s_j \boldsymbol{\mu}_j^T}{(s_i + s_j)^2} - \boldsymbol{\mu}_{ij} \boldsymbol{\mu}_{ij}^T. \end{aligned} \tag{2.12}$$

We also include a mechanism to delete components created by outliers. Here, we delete all components that do not reach a certain support $s_k > s_{\text{min}}$ after an age $v_k > v^{\text{del}}_{\text{min}}$, such that such outlier components age out over time.
Algorithm 1 summarizes our approach for online learning of a gating model and multiple Interaction ProMPs.

Algorithm 1 Incremental Skill Learning

```
input: $\boldsymbol{\Sigma}^g_{\text{init}} = \boldsymbol{I}, T_{\text{nov}}, v_{\text{min}}$
while new data $\boldsymbol{\tau}^h_n$,$\boldsymbol{\tau}^r_n$ do
    normalize data , Eq. (2.15)
    compute $\boldsymbol{w}^h_n$,$\boldsymbol{w}^r_n$ from $\boldsymbol{\tau}^h_n$,$\boldsymbol{\tau}^r_n$ , Eq. (2.3), Eq. (2.4)
    compute $p(\boldsymbol{w}^h_n|k)\ \forall k$
    if $p(\boldsymbol{w}^h_n|k) < T_{\text{nov}}\ \forall k$ then
        add new component , Eq. (2.9); k++
    else
        compute $p(k|\boldsymbol{w}^h_n)\ \forall k$ , Eq. (2.5).
        update $\forall k$ Eq. (2.6), Eq. (2.7), Eq. (2.8)
        if $v_k > v_{\text{min}}$ then
            check for merge, Eq. (2.11), Eq. (2.12)
```

2.2.3. A Skill Library for Collaborative Tasks

To demonstrate the use of the learned probabilistic mixture model for cooperative skills we assume we are now observing the human and obtain an observation $\boldsymbol{w}^h_*$. To determine the most probable cluster given the observations we need to model the posterior of the cluster given the observation $p(k|\boldsymbol{w}^h_*) = \lambda_{k*}$, where λ_{k*} is the responsibility of the k-th cluster for the observation $\boldsymbol{w}^h_*$ as defined in Equation (2.5). For an observation $\boldsymbol{w}^h_*$ we can now infer the most likely Interaction ProMP k^* using our probabilistic gating model

$$k^* = \arg\max_k \quad p(k|\boldsymbol{w}^h_*). \tag{2.13}$$

If the responsibility of all components is smaller than the novelty threshold T_{nov} the robot does not execute a response but asks the user for new demonstrations that get subsequently included in the library as described in Section 2.2.2. Otherwise, we condition the chosen component on the observed trajectory to infer the corresponding robot response. Here, the observation $\boldsymbol{o}$ is used to obtain a new Gaussian posterior distribution over the weights, with mean $\boldsymbol{\mu}_{new}$ and covariance matrix $\boldsymbol{\Sigma}_{new}$

$$\begin{aligned}\boldsymbol{\Lambda} &= \boldsymbol{\Sigma}_{\boldsymbol{k}^*}\boldsymbol{H}_t(\boldsymbol{\Sigma}_{\boldsymbol{o}} + \boldsymbol{H}_t^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{k}^*}\boldsymbol{H}_t)^{-1},\\ \boldsymbol{\mu}_{new} &= \boldsymbol{\mu}_{\boldsymbol{k}^*} + \boldsymbol{\Lambda}(\boldsymbol{o} - \boldsymbol{H}_t^{\mathrm{T}}\boldsymbol{\mu}_{\boldsymbol{k}^*}),\\ \boldsymbol{\Sigma}_{new} &= \boldsymbol{\Sigma}_{\boldsymbol{k}^*} - \boldsymbol{\Lambda}\boldsymbol{H}_t\boldsymbol{\Sigma}_{\boldsymbol{k}^*},\end{aligned} \tag{2.14}$$

where $\boldsymbol{\Sigma_o} = \boldsymbol{I}\sigma_o$ is the observation noise and $\boldsymbol{H}_t$ is the observation matrix as defined in (Maeda et al., 2014). More details can be found in (Maeda et al., 2017b). To obtain a corresponding robot motion we execute the mean robot trajectory of this posterior.

2.3. Experimental Evaluation

We evaluate our approach on 2D trajectory data and on a collaborative scenario with a 7DoF robot arm. For both, we show the qualitative applicability and evaluate the quantitative convergence w.r.t. to a baseline. We demonstrate that the proposed approach can learn personalized libraries for collaborative skills for different persons and report successful task completion via the decision accuracy of our gating model.

2.3.1. 2D trajectory data

In this section we demonstrate the application of the proposed incremental learning method on a non-robotic 2D task for letter aquisition. On this 2D data we visualize the incremental learning process and compare the results to the batch solution of the EM based method. For the 2D trajectory data experiment, demonstrations are given in the form of multiple hand-drawn letters, as illustrated in Figure 2.4 (a). All data is normalized to the range of [0,10] and we apply aequidistant spatial interpolation to decouple the trajectory representation from the varying speed of the demonstrations. Here, we incrementally learn a library of ProMPs. The system never has access to the whole training dataset at once, but only one new unlabeled demonstration is provided at each update step. The general procedure is shown in Figure 2.3, where the upper row shows the x-dimension of the learned library and the lower row the accumulated demonstrations. Initially, a single "a" is demonstrated and the first skill is added, with the initial covariance $\boldsymbol{\Sigma}_{init}$. Additional "a"s are demonstrated, recognized and used to update the mean and covariance of the corresponding cluster. Once a new demonstration is recognized to not belong to the existing cluster a new cluster is generated.

With an increasing number of samples, the variance converges to the variance of the demonstrations as the impact of the initialization covariance decreases. The final skill library consists of five clusters representing the different letters. Please note that in this experiment $\boldsymbol{\mu}_k^g, \boldsymbol{\Sigma}_k^g = \boldsymbol{\mu}_k^e, \boldsymbol{\Sigma}_k^e$. We evaluate the approach in a collaborative setting later in Section 2.3.2.

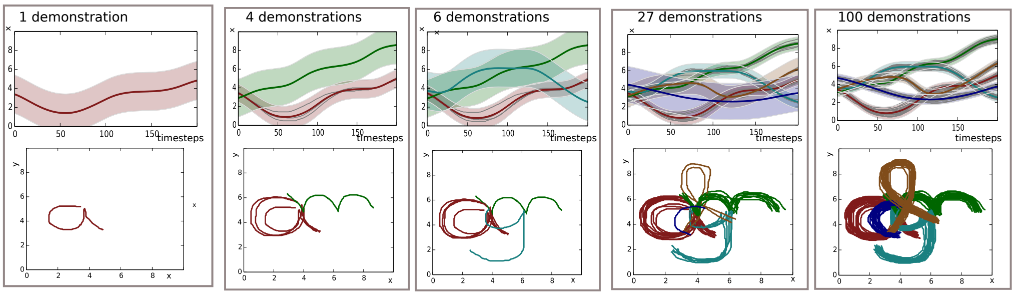
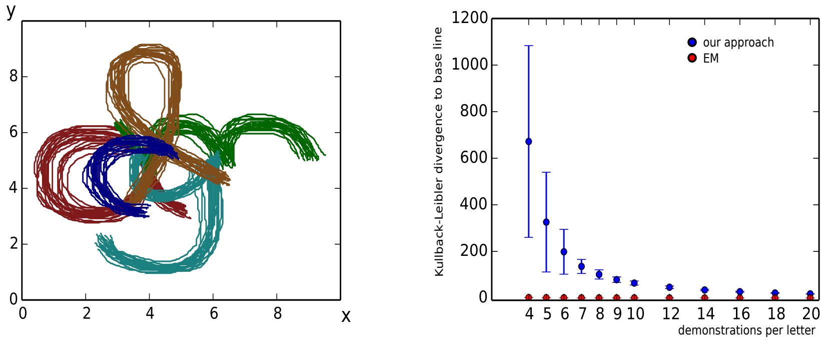


Figure 2.3.: Learning ProMPs of hand-drawn letter trajectories. All trajectory data is normalized to range of [0,10]. The demonstrations are normalized to [0,10] and provided incrementally and no batch data is stored. The intermediate results during training of the ProMP library are shown in the upper row, accumulated demonstrations are shown in the lower row. In the upper row, the shaded area represents two times the standard deviation, the solid lines show the mean, and the demonstrated trajectories are shown as gray lines. Here, our approach successfully updates existing components with new demonstrations and adds new components when required.

To demonstrate that the library learned with our new approach using the incremental processing of demonstrations converges to the solution of EM with batch learning, we compare the resulting skill libraries first qualitatively as shown in Figure 2.5 and quantitatively using the Kullback-Leibler Divergence to a baseline as shown in Figure 2.4 (b). Qualitatively speaking, our approach (Figure 2.5, upper row) represents all different letters as individual clusters and the trajectory means of the mixture model components match the means learned with EM in batch mode (Figure 2.5, bottom row).

While for fewer samples per letter the trajectory covariances learned with our approach are dominated by Σ_{init}, with increasing number of samples per letter it approximates the covariances of the EM solution as the influence of the initial covariance decreases. The same behavior can also be observed in the quantitative comparison. Hereby we compute the Kullback-Leibler (KL) divergence of our approach and EM to a baseline, computed with Maximum Likelihood estimation from labeled data. The KL-divergence of our approach is averaged over 100 trials, where the order of demonstrations is randomly permuted. In the batch EM case, we provided the method with the correct number of components, while in our



(a) (b)

Figure 2.4.: (a) The demonstrations in the first experiment are given in the form of hand-drawn letter trajectories, normalized to [0,10]. (b) We compare our approach against an EM approach, where for both we compute the KL-divergence to a baseline solution from labeled data. For increasing number of samples per letter our approach converges against the EM solution, while additionally being able to continuously integrate new data.

approach the algorithm had to find the correct number of components by itself. Figure 2.4 (b) shows that the KL-divergence between the solution of our approach and the baseline is large for fewer samples and decreases with increasing number of samples. Again the inital KL-divergence is comparably large as the initial covariance Σ_{init} dominates for few samples per cluster and decreases as the impact of Σ_{init} decreases.

The high variance in the KL-divergence for few samples is expected as the KL-divergence is sensitive to the entropy of the ground truth model, which depends on the selected demonstrations. The variance also shrinks as the entropy converges for multiple samples. The experiments show that our approach achieves results comparable to those of an EM approach. However, the advantage of our approach in contrast to the EM approach is the ability to work on incrementally incoming data while still achieving similar end results. That means our approach does not require all data in batch mode but incrementally learns and updates its models from new demonstrations, which is in particular beneficial in scenarios where not all data is known in the beginning but is only provided over time with new demonstrations.

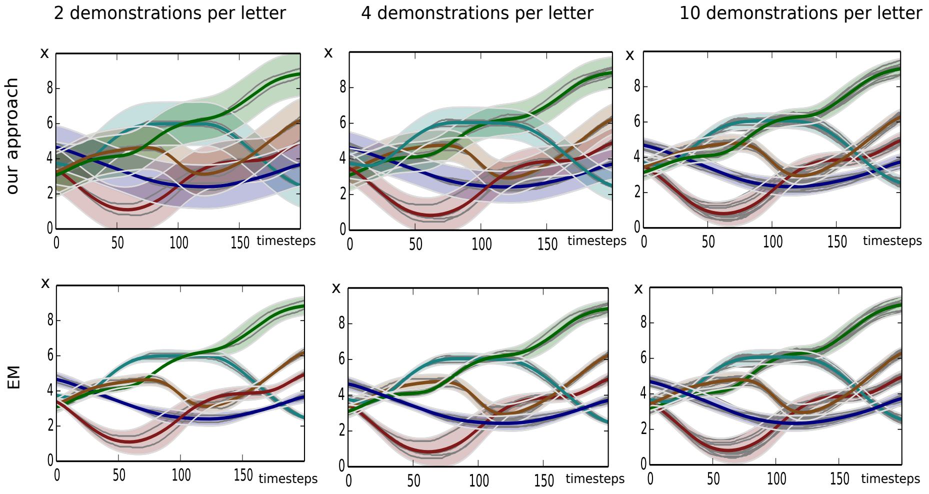


Figure 2.5.: For fewer samples per letter the covariance of the components in our approach is governed by the initial covariance. For increasing number of samples per letter the covariances converge to the underlying data covariances and result in comparable results to the EM approach.

2.3.2. Learning a Cooperative Skills with a Robotic Arm

The proposed approach is tested in a collaborative scenario, where a robot is supposed to assist a person in making a salad. The robot assists the person by first observing and recognizing the human action and second determining, adapting and executing an assistive response based on prior demonstrations. For the salad scenario, shown in Figure 2.6 (a), five different cooperative skills are required, namely:

- Board: The robot hands over the cutting board after the human grasped the knife.
- Tomato: The robot passes the tomato when the human reaches for the tomato.
- Bowl: The robot passes the salad bowl when the human reaches for the bowl.

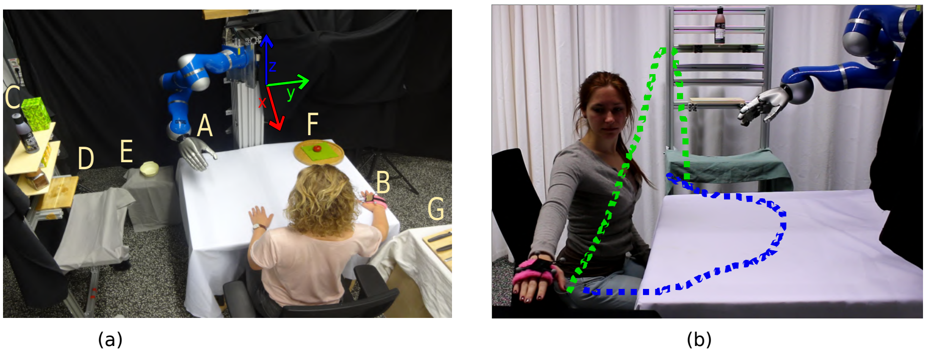


Figure 2.6.: (a) We evaluate our approach in a collaborative task where a robot (A) assists a human (B) in making a salad. The robot can hand over the board (D) the dressing (C), a tomato (F), the bowl (E) or assist with a standup motion. (b) We incorporate an automatic calibration procedure to determine the individual workspace boundaries of different subjects when subjects perform a circle in x-y plane (blue) and in y-z plane (green)

- Dressing: The robot gets the salad dressing from the shelf for the human.
- Standup: The robot supports the standup motion of the human.

Each of the cooperative skills is demonstrated separately and multiple times. To record the demonstrations for the collaborative skill we use sensory informations of a motion capturing system and the joint encoders of the robot. Thereby, the robot trajectory is shown by kinesthetic teaching, while the human action is recorded using motion capturing markers attached to the wrist (pink bracelet Figure 2.6 (b)). The used motion capturing system provides position and orientation of the human wrist (streamed to a ROS node) with an accuracy up to 1 cm at a framerate of 150 Hz. For the experiments in this thesis only the position was used. Additionally, for each skill a fixed grasp pose for the corresponding object is added to the robots skill database. In general, it would be beneficial to include also haptic feedback for grasping and camera based object tracking, however since this is not the main focus of this thesis only fixed object positions and grasp positions where used in an open loop fashion. The teaching is shown in Figure 2.8.

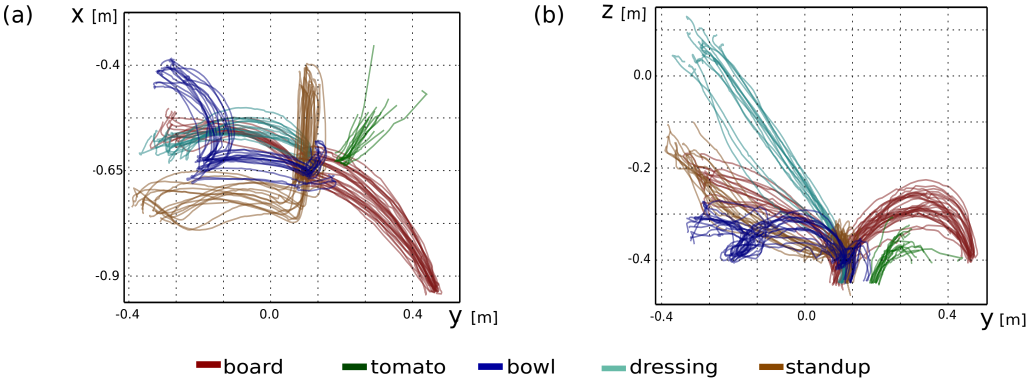
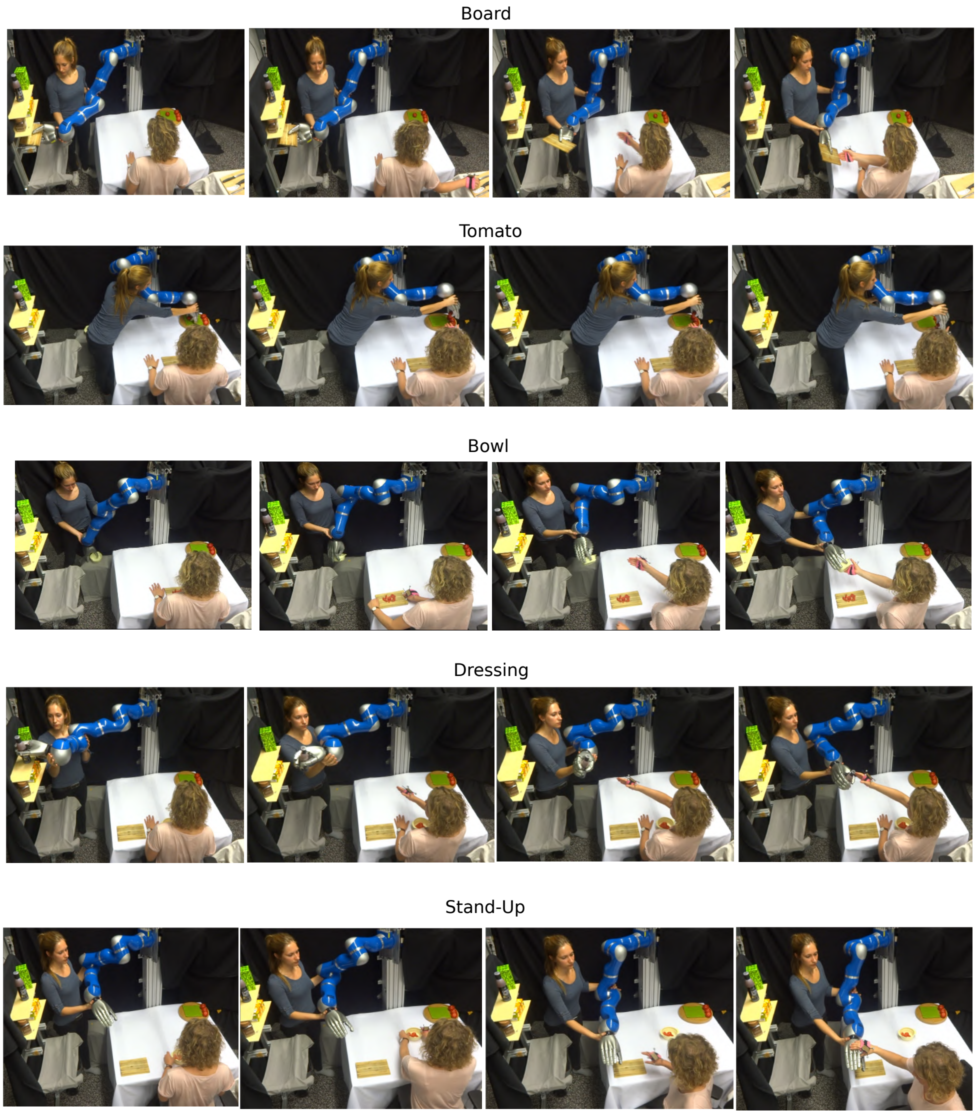


Figure 2.7.: For a human test subject (subj. 1) we recorded 15 demonstrations per skill. The trajectories are shown from in (a) from top-down and in (b) from front view.

Learning Personalized Skill Libraries

In an initial experiment, 15 demonstrations are recorded for every skill with a test subject (subject 1). The resulting human trajectories are shown from top-down and front view in Figure 2.7 (a) and (b). The recorded data is normalized to the range of [0,10] and we apply equidistant spatial interpolation to decouple the trajectory representation from the varying speed of the demonstrations. From the demonstrations our approach learns a cooperative skill library consisting of a gating model and multiple corresponding Interaction ProMPs. Hereby, the demonstrations are not provided as batch data but incrementally. An example of the gating model (which corresponds to the human part of the Interaction ProMPs) is shown in Figure 2.9. Five different skill clusters are clearly visible. Figure 2.11 (a) shows that similarly to the letter experiment, the averaged KL-divergence w.r.t. to the ground truth solution learned from labeled data decreases with the number of demonstrations per skill and is for more demonstrations comparable to the EM solution, while it enables incremental updates on new incoming data and does not require to store or recompute all batch data. In the interactive setting, the robot adapts its response to the human movement based on prior demonstrations. The output of the skill library is hereby a desired joint trajectory of the robot that gets executed by the robot's trajectory controller. Such adapted robot responses are shown for all skills in Figure 2.10.


Board

Tomato

Bowl

Dressing

Stand-Up

Figure 2.8.: Demonstrations are recorded as human and robot trajectories. For the robot we use kinesthetic teaching and the human wrist trajectory is tracked with a motion capturing marker. Each skill is demonstrated multiple times. We show here one example demonstration for each of the skills, namely board, tomato, bowl, dressing and stand-up.

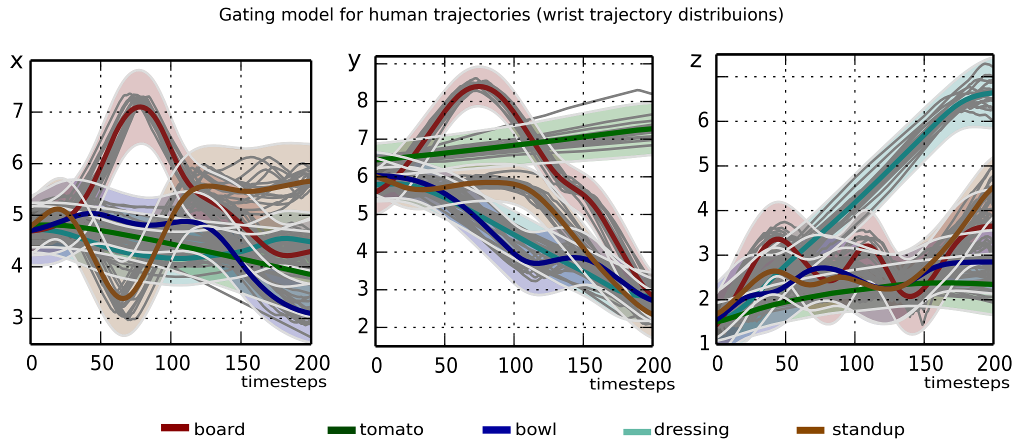


Figure 2.9.: On the demonstrations, we train a collaborative skill library, where we do not provide all demonstrations as batch data but incrementally add more demonstrations. The figure shows the resulting gating model, which corresponds to the human trajectory part of the Interaction ProMPs. The shaded area represents two times the standard deviation while the solid lines show the mean. The demonstrated trajectories are depicted as gray lines.

The adaptation of the robot response is achieved by conditioning the Interaction ProMP on the human trajectory as described in Section 2.2.3. An example of such an adaptation can be seen in Figure 2.11 (b), for the tomato skill. To further evaluate the applicability and robustness of the proposed approach, we conducted more experiments with different subjects and identical hyperparameters. For each subject individual demonstrations are recorded and a corresponding personalized skill library is incrementally learned. To evaluate the performance, the classification accuracy for recognizing the correct cooperative skill is evaluated by k-fold cross-validation. For subject 1 we use a training set of 10 demonstrations per skill and test on 5 demonstrations per skill, while all other subjects use 4 demonstrations per skill for training and 1 demonstration per skill for testing. The classification results averaged over 100 test and train sets are shown in Table 2.1. The first value corresponds to the percentage of successful classifications, the second to the percentage of wrong classifications (e.g tomato as bowl) and the third to the percentage of classifications as unknown. The results reveal that our approach works well for six of the subjects but classification accuracies for the other 4 subjects vary between skills. For the board skill subject 3, 7 and 9 have some movements

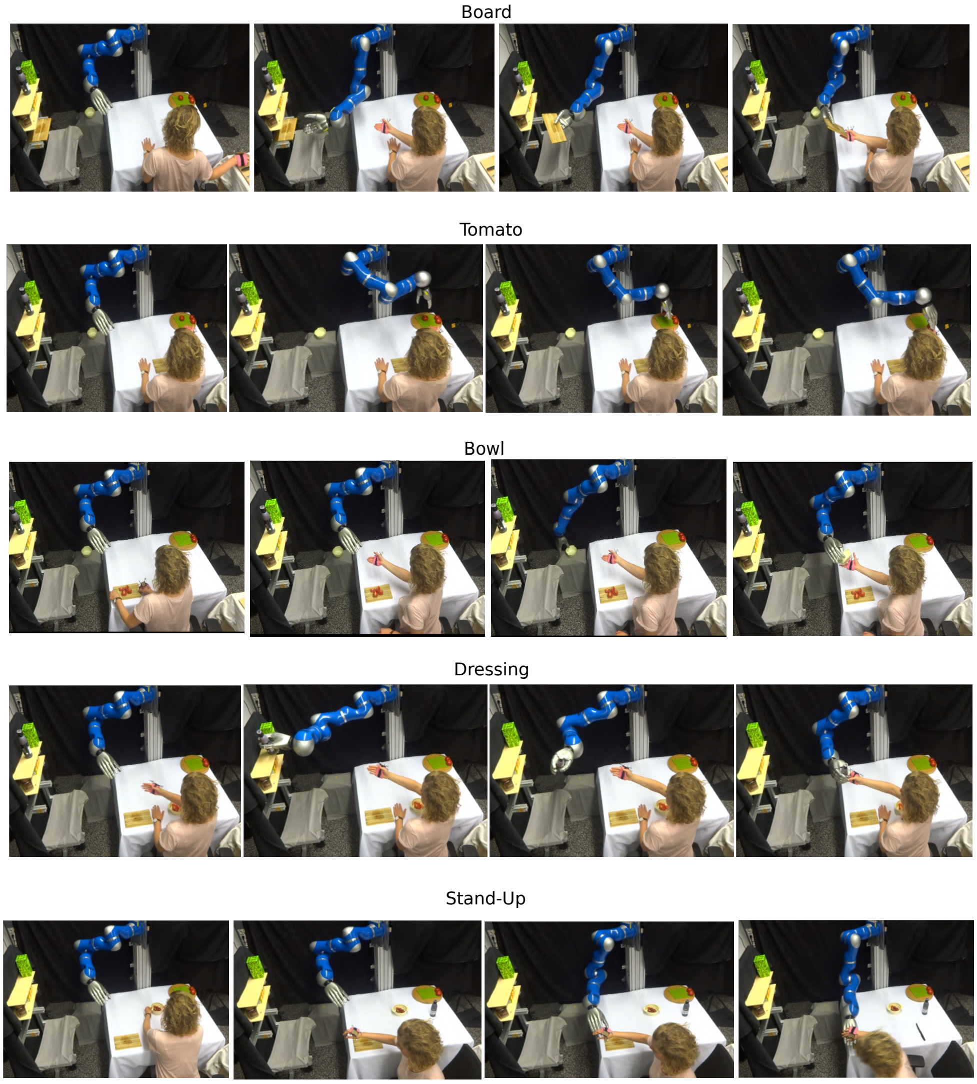


Figure 2.10.: For each of the skills the robot performs motions according to recognized previous demonstrations.

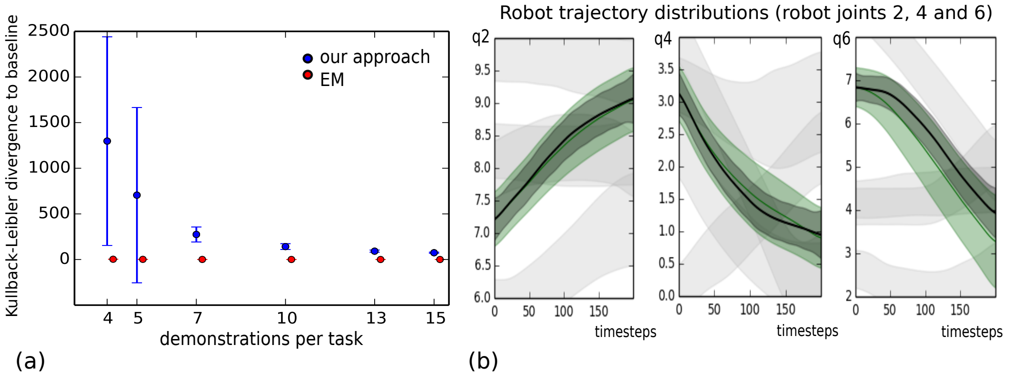


Figure 2.11.: (a) We again compare the KL-divergence of both our approach and an EM approach to a baseline from labeled data. For an increasing number of demonstrations per skill our approach converges to the EM solution but requires less recomputation and memory as new demonstrations arrive.(b) Once trained with demonstrations our model can subsequently be used to produce a corresponding robot response to an observed human trajectory. Therefore, first the gating model decides which of the Interac- tion Primitives (light gray) to activate (green). The activated primitive is then adapted to the observed human trajectory via conditioning (dark gray). The plots show joints q2, q4 and q6 of the robot arm, the shaded area represents two times the standard deviation, the solid lines show the mean.

with a high variance to the training set that are classified as unknown. However, the classification as unknown does not yield a wrong robot response and the robot would only ask for a new demonstration. Only for subject 10 the robot misclassifies the standup skill. Table 2.2 shows the number of learned components for the individual subjects. Depending on a variance of a subject's movement a single skill can be represented by multiple clusters since we used the same hyperparameters for all subjects and did not tune them individually. Additional clusters do not cause wrong classifications but can lead to unknown classification (subj. 3, 7 and 9). Results for subject 10 show that the wrong classifications for standup were due to too few learned clusters.

Data Preprocessing for Skill Transfer Between Subjects

We apply two steps of data preprocessing to ensure better transfer of learned skills between different subjects. First, we normalize the demonstrated trajectories with respect to the personal workspace boundaries of the subjects. These boundaries are extracted out of calibration data, where we let the subjects perform half circular movements in the x-y and z-y plane as depicted in Figure 2.6 (b) and record their wrist positions. From the recorded data we extract maximum and minimum values for each task space dimension $\{\boldsymbol{X}_{\text{min}}, \boldsymbol{X}_{\text{max}}\}$. For the z dimension, we noticed using only 90 percent of the extracted boundaries works well in our setup since subjects tend to stretch out more during calibration than when executing actual gestures. Given the personal boundaries, we normalize demonstrated trajectories to the range $[0, 10]$ and subtract the normalized start position of the trajectory

$$\boldsymbol{\tau}^{n}_{\text{norm}} = \frac{10(\boldsymbol{\tau}^{n} - \boldsymbol{X}_{\text{min}})}{\boldsymbol{X}_{\text{max}} - \boldsymbol{X}_{\text{min}}} - \boldsymbol{\tau}^{0}_{\text{norm}}, \tag{2.15}$$

where $\boldsymbol{\tau}^{n}_{\text{norm}}$ is the normalized n-th point of the trajectory and $\boldsymbol{\tau}^{0}_{\text{norm}}$ is the first normalized point of the trajectory. Here, we assume that the overall position of the human is static (e.g. seated on a chair), but exact start positions of the hand during skill execution might differ in between trial or subject.

To evaluate the data preprocessing we use four trajectories per skill per subject and incrementally compute a skill library per skill. We randomize the ordering of trajectories and average the number of resulting components per skill over 100 random seeds. The results are shown in Table 2.3, and Figure 2.12 and Figure 2.13 illustrate exemplary results of the resulting skill library components. In particular, we compare results without transformation relative to the start point (Figure 2.12) and results with transformation (Figure 2.13) . The experiments show that in general the transformation results in better generalization in between subjects and therefore less components per skill. While classification still works for too many components this can result in problems when trying to apply a library to unseen subject data. It also shows that while the skills dressing, tomato and bowl generalize well and mainly result in only one or two components over all subjects, the skill board and in particular standup result in multiple different model components and do not generalize so well for different subjects.

Evaluation of the Skill Libraries for Unseen Subjects

In this section, we evaluate how a library trained on data across 9 subjects performs in classification of trajectories from a new unseen subject. We train hereby

Table 2.1.: Classification Accuracy

	board	tomato	dressing	standup	bowl
subject1	1.0	1.0	1.0	1.0	0.99 (0|0.01)
subject2	1.0	1.0	1.0	1.0	1.0
subject3	0.85 (0|0.15)	1.0	1.0	0.72 (0|0.28))	1.0
subject4	1.0	1.0	1.0	1.0	1.0
subject5	1.0	1.0	0.99 (0.01|0)	1.0	1.0
subject6	1.0	1.0	1.0	1.0	1.0
subject7	0.77 (0|0.23)	1.0	0.88 (0|0.12)	1.0	1.0
subject8	1.0	1.0	1.0	1.0	1.0
subject9	0.84 (0|0.16)	1.0	1.0	1.0	1.0
subject10	1.0	1.0	1.0	0.77 (0.23|0)	1.0

Table 2.2.: Clusters per Subject

	4	5	6	7
subject 1	0	1.0	0	0
subject 2	0	1.0	0	0
subject 3	0	0.23	0.77	0
subject 4	0	1.0	0	0
subject 5	0.01	0.72	0.23	0.04
subject 6	0	1.0	0	0
subject 7	0	0.85	0.15	0
subject 8	0	1.0	0	0
subject 9	0	0.78	0.22	0
subject 10	0.23	0.77	0	0

Table 2.3.: Number of Clusters per Skill across subjects, with and (without) transformation

	1	2	3	4	5	≥ 6
tomato	1.0 (0.84)	0. (0.11)	0. (0.04)	0. (0.01)	0.(0.)	0.(0.)
board	0. (0.)	0.02(0.)	0.27 (0.0)	0.42 (0.0)	0.27 (0.12)	0.02 (0.88)
dressing	0.43(0.)	0.39 (0.)	0.12 (0.03)	0.03 (0.11)	0.03 (0.3)	0. (0.56)
bowl	0.06 (0.)	0.75 (0.0)	0.18 (0.)	0.01 (0.07)	0. (0.3)	0. (0.63)
standup	0. (0.)	0. (0.)	0. (0.)	0. (0.)	0.06 (0.)	0.94 (1.)

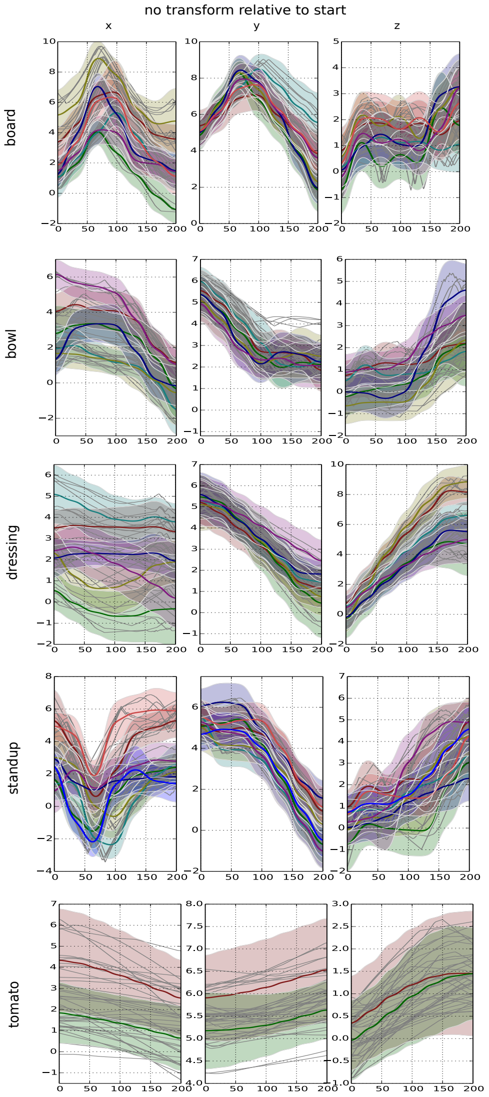


Figure 2.12.: We evaluate how the human motions (normalized data) for the different skills vary in between subjects and how many components a skill library trained in incremental fashion learns for each skill across subjects. Without a transform relative to the start positions there are usually more then one component learned per skill.

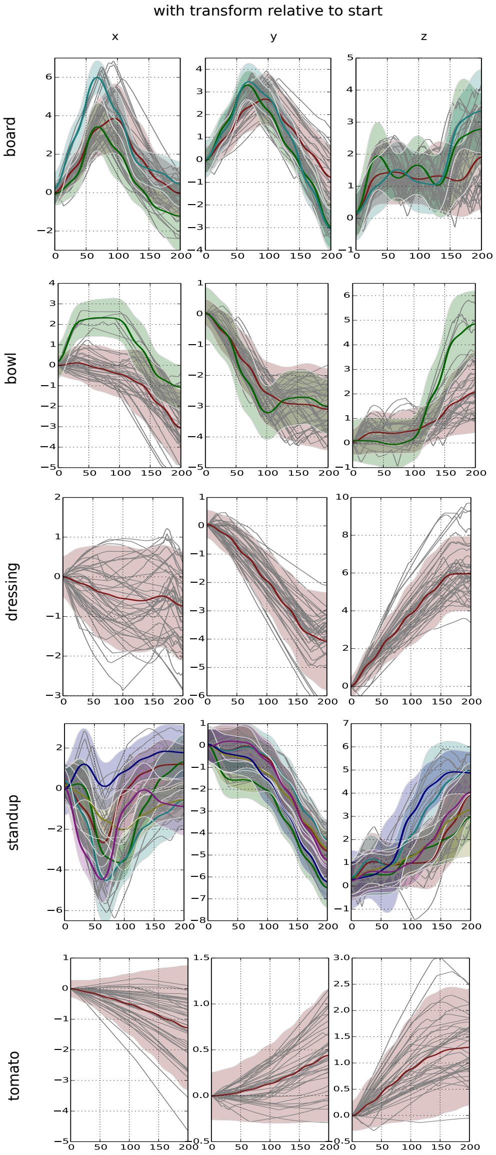


Figure 2.13.: The transformation of human trajectories relative to the start position results in higher similarity of trajectories across subjects and therefore in less components per skill.

on 5 trajectories per skill per subject and test for 5 trajectories per skill on the unseen subject. Hereby, we randomize the order of the training trajectories and average over 100 trials per unseen subject. Again we compare results with and without transformation of the trajectory data. Figure 2.14 shows the resulting accuracies for each subject. It shows that the transformation in general leads to better classification accuracy. While the transfer of the learned library to unseen subjects works well for tomato, dressing and bowl skill for most of the subjects, the board and standup skill do not transfer so well. Eventhough there are only little wrong classifications for standup and board there is a high percentage of unknown classifications which shows the trajectories of the new subject are too different from the learned library. Our approach could handle this by creating new model components for such cases. The results indicate that shorter motions, such as tomato, bowl or dressing, generalize better in between subjects. For more complex motions it might be beneficial to consider a different trajectory representation or additional modalities in the future.

2.3.3. Limitations

So far we use a gating model based on wrist motions of the human and their geometric representation. For better distinguishability of more complex gestures it would be important to also consider full arm motions and dynamics, for which the gating could be exchanged by another representation. We think this missing complexity of the gating so far limits scalability of our method to a large number of different skills and could be tackled in future work. Additionally, our experimental setup currently relies on the high accuracy of a motion capturing sytem. Another line of work is therefore to include camera based tracking of the human and objects, and compare the results of our method regarding e.g. reduced tracking accuraccy. Including also haptic sensory feedback and more advanced representations for grasping would be also beneficial for the skill library from a practical point of view.

2.4. Conclusion

In this chapter, we introduce a novel approach to learn a mixture model of probabilistic interaction primitives in an online and open-ended fashion. In contrast to existing batch approaches our approach is able to update existing interaction primitives continuously from new data and extend a cooperative skill library with new interaction patterns when needed. Experimental evaluation on a collaborative

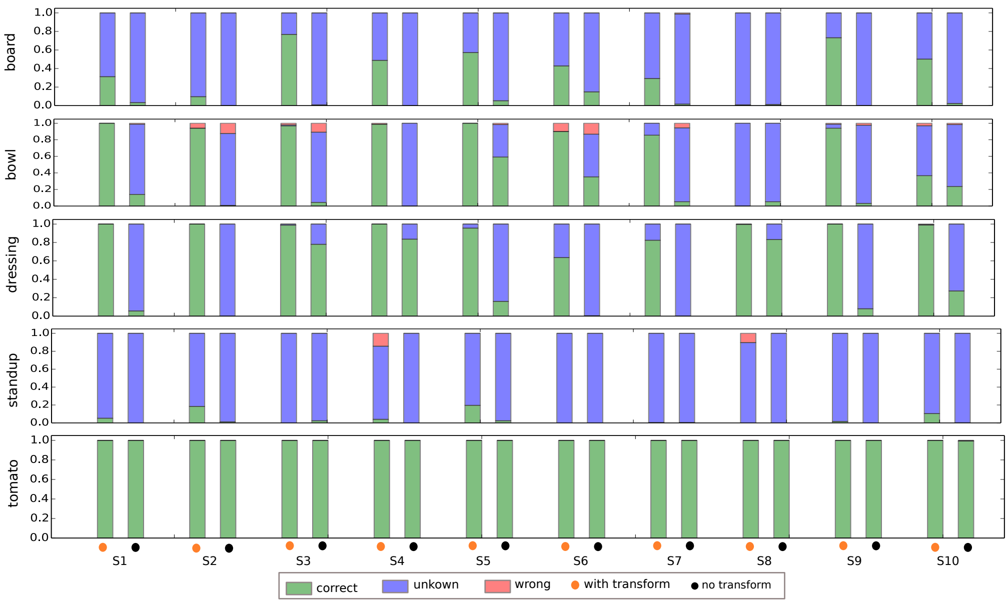


Figure 2.14.: We test a skill library learned on nine of ten subjects on the unseen tenth subject. It shows that the transformation of the data yields better results in terms of percentage of correct classifications (green), unknown classifications (blue) and wrong classifications (red) averaged over 100 combinations of test and training sets. It also shows that not all skills generalize equally well.

scenario with a 7DoF robot arm showed that our approach is able to learn multiple different collaborative skills from unlabeled training data and generate corresponding robot motions, based on prior demonstrations. Additionally, evaluations with 10 human subjects showed that our approach successfully learned a personalized collaborative library for the majority of subjects. Moreover, we evaluated how a library trained across multiple subjects generalizes to unseen new subjects.

However, since the experiments on different subjects indicate that motion data do not work equally well for all subjects and skills, we are currently investigating how to include other modalities such as gaze direction or voice commands in our gating model. Another line of research is to online adapt the hyperparameters, which are currently only hand-tuned, automatically to individual subjects and investigate more principled ways of hyperparameter selection, which can potentially improve classification results for individual subjects. Moreover, since for now the Interaction

ProMPs in the cooperative skill library are solely learned from demonstrations an important component for future work is to enrich and improve the trajectories of the robot, for example, by using reinforcement learning and include more sensory channels such as camera based perception instead of the motion capturing system, and additional haptic sensing for more advanced interaction skills.

3. Learning Intention Aware Online Adaptation of Movement Primitives

In contrast to classical robotic domains, where robots usually operate at a safe distance from humans, future robot applications such as elderly assistance or interactive manufacturing aim to bring robots closer to everyday contact with humans (Alenljung et al., 2017). In this context learning from demonstration (Osa et al., 2018) and the concept of movement primitives (Ijspeert et al., 2013; Calinon et al., 2007; Paraschos et al., 2018) offer a promising approach for non-expert users to teach new tasks to robots. In particular, Probabilistic Movement Primitives (ProMPs) (Paraschos et al., 2018) can capture the inherent variability in the demonstrated motions. However, when a robot is supposed to share a workspace in close proximity with a human, special requirements for online adaptation of learned robot motions arise. While ProMPs have been already extended for collaborative tasks (Maeda et al., 2017c) and offline planning methods with static obstacles (Koert et al., 2016; Colomé and Torras, 2017) exist, to the best of the authors' knowledge no method for online human aware adaptation of ProMPs in shared workspaces has been introduced so far. Such an approach needs to be able to react online to dynamic changes in human intentions and motion goals.

To avoid extensive replanning and the resulting inconsistency of robot motions, it is desirable to predict behavior changes of humans in advance. As human behavior might differ between situations and subjects it is desirable to learn both movement goals and motion behavior of humans from observations and in an online manner. Such motion models can predict potential collisions between the human and the robot in advance and adapt the robot's movements accordingly.

The contribution of this work are two novel approaches for intention aware online adaptation of ProMPs and their evaluation with non-expert users. The two approaches are inspired by time-dependent human collision avoidance behaviors, namely change in path direction and change in path velocity as also observed e.g., in pedestrian motions (Huber et al., 2014; Karamouzas and Overmars, 2010). Our first approach optimizes the shape of the ProMP for spatial obstacle avoidance,

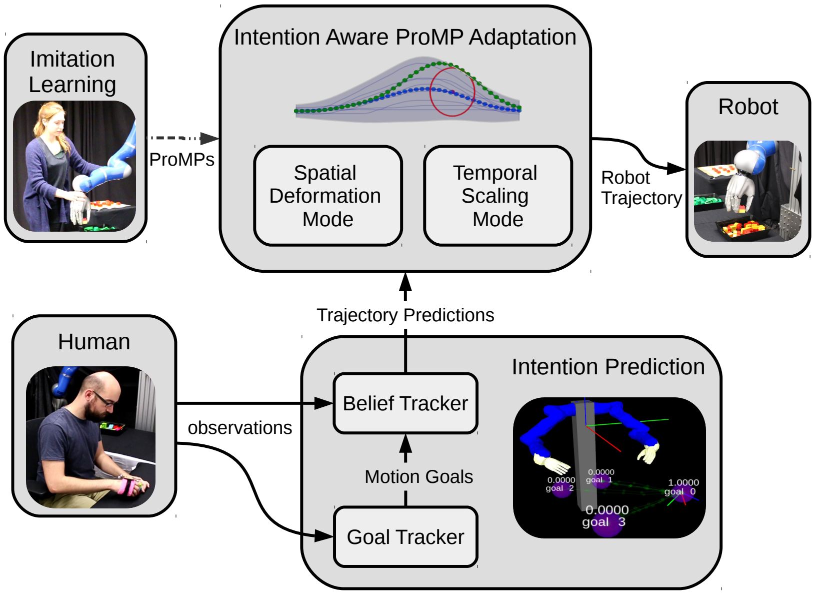


Figure 3.1.: We propose two novel approaches for online human aware adaptation of ProMPs, namely spatial deformation and temporal scaling. Therefore we learn a goal-directed probabilistic prediction model from observing human motions and use predictions from this model to online adapt ProMPs, which are originally learned via imitation learning.

taking into account information from the demonstrations, and follows a similar approach as Colomé and Torras (2017). However, while Colomé and Torras (2017) only proposed offline optimization our approach runs online and is able to react to dynamically changing human motions. The second novel approach optimizes the velocity profile of the ProMP to achieve obstacle avoidance while the motion path remains unchanged. For predicting human motions, both of the new online ProMP adaptation techniques use a goal-directed probabilistic prediction model learned from observations.

Commonly in pure motion planning the reaction of a human to the robot's motions is not the focus of investigations as long as collisions with the human are avoided. However, in human aware motion adaptation human presence gives need to also investigate how different types of robot motions influence a human working in the same workspace (Lasota and Shah, 2015; Arai et al., 2010; Dragan et al., 2013). Therefore, we conducted a user study with non-experts to evaluate the effects of our spatial and temporal motion adaptation approach on human task performance and subjectively perceived levels of comfort, safety and predictability of the robot's motions.

The rest of the chapter is structured as follows: Section 3.1, presents related work. Section 3.2 summarizes the concept of ProMPs and introduces our novel approaches for online adaptation of ProMPs to dynamic obstacles and a probabilistic model for predicting goal directed human motions. Section 3.3 presents results from a user study where we evaluated both online adaptation approaches and the prediction model on a pick and place task. In Section 3.4, we draw conclusions from the experiments and discuss possible future work.

3.1. Related Work

Efficient and safe coexistence of robots and humans has been a longstanding robotics challenge (Vasic and Billard, 2013; Kulić and Croft, 2005). In particular, when a human is in close proximity to the robot the situation differs from classical motion planning due to the human being highly dynamic and possibly reacting subjectively to different ways the robot moves (Lasota and Shah, 2015; Koppenborg et al., 2017; Dragan et al., 2013). While earlier approaches to human robot collaboration often consider safety zones or velocity limits (Vasic and Billard, 2013; Morato et al., 2014) more recent research investigates ways to generate human aware robot motions in close proximity and shared workspaces (Mainprice et al., 2015; Hayne et al., 2016; Losey and O'Malley, 2018; Busch et al., 2017; Sisbot and Alami, 2012). Mainprice et al. proposed a Gaussian Mixture Model (GMM) for predicting human motions and used constrained stochastic trajectory optimization to spatially deform robot trajectories (Mainprice and Berenson, 2013; Mainprice et al., 2015). Additionally, approaches for online trajectory deformation based on physical input signals from a human (Losey and O'Malley, 2018), human comfort and ergonomic postures (Busch et al., 2017), or optimizing human robot handovers have been presented (Sisbot et al., 2010; Sisbot and Alami, 2012).
These approaches mainly focus on deformation of trajectories. Adapting the motion speed has been used in human-robot interaction to decrease potential impact force (Kulić and Croft, 2005), slow down the robot when a human enters a monitored area (Morato et al., 2014), for online obstacle avoidance of two robots in a cooperative setting (Lopez et al., 2017), and for time-dependent collision

avoidance in navigation tasks with mobile robots (Bai et al., 2015; Trautman and Krause, 2010; Kruse et al., 2012).

For more efficient human robot co-working or collaboration and to avoid the need for extensive replanning, early prediction of human intentions is crucial. In particular, predicting human motion goals and reaching motions has been exploited in the literature (Ikeda et al., 2013; Ravichandar and Dani, 2015; Bai et al., 2015). Recently, also Gaussian Mixture models (Luo et al., 2018) or Probabilistic Movement Primitives (Dermy et al., 2017) have been used for early intention prediction of human motions. In collaborative assembly, (Unhelkar et al., 2018) controls the velocity of a robotic system along a linear axis dependent on potential collisions with co-workers.

Recent studies with non-expert users report the benefits of such human aware planning approaches for mobile robots (Trautman and Krause, 2010; Kruse et al., 2012) and in shared workspaces with robot manipulators (Lasota and Shah, 2015). They also report the contrast between legibility and predictability in motions (Dragan et al., 2013) and effects of motion speed and predictability (Koppenborg et al., 2017). However, non-expert user studies on different online replanning behaviors are rare in the literature, but important to better understand human responses to robot motions.

Contrary to many of the previously mentioned methods, our approach incorporates movement primitives which we will discuss in the following. Movement primitives (Ijspeert et al., 2013; Calinon et al., 2007; Paraschos et al., 2018) provide a lower dimensional representation of trajectories and an intuitive way for non-experts to teach new tasks to a robot by demonstrations. In particular, Probabilistic Movement primitives offer a framework to capture the inherent variability of the motions (Paraschos et al., 2018).

For Dynamic Movement Primitives (DMPs) the use of repellent forces was proposed for obstacle avoidance (Park et al., 2008; Hoffmann et al., 2009), and for dynamic systems approaches the use of potential fields is presented in (Khansari-Zadeh and Billard, 2012). Saveriano et al. (2017) propose human-aware motion reshaping using dynamical systems, where the robot adapts velocity and its motion goal online dependent on human motion. In particular, they also introduce a RGBD camera based approach for fast and efficient distance computation to the human. However, they do not incorporate prediction models for human motions and do not conduct user studies on perceived subjective safety or comfort.

ProMPs have been extended to collaborative tasks (Maeda et al., 2017c) and offline trajectory planning (Koert et al., 2016). However, the computationally expensive sample based Kullback-Leibler Divergence in Koert et al. (2016) prevents online replanning. Colomé and Torras (2017) proposed a demonstration free version of ProMPs and use the Mahalanobis distance to the demonstrations for static obstacle avoidance without online replanning. However, in a setting with a human present, dynamic obstacle avoidance, which is capable of human aware online replanning, is crucial.

3.2. Intention Aware Online ProMP Adaptation

We propose two novel approaches for online intention aware adaptation of ProMPs, namely spatial deformation and temporal scaling. In both cases, we first learn a goal-directed motion model from observations of human task execution and use this probabilistic prediction model to online adapt ProMP trajectories afterwards. For spatial deformation, we online deform the current path of the robot to avoid dynamic obstacles while staying close to demonstrated distributions. For temporal scaling, we solely adapt the velocity profile of the ProMP while staying on the original path. Figure 3.1 illustrates the different components of our approach.
We first recap on ProMPs and then present our two approaches for online adaptation of ProMPs to dynamic obstacles, namely spatial online deformation and temporal scaling of the ProMP. Moreover, we introduce our probabilistic model for goal-based prediction of human motions.

3.2.1. Online Spatial Deformation of ProMPs

We propose online spatial adaptation of ProMPs by optimizing the current weight vector of the ProMP which results in spatial deformation of the resulting trajectory. To this end, we propose a constrained optimization problem to obtain an updated weight vector $\boldsymbol{w}$ that minimizes the Mahalanobis distance to the original ProMP weight distribution $p(\boldsymbol{w})$ and constrain the minimal distance to obstacles as well as sudden changes in the resulting trajectory

$$\begin{aligned} \underset{\boldsymbol{w}}{\arg\min} \quad & (\boldsymbol{w} - \boldsymbol{\mu}_w)^{\mathrm{T}} \mathrm{diag}(\boldsymbol{\Sigma}_{\boldsymbol{w}}^{-1})(\boldsymbol{w} - \boldsymbol{\mu}_w) \\ \text{s.t.} \quad & \varepsilon_{\mathrm{o}} > \Delta(\boldsymbol{\phi}_{t-1}, \boldsymbol{\phi}_t, \boldsymbol{w}, \boldsymbol{O}_t) \quad \forall t, \\ & \varepsilon_{\mathrm{w}} > (\boldsymbol{\phi}_t \boldsymbol{w} - \boldsymbol{\phi}_t \boldsymbol{w}_{\mathrm{curr}})^{\mathrm{T}} (\boldsymbol{\phi}_t \boldsymbol{w} - \boldsymbol{\phi}_t \boldsymbol{w}_{\mathrm{curr}}), \end{aligned} \tag{3.1}$$

where ε_{o} denotes the bound for the minimal distance to an obstacle, $\boldsymbol{O}_t$ denotes a vector of obstacles at time t, Δ denotes the minimum distance of the discretized

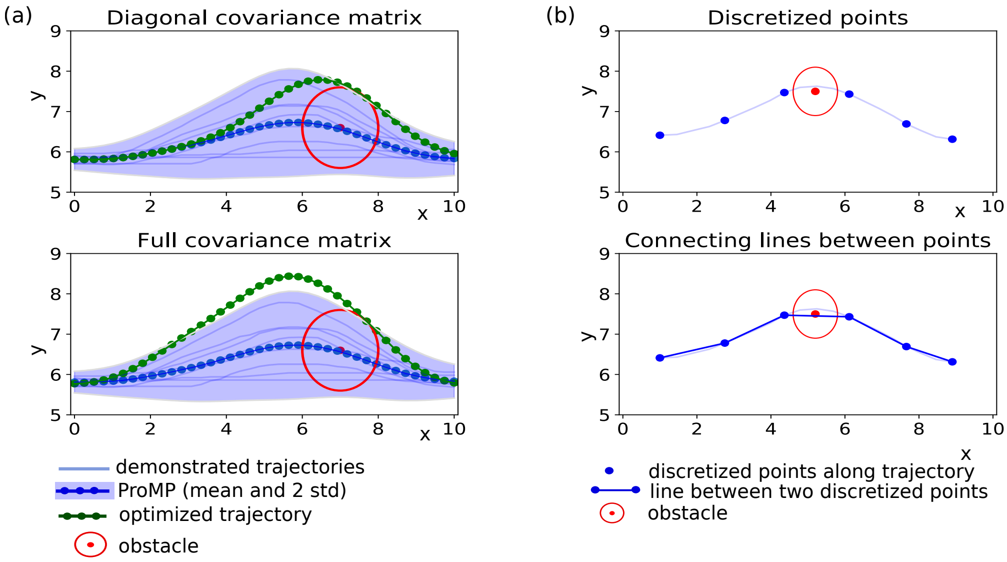


Figure 3.2.: (a) When only the diagonal of the covariance is used for optimization the trajectory stays closer to the ProMP's mean in regions unaffected by the obstacle. When using the full covariance matrix, the optimized trajectory will stay closer to the correlation of the demonstrations. (b) It is advantageous to not only compute obstacle distances for discretized points but of connections between points to enable a more sparse discretization.

robot trajectory to the obstacle vector, which we discuss in more detail later, and ε_{w} limits the change of the weight vector in the current position of the trajectory.

For the Mahalanobis distance, we use the diagonal of the covariance matrix of the weight distribution, as a high correlation between the weights prevents the trajectory from only deforming in regions affected by obstacles. When using the full covariance the optimized trajectory stays closer to the correlations in the demonstrations which may be also desirable in certain applications. However, in our application the optimized trajectory should stay close to the mean in areas unaffected by obstacles. This is also illustrated in Figure 3.2 (a).

Using the Mahalanobis distance to stay close to the demonstrated distribution has also been proposed in Colomé and Torras (2017) for offline trajectory optimization. However, Colomé and Torras (2017) use it as a constraint while we use it as the objective of the optimization problem. As we run the optimization online, we additionally constrain the possible changes in Cartesian or joint space resulting from a change in the weight vector to avoid jumps in the trajectory. To compute the closest distance of the resulting trajectory to obstacles, we discretize over time. However, a sparse discretization over time can be problematic as illustrated in Figure 3.2. Therefore, we compute the minimum distance of the connecting line between two subsequent time steps to the obstacles, as also proposed in Pavlichenko and Behnke (2017) for stochastic trajectory optimization. We compute the minimum distance $d_{\text{min}}(\boldsymbol{\phi}_{t-1}, \boldsymbol{\phi}_t, \boldsymbol{w}, \boldsymbol{o})$ to an obstacle $\boldsymbol{o}$ as the minimum distance between the line connecting the two subsequent trajectory points $t-1$ and t and the obstacle

$$\begin{aligned} d_{\text{min}}(\boldsymbol{\phi}_t, \boldsymbol{\phi}_{t-1}, \boldsymbol{w}, \boldsymbol{o}) &= \frac{|\boldsymbol{v}_1 \times \boldsymbol{v}_2|}{|\boldsymbol{v}_1|}, \\ \boldsymbol{v}_1 &= \boldsymbol{\phi}_t \boldsymbol{w} - \boldsymbol{\phi}_{t-1} \boldsymbol{w}, \\ \boldsymbol{v}_2 &= \boldsymbol{\phi}_{t-1} \boldsymbol{w} - \boldsymbol{o}, \end{aligned} \tag{3.2}$$

where $\times$ denotes the cross product and $|\cdot|$ denotes the Euclidean norm. We compute the minimum over all obstacles at time step t as

$$\Delta(\boldsymbol{\phi}_{t-1}, \boldsymbol{\phi}_t, \boldsymbol{w}, \boldsymbol{O}_t) = \min_{o \in O_t}(d_{\text{min}}(\boldsymbol{\phi}_{t-1}, \boldsymbol{\phi}_t, \boldsymbol{w}, \boldsymbol{o})). \tag{3.3}$$

This allows for a more sparse discretization which results in more efficient computation. Figure 3.2 (b) illustrates the advantages over computing point-wise distances. To increase efficiency we utilize knowledge from the demonstrations: we initialize the optimization of the spatially deformed trajectory with a demonstrated trajectory that has the maximal distance to the obstacle compared to other demonstrations.

3.2.2. Online Temporal Scaling of ProMPs

Instead of modifying the chosen path direction, a common technique for humans to avoid time-dependent collisions with dynamic obstacles is to adapt the velocity along the robot's path (Huber et al., 2014; Karamouzas and Overmars, 2010). To achieve adaptive online velocity scaling of ProMPs we propose the use of a generalized logistic function

$$\sigma(\bar{z}) = \delta z_0 + \frac{\delta z_N - \delta z_0}{1 + (1/\varepsilon_{\text{start}}) \exp(m(\bar{z}_{\text{c}} - \bar{z}))}, \tag{3.4}$$

to compute the phase velocity δz for a given phase z, where $\bar{z}$ is the phase scaled to $[0, 100]$ this is $\bar{z} = 100z$, δz_0 is the starting phase velocity, δz_N is the resulting end velocity, m controls the slope of the velocity change and $\bar{z}_\text{c}$ denotes the phase where the velocity change starts, which is the point when the resulting $\sigma(\bar{z}_\text{c})$ deviates by a predefined small value ε_start from δz_0. This function can encode smooth deceleration and acceleration profiles depending on the chosen parameter values as illustrated in Figure 3.3. The phase velocity δz is then computed as

$$\delta z = \delta z_\text{max}\sigma(\bar{z}), \tag{3.5}$$

where δz_max denotes the upper limit for the phase velocity. Once we detect potential collisions with obstacles along the path we compute the phase of collision $\bar{z}_\text{stop}$ with an obstacle from a discretized phase vector and subsequently adapt the parameters of the generalized logistic function for a deceleration dependent on the slowing down phase duration

$$\gamma = \bar{z}_\text{stop} - \bar{z}_\text{c}, \tag{3.6}$$

where $\bar{z}_\text{stop}$ is the point in time where the phase velocity decreases below a predefined small value ε_stop, resulting in the robot stopping. Here we used $\varepsilon_\text{stop} = \varepsilon_\text{start} = 0.1$. For a given $\bar{z}_\text{stop}$ and the current phase $\bar{z}_n$ we can compute γ by solving the constrained optimization problem

$$\begin{aligned} \arg\min_{\gamma} \quad & (\gamma - \gamma_{opt})^2, \\ s.t. \quad & \bar{z}_\text{stop} - \gamma > \bar{z}_n, \end{aligned} \tag{3.7}$$

where γ_{opt} denotes a desired optimal slowing down duration, that needs to be chosen a priori. Given the optimized γ^* we can compute from Equation (3.6) the phase where the velocity change starts $\bar{z}_c^* = \bar{z}_\text{stop} - \gamma^*$. Next, we compute the slope m^* by solving Equation (3.4) for m, plugging in the knowledge we already have about the slowing down parameters

$$\begin{aligned} \sigma(\bar{z}_\text{stop}) &= \varepsilon_\text{stop}, \\ \delta z_0 &= 1, \\ \delta z_N &= 0, \\ m^* &= -\frac{1}{\gamma^*} \log\left(\frac{\varepsilon_\text{stop}\varepsilon_\text{start}}{1-\varepsilon_\text{stop}}\right), \end{aligned} \tag{3.8}$$

and use m^* and $\bar{z}_c^*$ to update the velocity profile. This results in a smooth slowing down of the ProMP if an obstacle is predicted in advance to be on the path and in a hard stop if the obstacle crosses the path unexpectedly. Once the robot has no potential collisions along the path anymore, we adapt the sigmoid function to accelerate again to the original speed.

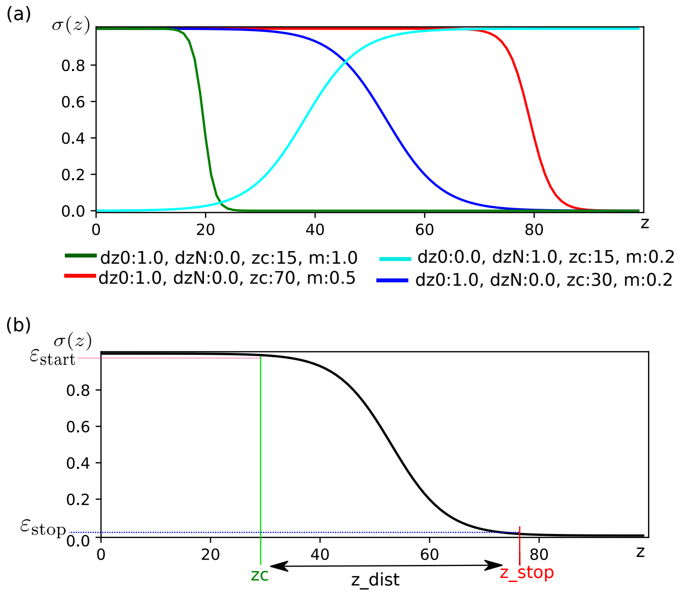


Figure 3.3.: (a) Depending on the choice of the parameters the generalized sigmoid function results in smooth slowing down (dark blue), rapid stopping (green) or accelerating behavior (light blue) along the path. (b) For a given desired stop phase z_{stop} and a given optimal deceleration duration we can compute the parameters of the resulting deceleration profile.

3.2.3. Probabilistic Model for Human Trajectory Prediction

To avoid the need for extensive online replanning and to ensure consistency in the robot motions we additionally propose a probabilistic model for human intention and trajectory prediction. This model can run online and learns human motion goals and transition probabilities between them from observations. The model consists of a goal tracker, which extracts human motion goals from observations using an incremental Gaussian Mixture Model and a belief tracker that computes probabilities of currently active goals and transitions between them from a history of previous observations.

Goal Tracker

We propose an online and open-ended approach to learn a distribution over possible Cartesian movement goal positions from human wrist trajectories. Hereby, a goal tracker extracts potential motion goals $\boldsymbol{g}$ from human motion data by learning a Gaussian Mixture Model (GMM) over zero velocity points $\boldsymbol{p}_i$. The distribution $p(\boldsymbol{g})$ over goals $\boldsymbol{g}$ is represented as

$$p(\boldsymbol{g}) = \sum_{k=1}^{K} \alpha_k \, \mathcal{N}\big(\boldsymbol{g} \,|\, \boldsymbol{\mu}_k^g, \boldsymbol{\Sigma}_k^g\big) \tag{3.9}$$

where α_k denotes the prior, K denotes the total number of goals, $\boldsymbol{\mu}_k^g$ is the mean and $\boldsymbol{\Sigma}_k^g$ is the covariance of the k-th component respectively. We assume uninformed priors in our model such that $\alpha_k = 1/K$.

In an online setting, the total number of motion goals is not known a priori. Therefore, we propose to learn the model incrementally using an incremental Gaussian Mixture model (Engel and Heinen, 2010). Hereby, we first extract potential goal points $\boldsymbol{p}_i$ from observations of the human wrist position. The potential goal points are zero velocity points that are points where the change in wrist position stays below a predefined threshold for a certain amount of time. Once a new potential goal point is detected the subsequent Expectation Maximization like algorithm computes in the Expectation step the responsibilities of each goal for a new potential goal point $\boldsymbol{p}_i$. This is the probability of the point belonging to an existing mixture component

$$\begin{aligned} p(\boldsymbol{g}_k|\boldsymbol{p}_i) &= \frac{p(\boldsymbol{p}_i|k)p(\boldsymbol{g}_k)}{p(\boldsymbol{p}_i)} \\ &= \frac{\alpha_k \, \mathcal{N}\big(\boldsymbol{p}_i \,|\, \boldsymbol{\mu}_k^g, \boldsymbol{\Sigma}_k^g\big)}{\sum_{j=1}^{K} \alpha_j \, \mathcal{N}\big(\boldsymbol{p}_i \,|\, \boldsymbol{\mu}_j^g, \boldsymbol{\Sigma}_j^g\big)}. \end{aligned} \tag{3.10}$$

These responsibilities are then used to update the existing models and their parameters in the Maximization step:

$$\begin{aligned}
s_k =& s_k + p(g_k|\boldsymbol{p}_i), \\
v_k =& \frac{p(g_k|\boldsymbol{p}_i)}{s_k}, \\
\boldsymbol{\mu}_k^g =& \boldsymbol{\mu}_k^g + v_k(\boldsymbol{p}_i - \boldsymbol{\mu}_k^g), \\
\boldsymbol{\Sigma}_k^g =& (1 - v_k)\boldsymbol{C}_k + v_k(\boldsymbol{p}_i - \boldsymbol{\mu}_k^g)(\boldsymbol{p}_i - \boldsymbol{\mu}_k)^{\mathsf{T}} - \\
& (\tilde{v}_k - v_k)(\boldsymbol{\mu}_k^g - \boldsymbol{\mu}_k^{\text{g,old}})(\boldsymbol{\mu}_k - \boldsymbol{\mu}_k^{\text{g,old}})^{\mathsf{T}},
\end{aligned}$$

where s_k can be interpreted as a measure for the amount of data the component already modeled well, and v_k is an update weight. If all likelihoods $p(\boldsymbol{p}_i|k)$ are below a threshold P_{new} a new component is initialized according to

$$\begin{aligned}
\boldsymbol{\mu}_{k+1} &= \boldsymbol{p}_i, \\
\boldsymbol{\Sigma}_{k+1} &= \boldsymbol{\Sigma}_{init}, \\
s_{k+1} &= 1.
\end{aligned} \tag{3.11}$$

We additionally check the goals for minimal support to regard outliers and delete mixture components if they did not reach a threshold s_k^{min} after a certain lifetime.

The GMM is subsequently used by the belief tracker to track activation of goals and predict human trajectories.

Belief Tracker

Similar to the approach of goal directed motion prediction in Bai et al. (2015) we introduce a probabilistic model to track the current goal of human motions. Based on a given sequence of observed human wrist positions $\boldsymbol{o}_t = (x_t^h, y_t^h, z_t^h)$ we can update our belief over goals of the human $b_t(\boldsymbol{g}_k)$, this is the belief towards which goal the human is currently reaching to. For this, we use Bayes Theorem to compute the updated belief over goals

$$\begin{aligned}
b_{t+1}(\boldsymbol{g}_k) &= p(\boldsymbol{g}|\boldsymbol{o}_t, \boldsymbol{b}_t) \\
&= \frac{p(\boldsymbol{o}_t|\boldsymbol{g}_k, b_t)p(\boldsymbol{g}_k|b_t)}{\sum_j p(\boldsymbol{o}_t|\boldsymbol{g}_j, b_t)p(\boldsymbol{g}_j|b_t)} \\
&= \frac{p(\boldsymbol{o}_t|\boldsymbol{g}_k, b_t)b_t(\boldsymbol{g}_k)}{\sum_j p(\boldsymbol{o}_t|\boldsymbol{g}_j, b_t)b_t(\boldsymbol{g}_j)},
\end{aligned} \tag{3.12}$$

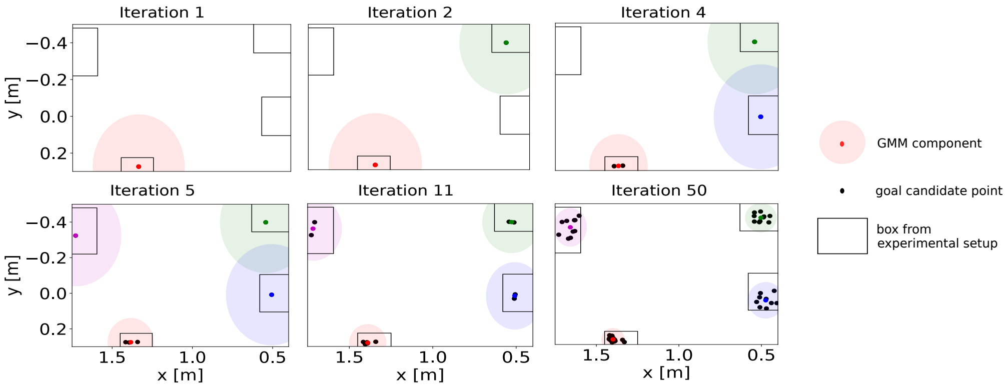


Figure 3.4.: We incrementally learn a GMM over motion goals using zero velocity points as goal candidates. The figure shows an example of this procedure in the experimental setup that we used. For each new goal candidate point our approach either creates a new mixture component or updates the existing components.

where we compute $p(\boldsymbol{o}_t|\boldsymbol{g}_k, b_t)$ by assuming noisy goal directed movements of the human as also proposed in Bai et al. (2015)

$$p(\boldsymbol{o}_t|\boldsymbol{g}_k, b_t) = \mathcal{N}(\boldsymbol{o}_t|\hat{\boldsymbol{o}}_k, \boldsymbol{I}\sigma_k), \text{ with} \\ \hat{\boldsymbol{o}}_k = \boldsymbol{o}_{t-1} + \frac{\boldsymbol{g}_k - \boldsymbol{o}_{t-1}}{|\boldsymbol{g}_k - \boldsymbol{o}_{t-1}|} v_h \Delta t, \tag{3.13}$$

where $\boldsymbol{I}$ is the identity matrix, σ_k denotes noise along the trajectory towards a goal, v_h denotes the current estimated human velocity and $|\cdot|$ denotes the Euclidean norm.

Additionally, we learn the transition probabilities of a goal given a sequence of prior goals $\mathcal{G}$ from the observations

$$p(\boldsymbol{g}_k|\mathcal{G}) = \frac{\#\boldsymbol{g}_k|\mathcal{G}}{\sum_j \#\boldsymbol{g}_j|\mathcal{G}}, \tag{3.14}$$

where $\#\boldsymbol{g}_k|\mathcal{G}$ denotes the number of occurrences of goal k given the sequence of goals $\mathcal{G}$. Since the robot's behavior may influence the transition probabilities of the human between goals, so that it can change compared to human task execution without a robot, we propose to update these probabilities online using exponential

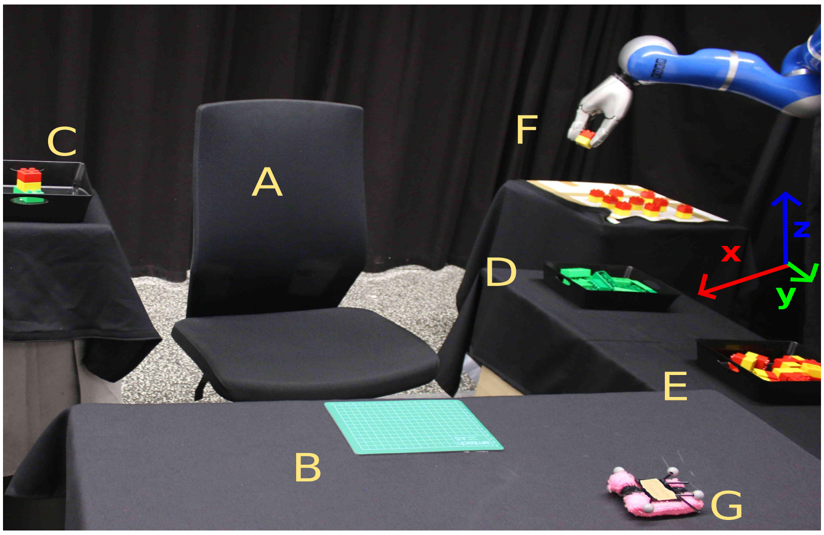


Figure 3.5.: We evaluate the proposed approaches for intention aware online adaptation of ProMPs on a pick and place task. The human (A) assembles parts from D and E in area B and delivers them to C, while the robot refills the parts in E from F.

decay for a new occurrence of goal k given the history of goals G

$$\begin{aligned} p'(\boldsymbol{g}_k|\mathcal{G}) &= p(\boldsymbol{g}_k|\mathcal{G})(1-\beta)+\beta, \\ p'(\boldsymbol{g}_j|\mathcal{G}) &= p(\boldsymbol{g}_j|\mathcal{G})(1-\beta) \quad \forall \quad j \neq k, \end{aligned} \tag{3.15}$$

where $\beta \leq 1$ denotes the decay factor.

We represent the predicted distribution over human trajectories with M particles, where each of these particles is initialized with a goal sampled according to $p(\boldsymbol{g})$. The particle is propagated S time steps into the future towards this goal according to the probabilistic motion model in Equation (3.12), and in case it reaches this goal a new goal is sampled according to the transition probabilities. The resulting vector of predictions can then be used as an obstacle vector for intention aware online adaptation of ProMPs as introduced in Section 3.2.1 and 3.2.2.

3.3. Experimental Evaluation

In this section, we introduce the experimental setup and present results from evaluations with non-expert users. We first describe the experimental setup in Section 3.3.1. In Section 3.3.2 we report results of the evaluation of the goal tracker and in Section 3.3.3 we report the results from the experimental evaluation of the proposed methods for online adaptation of ProMPs to human subjects.

3.3.1. Experimental Setup

We evaluate our proposed methods on a pick and place task in a shared workspace as shown in Figure 3.5 with 25 non-expert subjects. The task of the subjects (seated at A) is to assemble parts collected from D and E. The assembly area is at B and assembled parts should be delivered to C. The order in which the human takes D and E can be chosen freely and the subjects' motions are tracked via motion capture (G). The robot delivers parts from F to E, refilling the parts at E, for which two task space ProMPs (from F to E and from E to F) were learned from kinesthetic teaching as shown in the upper row of Figure 3.6. During spatial deformation we kept the orientation fixed. The demonstrations did also include trajectories that avoid potential positions of the human in the robot's workspace. After a task familiarization phase each subject performed the task under 4 different conditions (modes). In the first mode the human executed 10 repetitions of the assembly task without the robot moving. From this data, as shown in Figure 3.7 (a), the initial motion model including motion goals and transition probabilities between goals was learned. In the next three modes the human performed 15 repetitions of the assembly task while the robot moved in three different ways that we will refer to as N (no adaptation, Figure 3.6 second row), D (spatial deformation, Figure 3.6 third row) and T (temporal scaling, Figure 3.6 last row). We recorded human and robot trajectories and additionally the subjects answered questionnaires after each experiment and three questions comparing the different adaptation modes at the end of all experiments.

3.3.2. Learning Motion Goals

When the human performed the task without the robot we recorded the wrist position of the subjects to learn motion goals and transition probabilities and average stay durations at the goals. Figure 3.4 shows how a GMM is build incrementally for the motion goals. We compare the mean of the learned goals for all subjects

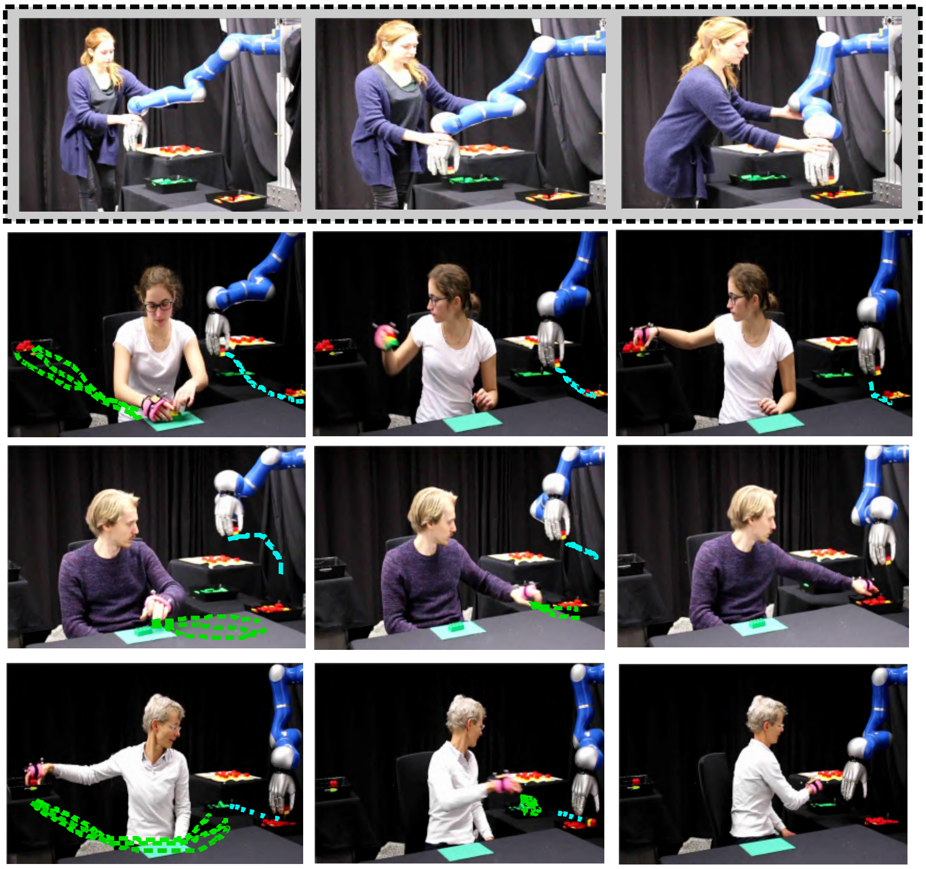


Figure 3.6.: We learn probabilistic movement primitives for a pick-and place task via kinesthetic teaching (upper row). When executed in a shared human robot workspace as shown in row 2-4 the learned motions need to be adapted online. To this end, we propose a novel approach for human aware execution of ProMPs by incorporating goal directed predictions of human motions (green) with two novel approaches for online adaptation of ProMPs, namely spatial deformation (third row) and temporal scaling (last row). We compare those two approaches to non reactive ProMP execution (second row).

with the measured positions of the goals. Table 3.1 shows that for all subjects the remaining error is below 5cm. This error can be caused due to individual placements of the motion capture markers on the subjects' hands and different

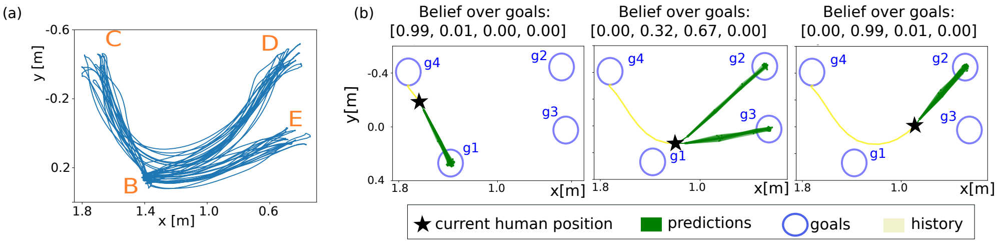


Figure 3.7.: (a) First we record motion data from the human wrist position while the robot is not moving. The recording is done using a motion capturing system and corresponding markers attached to the human wrist. (b) Using a probabilistic goal directed motion model,that was learned from observation, the belief over the goals and the predictions (green) change given an observed human trajectory (yellow) and the current position (black star).

distributions of the parts in the boxes. The learned motion goals were subsequently used for trajectory predictions as shown in Figure 3.7 (b).

3.3.3. Intention Aware Online Adaptation of ProMPs

We evaluated human response to three different robot adaptation behaviors, namely no adaptation of the ProMPs (Mode N), online spatial deformation (Mode D) and temporal scaling (Mode T), both according to predicted human trajectories. We randomized the order of the modes and the subjects were not told how or if the robot would respond to them.

During the experiments, we recorded trajectories of the human and the robot and additionally, the humans answered a questionnaire after each individual mode.

	Average error of learned goals [cm]
Goal 1	4.3 ± 1.6
Goal 2	4.2 ± 1.4
Goal 3	3.9 ± 1.3
Goal 4	3.8 ± 1.5

Table 3.1.: Motion Goals

The questionnaire consisted of five questions:
Q1: "The robot adapted its movements to me",
Q2: "I felt disturbed by the movements of the robot",
Q3: "The behavior of the robot was predictable for me",
Q4: "I felt uncomfortable due to the robot's movements",
Q5: "I trusted the robot not to hurt me".
For all questions, we evaluated approval on a 5 point Likert scale. The subjects also took notes on how they would describe the robot behavior after each mode. Additionally, the subjects were asked to answer three comparison questions, in the end, on which of the modes made them feel 1) most uncomfortable, 2) most safe and 3) least disturbed.

We evaluated idle times of the human and robot, average trajectory length and time per assembly of the human, and number of finished pieces of the robot. Figure 3.8 shows the results. After removing one outlier subject, who tested the robot extensively such that idle times deviated from the other subjects, we ran non-parametric ANOVA with Kruskal-Wallis and a posthoc Conover's test, since the data showed no normal distribution according to the Shapiro-Wilk test. We chose a significance level of $\alpha = 0.05$ which is common in the literature (Lasota and Shah, 2015). The tests show that the human idle time was significantly higher when the robot was present compared to the no robot mode (p<0.0004). The average idle time was higher in the temporal mode than in the spatial mode and in the unreactive mode. In the experiments we noticed that it was after some time very easy for the subjects to adapt their own task rhythm to the constant rhythm of the robot in the unreactive mode. The robot idle times at the non responsive mode result from the grasping time of the hand. The total trajectory length per assembly of the human shows that in the first experiment without the robot the average length was lower in the modes where the robot was present. As for all subjects the experiment without the robot was the first experiment, this can be explained by task adaptation of the human after the first experiment. The behavior of the robot did not influence the average trajectory length. In combination with the idle times this shows that the human rather stopped and waited for a situation to clarify instead of spatially evading the robot in the experimental setting. The data shows that the number of finished pieces is significantly lower in the temporal mode (p<0.00002). The human always finished 15 pieces as this marked the end of the experiment. The mean assembly time of the humans was smallest in the spatial deformation mode, however no statistical significance was found (p = 0.96).

Figure 3.9 shows the result of the subjective comparison of the three modes. The upper row shows that 32% of the subjects felt most uncomfortable in the

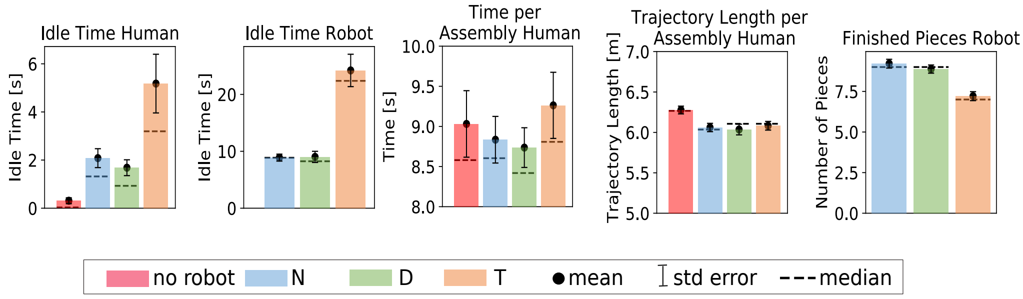


Figure 3.8.: During the user study, we recorded and evaluated motion data from the subjects and the robot for the different adaptation modes, namely no reaction (N), spatial deformation (D) and temporal scaling (T).

unresponsive mode, 52% percent felt most uncomfortable during temporal scaling and 4% during spatial deformation. 8% reported that they never felt uncomfortable. When looking at these two groups individually it shows that the ones that felt uncomfortable with the unresponsive robot in particular felt safe at temporal scaling and the ones that felt most uncomfortable with temporal scaling felt most safe during spatial deformation. In general, subjects reported that they felt more undisturbed at spatial deformation. In the experiment notes subjects reported that during the temporal scaling mode they felt the robot's productivity decreased when the robot needed to stop because of them and this "ineffective task execution" made them feel uncomfortable. Additionally, subjects reported the robot stopping in too close distance to a goal disturbed them. However, another group of subjects reported that they found the motions of the robot to be very controlled, safe, and reactive when it was in temporal scaling mode. Figure 3.10 shows the subjective answers to the questions on the single modes. We ran a non parametric ANOVA using Kruskal Wallis test and posthoc Conover's test on this data. The test showed that all subjects found the robot significantly more adaptive in temporal scaling and spatial deformation mode compared to no reaction ($p=0.0002$). Subjects also felt significantly less disturbed in spatial deformation mode compared to no reaction ($p=0.032$) and they found the robot significantly less predictable in temporal scaling mode compared to no reaction ($p=0.012$). When only considering the subjects that felt most uncomfortable at temporal scaling mode the subjects found the robot significantly more unpredictable in temporal mode than in spatial or non reactive ($p<0.048$), and felt significantly more uncomfortable in temporal scaling than in spatial deformation mode ($p=0.02$). In average the robot was perceived

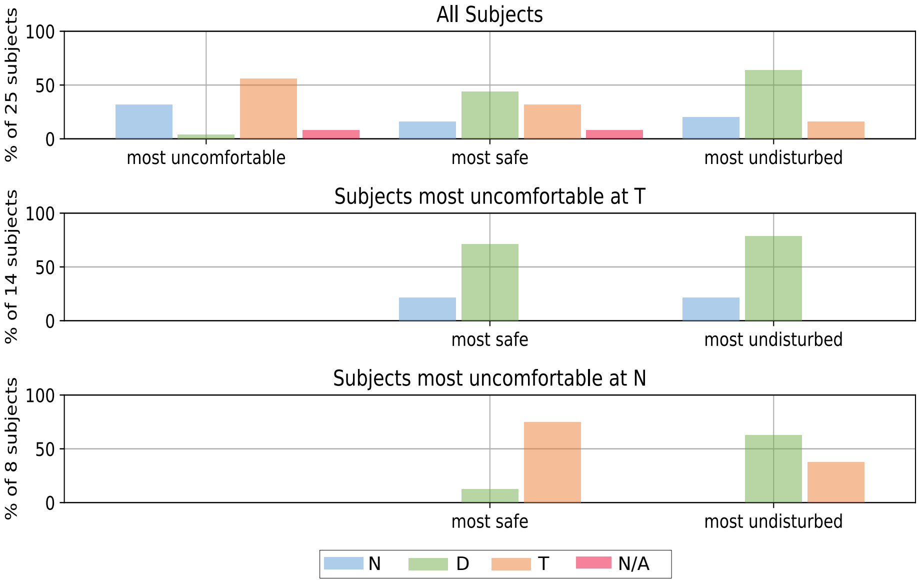


Figure 3.9.: The subjects answered comparison questions on in which mode (no reaction (N), spatial deformation (D) and temporal scaling (T)) they felt 1) most uncomfortable due to the robot's movements, 2) most safe and 3) most undisturbed.The results are visualized as box plots including the median (black horizontal line), interquartile range (box), 1.5 whiskers (fine black lines), and outliers (black circles).

less adaptive in the temporal mode. On the other hand subjects that felt most uncomfortable in the non reactive mode found in average the temporal mode more adaptive than the spatial deformation mode. In terms of comfort and safety no statistical significance can be found between the modes when looking at data of all subjects.

The results of this user study already provide valuable insights on human reactions to online adaptation of ProMPs. However, for future studies a wide variety of experimental settings should be evaluated in order to get more generalizable insights. In particular, also different slowing down and speeding up behaviors in the temporal scaling mode should be compared, as in the experiments we noticed that to abrupt slowing down or speeding up may irritate the users.

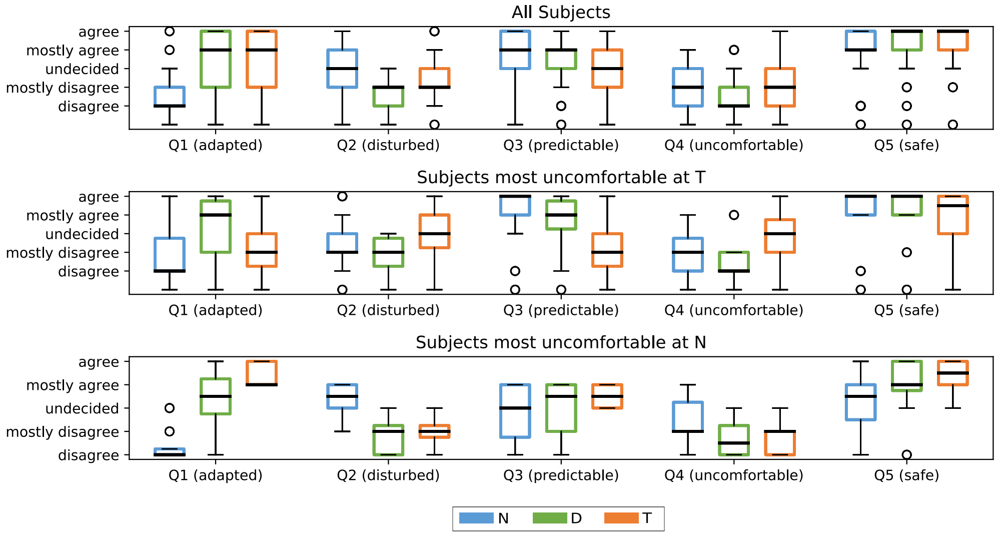


Figure 3.10.: The subjects additionally answered a questionnaire after each experiment mode where we evaluated 5 questions that were answered on a 5 point Likert scale. The results are visualized as box plots including the median (black horizontal line), interquartile range (box), 1.5 whiskers (fine black lines), and outliers (black circles).

3.3.4. Limitations

While the results of the user study provide valuable insights in human reactions to online adaptation of the ProMPs we also want to report limitations of the current approach as noticed during the evaluation with the subjects.

First, our prediction model assumes goal directed motions, which works well for straight motions but leads to changing goal predictions for curved reaching trajectories as they occurred often when subjects reached back from the delivery box to a new part. This would sometimes result in reactions of the robot to these temporarily goal predictions that irritated the human subjects as they did not have insight in the underlying prediction model. We also noticed that if subjects were very hesistant the prediction model at some point predicted a stop of the person which released the robot until the person moved again. For some persons this resulted in a stop and go behavior in the temporal mode which subjects reported to made them feel very uncomfortable. Investigation how to avoiding such problems by fine tuning parameters of the prediction model and the ProMP adaptation online per subject would be an interesting line of future work in our opinions.

3.4. Conclusion

We presented two novel approaches for intention aware online adaptation of ProMPs, namely online spatial deformation and temporal scaling. We evaluated both approaches on a pick and place task with 25 non-expert subjects where we analyzed motion data as well as questionnaires on subjective comfort level and perceived safety. The subjects reported a higher level of perceived safety and felt less disturbed during intention aware adaptation, in particular during spatial deformation. The results indicate that human responses to different kinds of robot behavior do not necessarily generalize across all subjects. In particular, temporal scaling was perceived by one group of subjects as disturbing and unpredictable but as safe and predictable by another group.

Subjects in general felt uncomfortable and got annoyed if they found the robot behavior unpredictable or if they did not understand why a certain robot response was occurring. Therefore, incorporating more communication including motion cues and/or visual feedback should be investigated. Additionally, the experiments revealed that different subjects preferred different robot behaviors. For future work, we plan to investigate how to derive a hierarchical model from these insights that would online classify user types and adapt robot behavior accordingly. Moreover, combinations of spatial and temporal ProMP adaptation could be investigated. Extending the prediction model to incorporate more complex human trajectory behavior and additional intention cues such as gaze direction and body posture is another line for future research.

4. Multi-Channel Interactive Reinforcement Learning for Sequential Tasks

Future robots are expected to cope with a variety of different tasks which renders manual programming of each task highly difficult. One promising approach to learn e.g. new skills from non-expert users is to learn skill libraries from demonstrations (Pastor et al., 2009; Koert et al., 2018). However, even if a robot has already learned a prior skill library it remains a challenge to learn how to sequence such skills correctly to perform more complex tasks. Reinforcement Learning (RL) hereby offers a way for a robot to learn from experience while observing the effects of the chosen actions, i.e. skills, on the environment (Sutton and Barto, 2011; Kober et al., 2013). The main challenges in RL are principally related to the uncertainty that the agent has about the environment it is interacting with, which is usually modeled as a Markov Decision Process (MDP). Since the agent usually cannot access the model of this MDP, there is the need to explore the states of the MDP sufficiently well, in order to understand which actions are convenient to take in which states. This would not be such an issue if the cost of collecting samples was not involved in the problem, but, especially in real robotic applications, the cost and required time for collecting samples can be a limiting factor. Additionally, in real robotic scenarios, there might also be a cost assigned to taking wrong actions that is too high to allow for extensive exploration of the agent since it can lead to breaking valuable objects, the robot hardware or even cause harm to human subjects. One possible way to tackle these problems, speed up the learning and make RL applicable for more robotic tasks is to provide humans possibilities to interact with the robot during the RL learning process. Such interaction provides on the one hand potentially helpful input and advice, but on the other hand also the need to meet the human preferences and the challenge to cope with potentially wrong input and suggestions of the human. The use of human feedback in RL is thereby not new and was already successfully used, for instance, to provide demonstrations of good behavior (Argall et al., 2008), to provide a supplementary reward (Knox and Stone, 2011), or to manipulate the available actions (Cruz et al., 2016). In particular, when it can be assumed that an RL agent does not have access to the background knowledge that a human has, but has to learn solely based on environmental feedback, human guidance can be helpful.

However, many of the results with interactive RL algorithms in the literature were obtained in simulation environments (Thomaz and Breazeal, 2008; Knox and Stone, 2011; Li et al., 2016) and there is a lack of experimental studies for interactions of inexperienced users with RL systems on real robotic tasks (Suay and Chernova, 2011; Knox et al., 2013), in particular under the assumption that human input may partially be incorrect. Therefore, in this chapter, we present a novel interactive RL framework that integrates multiple channels for human input i.e. action suggestion, action prohibiting, feedback and sub-goal definitions and evaluate our approach in two robotic tasks with non-expert users. While our definition of different input channels is similar to the approach in Suay and Chernova (2011), we additionally incorporate a concept of self-confidence that allows the robot to question potentially wrong human input after an initial training phase.

The main contribution of the chapter is hereby the evaluation of our multi-channel interactive RL framework, which includes our concept for self-confidence of the robot, on two sequential robotic tasks with 20 inexperienced human subjects. The second robotic task is specifically designed to investigate the effects of partially wrong human input on the learning process. Hereby, the concept of self-confidence enables the robot to achieve learning progress even if human input is incorrect. Moreover, we evaluate how humans react to the robot's suggestions when the robot considers their input incorrect and which input channels are preferred by humans. We consider this evaluation in particular important for the successful design of future interactive robotic RL algorithms and interfaces for inexperienced users.

The rest of the chapter is structured as follows. Section 4.1 summarizes related work. Section 4.2 introduces our framework for multichannel interactive RL and the concept of self-confidence of the robot. In Section 4.3, we evaluate the proposed approach on two sequential robotic tasks, report our results obtained in experiments with human users and discuss the results and limitations and Section 4.4 concludes the chapter and gives an outlook on possible directions for future work.

4.1. Related Work

Different forms for human input to Reinforcement Learning have been proposed in the literature. We first give a brief overview of these approaches, where we mainly focus on the use of human feedback and human action advice in RL algorithms. Subsequently, we discuss related work that integrates multiple human input channels and existing evaluations of such approaches on robotic tasks and with human subjects.

As one of the first approaches, inspired by animal clicker training, human feedback has been used as a reward signal in Reinforcement Learning with a robotic dog by Kaplan et al. (2002) and animated characters by Blumberg et al. (2002) to reinforce desired behaviors. The use of object and action related feedback in a sequential kitchen task was evaluated by Thomaz et al. (2005), where users interactively trained an agent in a simulated kitchen environment. Knox and Stone (2008) introduced the TAMER framework to learn a human reward model from infrequent feedback and extended their approach for combining the learned human feedback model with environmental rewards in Knox and Stone (2010). Knox and Stone (2011) discusses different ways of combining a model learned from human input with RL, namely reward shaping, Q-augmentation, control sharing and action biasing. Our concept for self-confidence matches the definition of the here introduced combination parameters, however, Knox and Stone (2011) did not use these parameters to allow the robot to question human input, but only for the action selection mechanism.

Judah et al. (2010) proposed to iterate between practice sessions and sessions where users can label trajectory data with good or bad feedback, which can be useful in situations where e.g. realtime feedback is impossible due to speed or where demonstrations are hard to provide due to difficult control, however, this is not the case for type of tasks we consider in this chapter. Griffith et al. (2013) introduced the ADVISE algorithm which uses policy shaping to incorporate human feedback and treats human feedback not as an evaluative reward, but as a label on the optimality of the policy.

Besides the use of human feedback signals also action advice during the RL process was explored in the literature. Maclin and Shavlik (1996) were one of the first to include human action suggestions in reinforcement learners. Wang et al. (2003) used such action suggestions and subgoal suggestions to bias the reward in robot navigation tasks. Kuhlmann et al. (2004) present a framework where a soccer robot can learn from advice in the form of natural language commands which is combined with an RL function approximator. Argall et al. (2008) introduced advice operators for continuous state and action spaces and evaluated them on a segway positioning task where post-execution feedback was used as demonstration data. Moreno et al. (2004) introduced the concept of supervised RL where multiple advise sources are combined together with the learned policy and a decision module decides on which actions to take. They introduce credit assignment for different advice sources, which relates to our concept of self-confidence, but in contrast to their approach, we use a shared control strategy for the selection between human advice and the robot's policy and increase the self-confidence only after an initial training phase. Moreover, Moreno et al. (2004) did not evaluate their concept for credit assignment in interaction with human subjects. It was also explored how agents can advise other RL agents over action suggestions, which might not directly correspond to human advice but provides valuable insights to beneficial advice strategies (Torrey and Taylor, 2012, 2013) and has been additionally linked to the concept of affordances to reduce the state and action space during exploration (Cruz et al., 2016). Here, Torrey and Taylor (2013) report that different ways of advising may become beneficial when teaching on a budget such as early, importance, mistake correcting and predictive advising and Cruz et al. (2018) studied on a simulated robot domestic scenario which types of advisors are most beneficial during learning of the agent.

Human input to RL has also been used in combination with policy search methods and to improve robot skills on a trajectory level (Celemin et al., 2019; Celemin and Ruiz-del Solar, 2019, 2016) which is also very relevant for robotic applications, however, it should be noted that in this chapter we focus only on the sequencing of skills as high-level actions.

The combination of multiple human inputs in RL algorithms was proposed e.g. over the reward structure (Wiewiora et al., 2003), initially unknown spoken word commands for positive or negative feedback and action suggestion for a robotic pick and place task (Grizou et al., 2013) or protocol programs as an intermediate layer between RL-agent and environment to combine different RL algorithms with different forms of human feedback (Abel et al., 2017). While the approach of Abel et al. (2017) is highly related to our work in terms of the idea of incorporating different inputs in a modular structure it was not evaluated on real robotic tasks with human users but only in simulated environments such as grid world and pong. Our combination of channels for feedback and action advice also relates to the approach of Suay and Chernova (2011), but additionally incorporates the concept of self-confidence to allow the robot to question human input.

Overall, only a few studies exist about applications of interactive RL frameworks on real robotic applications (Suay and Chernova, 2011; Knox et al., 2013) and their

evaluations with inexperienced human users. Human users can thereby significantly differ from simulated oracles and studies with real subjects hereby provide valuable insights into actual human behaviors. Isbell et al. (2006) reported that in environments with humans many RL assumptions e.g. on the reward can be violated due to drift or bias. Thomaz et al. (2006) conducted a user study with a simulated kitchen task where they reported that humans also want to provide a future reward to guide the robot. They also found that users used object-specific rewards to guide the robot even though it was conceived as feedback. Thomaz and Breazeal (2008) report that users might use feedback as a motivation signal and that humans may change their reward signal when the agent is learning. Judah et al. (2010) showed that humans might get annoyed if an RL- algorithm does not directly respond to their feedback and Loftin et al. (2014) state that there is a need to learn models of human feedback strategies and suggest to design RL-agents to understand and adapt to different users' training strategies.
However, these prior studies consider only tasks, where in general the humans gave only useful input and to the best of the author's knowledge there is no related work on applications of interactive RL for robotic tasks that were specifically designed to include incorrect human feedback and where the robot would start to actively question the human input during the learning process.

4.2. Multi-Channel Interactive Reinforcement Learning

In this section, we present our approach to integrate multiple channels for human input into a framework for interactive RL. We hereby focus on learning sequential tasks. This refers to learning how to sequence high-level actions which can e.g. represent skills in the form of motion primitives that the robot already learned before. In a RL setting an agent, which is the robot in our case, interacts with the environment and tries to learn a policy π to decide which action a it should choose in a state s to maximize its accumulated rewards. A reward r is given after each transition into a next state s' and the rewards are accumulated over an episode of a finite number of time steps t. While in the classical RL setting the robot learns independently, in interactive RL the robot may also incorporate human input during the learning process.
We first discuss the relevant channels for human input during RL of sequential robotic tasks in Section 4.2.1. Afterwards, we present the different components of our multi-channel interactive RL framework, namely the RL-module in Section 4.2.2, the human-advice module in Section 4.2.3, the Action Selection mechanism

in Section 4.2.4 and our novel concept of self-confidence of the robot in Section 4.2.5. Figure 4.1 shows an overview of the approach and Algorithm 2 summarizes the different steps during learning, where i refers to the episode index. In Section 4.2.6 we provide details on the current implementation for the components of the framework, which was used for the experimental evaluations in this paper.

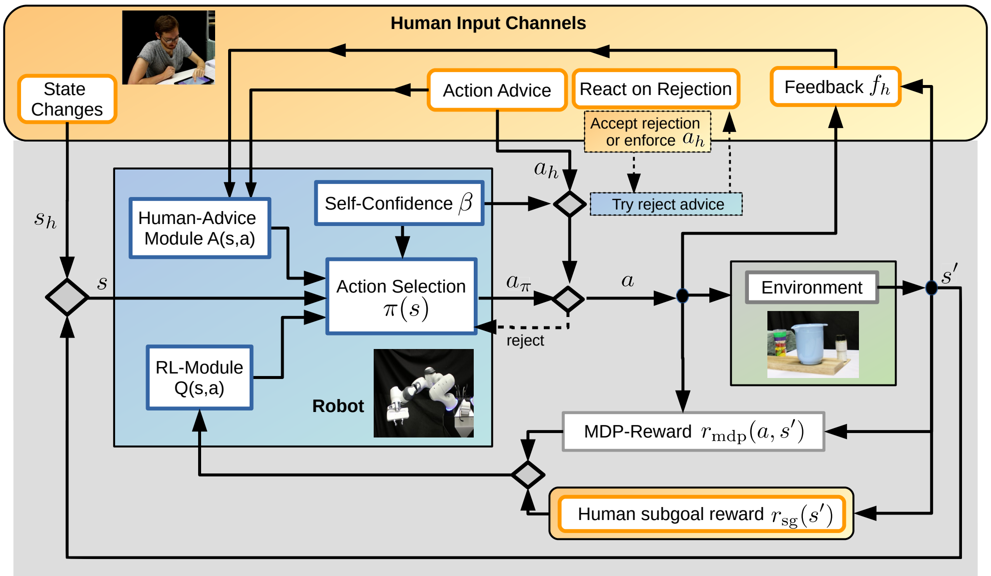


Figure 4.1.: We propose an approach that integrates multiple human input channels in an interactive Reinforcement Learning framework for sequential tasks. The human can hereby prevent or suggest actions, give feedback after action execution, modify the state or define subgoal rewards. Through interaction with the environment and the human input, the robot learns a policy that tries to maximize the overall reward and chooses actions based on the Human-Advice Module, the RL-module, and its own self-confidence. The self-confidence also eventually enables the robot to question human action advice if it contradicts the robot's policy.

4.2.1. Human Input Channels

We use human input as a valuable source of information to guide the exploration of a robot in a RL setting, speed up the learning, and prevent disasters which could be caused by exploratory actions in real robotic scenarios. In our approach, we, therefore, consider multiple input channels, such as action suggestions or prevention of action executions, feedback after the execution of an action, subgoal reward definition, or state modifications by the human user.

Human Advice on Planned Actions

Our framework allows the human to give specific advice on the next planned action of the robot. This input channel can be used before the actual execution of an action by the robot. Therefore, the robot first communicates the planned action according to its policy a_π to the human. The human then has two options to react to the proposed action with action advice a_{h}. First, the human can simply reject the proposed action which is represented by $a_{\mathrm{h}} = -1$. In case of such a rejection, the robot in the next step might propose another action according to its updated policy. The second option for the human is to suggest an alternative action to the robot which is indicated by $a_{\mathrm{h}} \geq 0$. This action then overwrites the proposed action in a supervised RL-fashion. If there is no input from the human on the proposed action, i.e. $a_{\mathrm{h}} = \emptyset$ the robot just executes the proposed action. To summarize, the action a to be executed in the next step is chosen as

$$a = \begin{cases} \emptyset & \text{if} \quad a_{\mathrm{h}} == -1 \\ a_{\mathrm{h}} & \text{if} \quad a_{\mathrm{h}} \geq 0 \\ a_\pi & \text{if} \quad a_{\mathrm{h}} == \emptyset \end{cases} \tag{4.1}$$

The option of rejecting actions is particularly important to prevent disaster actions in real scenarios. We assume that this option can be used even if the human has no idea what the optimal action would be, but still recognizes potentially disastrous actions. The active suggestion of actions by the human can be used in tasks where the human knows how to guide the exploration of the robot.

Human Feedback after Action Execution

After the execution of an action a the human can provide feedback f_{h} for this action. For our experiments we consider three options for feedback that is ‘Good’, ‘Bad’ or ‘I don’t know’ represented by

$$f_{\mathrm{h}}(s, a) = \begin{cases} 1 \\ -1 \\ 0 \end{cases} \tag{4.2}$$

If no feedback is provided by the human this is also counted as neutral feedback $f_{\text{h}}(s,a) = 0$. The option to provide feedback after action execution can be beneficial in tasks in which the human is not able to foresee the effects of choosing a specific action but is able to judge it afterwards.

Human Subgoal Rewards

Action advice and feedback after action execution are incorporated online during learning. Additionally, we provide an input channel to define subgoal rewards before the learning phase for the task starts. These subgoal rewards reward the occurence of certain states and are limited to a discrete state space in the current formulation. A subgoal reward $r^i_{\text{sg}}(s)$ is hereby defined for part of the state vector $s = \{s_0, .., s_d, ..s_D\}$

$$r^i_{\text{sg}}(s) = \begin{cases} 1 \text{ if, } s == s_{\text{sg}} \\ 0 \text{ else} \end{cases} \tag{4.3}$$

and the human may define multiple subgoal rewards. The final subgoal reward definition is given by

$$r_{\text{sg}}(s) = \sum_i r^i_{\text{sg}}(s). \tag{4.4}$$

All human-defined rewards are constant throughout the learning, thus need to be specified for a task apriori by the user. We, therefore, consider this channel most useful in tasks where a human is sure about the subgoals before task execution. In the current version, our framework does not handle cases in which the human could change subgoals because they notice during the interaction that the originally defined subgoals were wrong. Therefore, the definition of subgoals requires a good understanding of the task by the human which can be a limiting factor in more complex tasks.

Human State Modifications

In many situations, it might also be helpful for the robot's learning to make use of the option that a human can physically modify the state of the environment. Hereby, the next state s' can be changed by human interference with the environment. This can be used e.g. to undo the effects of an action, to let the robot experience the same situation again, or to have a reset to a specific state after the end of an episode. The environmental state modification by the human is modeled as

$$(s)^{t+1} = \begin{cases} (s')^t & \text{if} \quad s_\mathrm{h} == \emptyset \\ s_\mathrm{h} & \text{else,} \end{cases} \tag{4.5}$$

where t denotes a single step of an episode and s_h is the environmental state after the modification by the human. In particular, such modifications can help the agent to reach states that otherwise would take longer to be reached through pure exploration and help the agent to gather experience in these states. State modifications can also enable the human to let the agent visit important states more frequently.

4.2.2. Reinforcement Learning Module

Whenever the robot takes an action a in a state s it transits to the next state s' and receives a reward that consists of the environmental reward r_mdp and the human subgoal reward r_sg, which was introduced in Section 4.2.1. The goal of the robot is to maximize the combined reward

$$r(s, a) = r_\mathrm{mdp}(s, a) + \eta r_\mathrm{sg}(s, a), \tag{4.6}$$

where η is a factor to scale the influence of human subgoal rewards which needs to be handtuned in our current version of the framework. The robot therefore uses the samples s, a, s', r to learn a policy $\pi(s)$ that maximizes the total reward.
The Reinforcement Learning module hereby learns a Q-function $Q(s, a)$, that is the cumulative reward for starting in s, applying action a, and, in the resulting state s', act optimally.

4.2.3. Human-Advice-Module

For our approach, we use human input during learning in two ways. On the one hand, direct action suggestion influences the exploration which might change the next performed action by the robot. This is comparable to a supervised RL setting (Moreno et al., 2004). On the other hand, since human input might be sparse and the human might not repeatedly give input in the same states, we also learn a model of human feedback and advice, such that this model can be used to guide the agent even if no human input is given. This is comparable to the approach of learning a model for human reward as proposed in Knox and Stone (2008), even though their model is solely based on human feedback as an input.

In our framework, the human advice module, therefore, learns a function $A(s,a)$ that indicates which action is the human would most likely suggest or provide positive feedback on. Since we assume that both human feedback and advice would try to guide the robot in the same way, we learn a joint human advice module from feedback and advice.

4.2.4. Action Selection

Based on the RL module and the human advice module the robot decides in each step which action a_π to present to the human for potential advice. The literature contains different comparisons of how to combine a RL-based Q-function and a human advice module. In Knox and Stone (2011) it is discussed that methods that act on the action selection rather than change the Q-function directly generally work better and outperform other combination methods. Following this argumentation, we believe it is beneficial to use a control sharing approach for action selection in our framework.

The robot hereby follows the human advice module with the probability $1-\beta$, if the advice module is not indifferent about all actions in the state

$$\begin{aligned}
&\text{if not} \quad A(s,a_j) == \max_a(A(s,a)) \quad \forall \quad a_j, \\
&\text{with probability } 1-\beta, \quad a_\pi = \arg\max_a[A(s,a)], \\
&\text{with probability } \beta, \qquad a_\pi \quad \text{according to policy based on RL-module}, \qquad (4.7)
\end{aligned}$$

where β denotes the self-confidence of the robot. Our concept of this self-confidence is explained in detail in Section 4.2.5. Alternatively, with probability β, the robot follows a policy based on the Q-function of the RL module.

4.2.5. Self-Confidence

If the human understands the task at hand well and provides useful input, human input can speed up the learning of the robot. However, incorrect human input can also slow down or even prohibit learning of the task. Therefore, we introduce the concept of self-confidence β of the robot into our interactive RL framework. First, this self-confidence is used as a combination parameter of the RL and the human advice module, as described in Section 4.2.4. Second, the self-confidence can also be used by the robot to question an action suggested by the human if it contradicts the robot's learned Q-function. This can be expressed by the probability of trying to reject a human suggestion $p(\text{reject } a_{\text{h}}) = \beta$. Such a rejection is implemented in our framework as a feedback message to the human whenever the robot considers the human action input to not be beneficial. However, it still leaves the freedom of choice to the human such that they can decide whether to execute the originally advised action regardless or rather follow an alternative suggestion based on the robot's Q-function.

At the beginning of the learning process, the robot has no own experience, which is represented by a low self-confidence e.g. $\beta = 0$. Due to that, it will follow all suggestions given by the human or the human advice module and always assume human input to be beneficial and correct. However, while the robot learns from its own experience it will slowly converge towards a more informative Q-function and can eventually distinguish between good and bad human advice. With this, the self-confidence can eventually increase during learning, allowing the robot to question and deal with potentially incorrect human input. As such, the self-confidence needs to be computed such that it provides the robot a notion of the quality and convergence of its learned policy, which is represented by the Q-Function of the RL-module. In particular, the self-confidence can vary for different states and should relate the robot's trust in its own policy with its trust in the human input.

4.2.6. Component Implementation

This subsection presents the implementation that was chosen for the single components of the framework in the experimental evaluation for this paper. While for now the chosen implementations follow rather simplistic approaches and are tailored for our experimental setting the modularity of the framework allows easy replacements with more complex implementations of single components for future applications.

Tabular Q-Learning Reinforcement Learning Module

In the experiments in this paper, we use tabular Q-Learning as a RL-algorithm. The Q-function is hereby represented by a table with $S \times A$ entries, where S is the total number of states and A the total number of actions. This table is initialized with zeros. In Q-Learning (**?**) for each sample $< s, a, s', r >$ the Q-function is updated

Algorithm 2 MINT-RL

```
init $Q, A$ e.g. tabular as $\boldsymbol{Q}[s,a] = 0$ and $\boldsymbol{A}[s,a] = 0 \quad \forall s, a$
init visits per state $\boldsymbol{v} = 0 \quad \forall s,$
init $\beta = 0, \quad s = s_0, \quad i = 0$
while i $<$ Maximum Episodes do
    $\boldsymbol{v}[s] = \boldsymbol{v}[s] + 1$
    Chose $a_\pi$ from action selection policy $\pi(s, \boldsymbol{Q}, \boldsymbol{A}, \beta, \boldsymbol{v})$ e.g. Shared Control with $\varepsilon$-greedy Alg. 5
    present $a_\pi$ to human
    $a_\text{h} \leftarrow$ human action advice
    if $a_\text{h} == \emptyset$ then
        $a = a_\pi$
    else
        if $a_\text{h}$ not optimal according to $Q(s,a)$ then
            $p$ = random sample from uniform distribution
            if $p < \beta$ then
                suggest human to reject $a_\text{h}$
                if human accepts rejection then
                    $a = a_\pi$
                else
                    $a = a_\text{h}$
            else
                $a = a_\pi$
        else
            $a = a_\text{h}$
    $s$' $\leftarrow$ execute $a$ in $s$
    $r \leftarrow r_\text{mdp}(a, s') + r_\text{sg}(a, s')$
    $f_\text{h} \leftarrow$ human feedback
    update $Q$ from $r, s, a, s'$ e.g. Tabular Q-Learning Alg. 3
    update $A$ from $s, a_\pi, a_h, f_h$ e.g. Tabular Human-Advice-Module Alg. 4
    if human changes the state then
        $s = s_\text{h}$
    else
        $s = s'$
    update $\beta$ e.g with const linear increase Alg. 6
    i++
```

according to

$$Q(s,a) = Q(s,a) + \alpha(s)(r(s,a) + \gamma \max_a Q(s',a) - Q(s,a)), \tag{4.8}$$

where $\alpha(s)$ is the learning rate in state s and γ is the discount factor. We chose here to decrease the learning rate over time dependent on the number of visits $v(s)$ of a state that is $\alpha(s) = 1/v(s)$, as this is a common practice in the literature. Therefore we initialize a vector $\boldsymbol{v}$ of length S with zeros and update it whenever a state is visited. Algorithm 3 summarizes the update procedure of the Q-function for each sample. For future applications, this implementation of the RL-module could be replaced by another off-policy RL-algorithm, if desired.

Tabular Human Advice Module

For simplicity of the evaluations, in this paper, we represent the human advice module as a tabular function $A(s,a)$, which we initialize with zeros for all state-action pairs. This tabular function is updated whenever human advice or feedback is received.

In particular, when the human suggests an action a_h in a state s we increase the value of $A(s,a_h)$ and if the human rejects a suggested action a_π (indicated by $a_h == -1$) we decrease the value of $A(s,a_\pi)$.

Algorithm 4 Tabular Human Advice Module

```
input: $s, a_\pi, a_h, a, f_h, \boldsymbol{A}$
if $a_h$ not $\emptyset$ then
    if $a_h == -1$ then
        $\boldsymbol{A}[s,a_\pi] = \boldsymbol{A}[s,a_\pi] - 1$
    else
        $\boldsymbol{A}[s,a_h] = \boldsymbol{A}[s,a_h] + 1$
$\boldsymbol{A}[s,a] = \boldsymbol{A}[s,a] + f_h$
```

Algorithm 3 Tabular Q-Learning Update

```
input: $r, s, a, s', \gamma, \boldsymbol{v}, \boldsymbol{Q}$

$\alpha = 1/\boldsymbol{v}[s]$
$\boldsymbol{Q}[s,a] = \boldsymbol{Q}[s,a] + \alpha(r + \gamma \max_{a'} \boldsymbol{Q}[s',a'] - \boldsymbol{Q}[s,a])$
```

For human feedback which follows after an action a was performed in state s, we increase or decrease the values of $A(s,a)$ accordingly. The implementation for the update of the human advice module is summarized in Algorithm 4.
It should be noted that this simplistic view on the human advice module can be easily exchanged by any more complex model, which learns a similar function from human advice and feedback.

Shared Control with ε-greedy Policy for Action Selection

We implemented a shared control approach between the human advice model and an ε-greedy policy based on the RL module. The robot hereby follows the human advice module with a probability of $1-\beta$, if the human advice module is not indifferent about all actions in the state, as described in Section 4.2.4. Alternatively, the robot follows an ε-greedy policy based on the Q-function of the RL module. Thereby, it selects a random action with a probability of $\varepsilon(s)$ and otherwise chooses the action according to the maximum of the Q-function as

$$a_\pi = \arg\max_a [Q(s,a)] \tag{4.9}$$

Hereby we decrease $\varepsilon(s)$ based on the number of visits v_s of a state according to $\varepsilon(s) = 1/\sqrt{v_s}$, which is a common choice in the literature. However, since in interactive RL the number of learning steps can be much lower than in classical RL we think in future work different forms of computation should be investigated.
In case the human advice module gives equally good advice for more than one action we follow an ε-greedy policy with the RL module on this subset of actions. Algorithm 5 summarizes this policy.

Heuristic Increase for Self-Confidence

In general, the increase of the self-confidence should resemble the convergence of the learning process of the RL-module. The choice of a good theoretical model for such an increase is not straightforward. Due to that, in our experiments, we tailored the increase of the self-confidence for the chosen problems and constantly increased the self-confidence by a predefined factor after a defined number of initial training episodes I_{min}. Algorithm 6 summarizes this heuristic increase. Even if the current implementation does not represent an actual self-confidence of the robot in the correctness of its policy, being able to question the human's inputs can provide valuable insights to human reactions to the general concept of robotic self-confidence.

Algorithm 5 Shared Control with ε-greedy Policy

```
input: $s, \boldsymbol{Q}, \boldsymbol{A}, \beta, \boldsymbol{v}, \boldsymbol{\varepsilon}$
$\varepsilon = 1/\sqrt{\boldsymbol{v}[s]}$
$p_1$ = random sample from uniform distribution
if $p_1 <= (1 - \beta)$ then
    $\boldsymbol{a}_A = \arg\max_a \quad \boldsymbol{A}(s, a)$
    if len($\boldsymbol{a}_A$) == 1 then
        $a_\pi = \boldsymbol{a}_A$
    else
        $p_2$ = random sample from uniform distribution
        if $p_2 < \varepsilon$ then
            $a_\pi$ = random choice from $\boldsymbol{a}_A$
        else
            $\boldsymbol{a}_{A,Q} = \arg\max_a \quad \boldsymbol{Q}(s, a) \quad \forall a$ in $\boldsymbol{a}_A$
            $a_\pi$ = random choice from $\boldsymbol{a}_{A,Q}$
else
    $p_3$ = random sample from uniform distribution
    if $p_3 < \boldsymbol{\varepsilon}$ then
        $a_\pi$= random choice between all actions
    else
        $\boldsymbol{a}_Q = \arg\max_a \quad \boldsymbol{Q}(s, a)$
        $a_\pi$ = random choice from $\boldsymbol{a}_Q$
```

Algorithm 6 Increase Self-Confidence Heuristic

```
input $i$ and $\beta$
if $i > I_{\text{min}}$ then
    $\beta = \min\{\beta + \delta_\beta, 1\}$
```

4.3. Experimental Evaluation

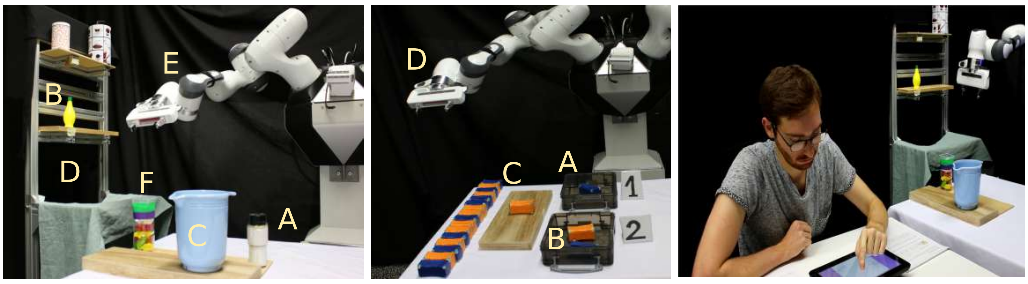
In this section, we evaluate our approach on two sequential robotic tasks (shown in Figure 4.2) with both, simulated and real human input from experimental evaluations with 20 users (12 male, 8 female). Regarding their age groups, two participants were between 18 and 20 years, 10 were between 21 and 29 years, 5 were between 30 and 39 years and three were between 50 and 59 years old. In addition, the participants were mostly inexperienced with robots, which means 11 of the participants reported never having interacted with a robot before our study, four having one other encounter before our study, another four having one to ten encounters and only one having more than 20 encounters with a robot before the study. Concerning the obtained results, we want to point out that 20 subjects are only a small sample size and we believe the results can therefore only indicate trends and tendencies.

In the experiments with simulated human feedback, we show the principle influence of different input channels on the tasks. In the first robotic task, we evaluate how the real human subjects use and understand the input channels, and which types of input they prefer. In the second robotic task, we additionally investigate how humans react to the concept of self-confidence of the robot and how they respond if a robot starts to make own suggestions, once it recognizes human input might be incorrect.

In the following, we report results for both tasks with simulated human input and subsequently the findings from the conducted human experiments. Hereby, all statistical tests are performed on a significance level of $\alpha_\mathrm{s} = .05$. For the Mann-Whitney-U tests, we report the test statistic U, the standardized statistic z, the p-value, and the sample sizes n_1 and n_2. For the Wilcoxon signed-rank test we report the test statistic T, the standardized statistic z, the p-value, and the resulting sample size n after removing samples with zero differences.

4.3.1. Robotic Kitchen Task

In the first robotic task, we evaluate the influence of human input during interactive RL when it can be assumed that human input is mainly correct and the human has sufficient prior knowledge on how the task should be solved. The task is inspired by Sophie's kitchen task from Thomaz et al. (2005). Since the focus in the evaluation of this task is on the comparison of the human input channels we did not use the concept of self-confidence, assumed input to be always correct in the simulation and disabled the option for the robot to question human input.



(a) (b) (c)

Figure 4.2.: We evaluate the proposed framework on two sequential robotic tasks. (a) In the first task, the robot should learn how to finish a cocktail by pouring sugar (A) and lemon-juice (B) into a bowl (C). At the start of an episode, ingredients are either at the shelf (D) or close to the bowl and the robot starts in his start position (E). An episode fails if the robot pours ingredients onto the floor or adds the ingredient chocolate (F) to the bowl. (b) In the second task, the robot should learn to sort objects according to their weight into two boxes (A, B). However, this sorting criterion is not known to the human such that they might first think they should be sorted by color, which results in partially incorrect human input. In each episode, one object (C) is presented to the robot that starts at his start position (D) and the episode ends after the object was sorted in a box or unsuccessfully dropped at the start point. (c) We evaluate both tasks with 20 inexperienced human subjects that interact with the robot over a graphical user interface.

In our kitchen task, the robot should learn to add specific ingredients (that are known to the human) to a bowl in order to complete a cocktail. At the beginning of the task, all ingredients can be either on the shelf or close to the bowl. At least one ingredient is at each of those locations and the robot starts at his start position as depicted in Figure 4.2. The state space of the corresponding MDP is formally defined by the position of the arm, which can be either AT-BOWL, AT-HOME, or AT-SHELF, the positions of the objects, which can be AT-SHELF, CLOSE-TO-BOWL or IN-ARM and the state of the bowl which is defined by the contained ingredients. A bowl state where object 1 was added to the bowl but objects 2 and 3 are not would be represented by $< 1, 0, 0 >$. This definition of the state space with N objects results in $3 * 3^N * 2^N$ states. The actions are defined as GO-TO-BOWL, GO-

TO-SHELF, GRASP-OBJ-X, POUR, and PUT-DOWN, which results in 4+N actions. Not all actions are available in all states, e.g. the robot can only pour or put something down if it grasped an object before, and it can only grasp objects at the location where it currently is e.g. at the shelf or close to the bowl. An episode is successfully finished if the bowl contains the desired ingredients. An episode ends unsuccessfully if ingredients are poured to the floor, i.e. choosing the pouring action when the arm is not at the bowl, or if wrong objects are poured into the bowl. The reward is defined as

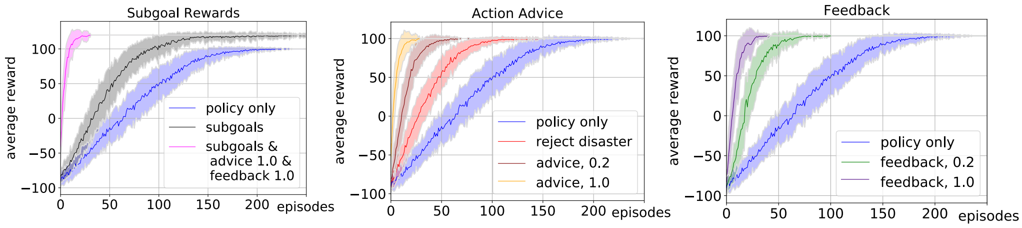
$$r = \begin{cases} 100 & \text{if episode ends successfully} \\ -100 & \text{if episode ends not successfully} \\ 0 & \text{if episode ends due to reaching the maximum number of steps} \\ 0 & \text{in all other states that do not end the episode} \end{cases}$$

In our experiments, we defined the missing ingredients for the cocktail as lemon-juice and sugar. Additionally, the ingredient chocolate is present in the setup. These 3 objects result in 648 states and 7 actions. The task setup is shown in Figure 4.2 (a).

Simulated Human Input

In this section, we evaluate the influence of the different input channels on the learning process of the robotic kitchen task using simulated human input.
First, we evaluate the influence of the human subgoal definition, where we chose $\eta = 10$. The parameter η was hereby hand-tuned with respect to the overall task reward and does not claim to be an optimal choice. In particular, in tasks with a sparse reward structure subgoal rewards can help to guide the exploration and accelerate the learning of the robot. Our simulated human input defines subgoals whenever one of the ingredients sugar or lemon juice is added to the bowl such that the state of the bowl changes to contain part of the desired ingredients, and rewards this with $+10$. Figure 4.3 (a) shows the comparison of the learning with subgoals (black) and without subgoals (blue). To obtain a mean value of the performance after each episode, we averaged the policy learned up to that point over 20 evaluations runs and repeated this for 50 experiments with different random seeds. The plot shows the mean and standard deviation of the average reward plotted over the number of episodes. It shows that subgoal definition results in a steeper learning curve.

Next, we evaluate the influence of different forms of action advice. The human oracle is hereby implemented in a way that it can reject actions that would lead to a disaster or advice actions that can be beneficial for the current state with a predefined probability. Figure 4.3 (b) shows a comparison of the learning curve without action advice (blue), with rejection of actions (red) and action suggestions with probability 1.0 (orange) and 0.2 (brown). The results show that suggestions of correct actions can speed up the learning up to a factor of 10, but even if the suggestions are only provided with lower probability or if only the option of preventing disaster actions is used the learning can be accelerated.
In Figure 4.3 (c) we show the results of the influence of feedback after action execution on the learning process. As in the advice, we assume the simulated feedback to always be correct and given with a certain probability. Again, the learning can be accelerated even if the feedback is not given in every state. The learning is hereby slightly slower than in the case of the advice since the feedback can not actively guide the exploration.



(a) (b) (c)

Figure 4.3.: We evaluate the influence of different types of simulated human input on the learning process of the robotic kitchen task. The plots show the average reward over 50 experiments and 20 evaluation runs. We plot the mean (solid line) and standard deviation (shaded area). (a) shows the influence of subgoal definition (grey) and the results with the combination of subgoals, feedback, and advice (pink). (b) shows the influence of rejecting disaster actions (red) and correct action suggestions with probabilities 1.0 (orange) and 0.2 (brown) for action advice. (c) shows the influence of correct feedback after action execution with probability 1.0 (purple) and 0.2 (green) in comparison to learning without human input (blue).

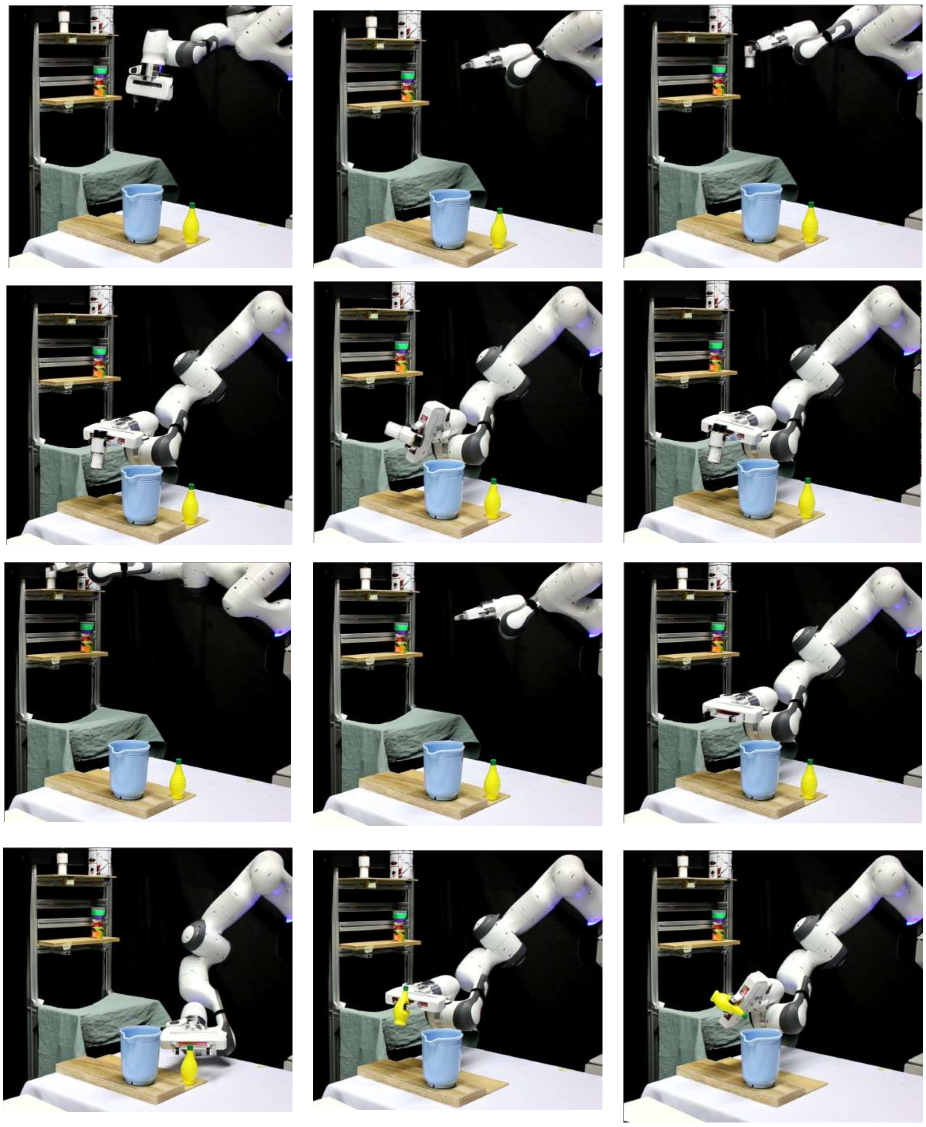
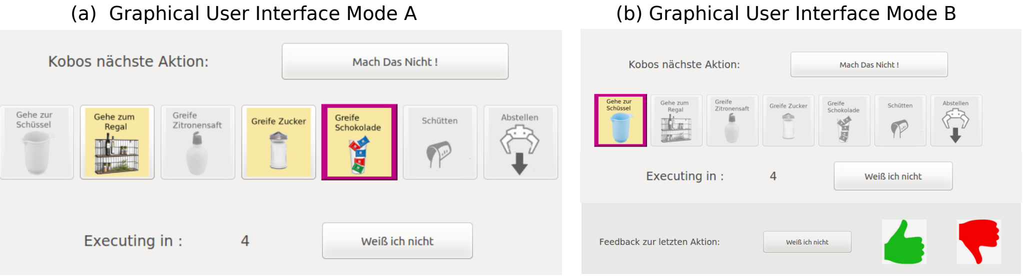


Figure 4.4.: The robot successfully learns to pour the ingredients sugar and lemon juice into the bowl, while incorporating human input. The figure shows the execution of the resulting sequence of the actions GO-TO-SHELF, GRASP-SUGAR, GO-TO-BOWL, POUR, GO-TO-SHELF, PUT-DOWN, GO-TO-BOWL, GRASP-LEMON-JUICE, POUR.


(a) Graphical User Interface Mode A

(b) Graphical User Interface Mode B

Figure 4.5.: The subjects interact with the robot named Kobo through a graphical user interface on a touch display. The main parts of the interface are labeled in German since the study was conducted in Germany. Here, we show the interfaces for the different modes in the Kitchen Task. The interface for the sorting task only differed in the displayed actions. (a) In Mode A, a proposed action is shown to the subjects by highlighting it in pink ("Kobos nächste Action"- Kobo's next action). The subjects then had 10 seconds to give input indicated by a timer running backward ("Executing in: "). The user has the options to suggest their own action (of the available actions in yellow), stop the proposed action and let the robot suggest another action ("Mach das nicht" - Do not do this) or indicate that they are indifferent about the proposed action ("Weiß ich nicht" - I do not know) and let the robot just execute it. If the subjects do not give input within the 10 seconds the robot executes the proposed action. (b) In Mode B, at first, only the upper part of the interface is active and the subjects are shown the proposed action ("Kobos nächste Action"- Kobo's next action) with pink highlighting. They have 5 seconds, indicated by the timer running backward ("Executing in: "), to stop the execution of the proposed action and let the robot suggest a new one ("Mach das nicht" - Do not do this). After the robot executed an action, the lower part of the interface gets activated and the subjects can give feedback about the executed action ("Feedback zur letzen Aktion" - Feedback for the last action) which can be positive (green thumbs up), negative (red thumbs down) or indifferent ("Weiß ich nicht " - I do not know"). If they do not give feedback within 10 seconds the robot continuous to propose the next action and the upper part of the interface gets active again.

Real Human Input

We evaluate our approach on the robotic kitchen task with 20 inexperienced users. We compare two different interaction modes with our framework. In Mode A the subjects can reject actions, that are presented by the robot and can also actively suggest alternative actions. The user interface that allowed these input channels is shown in Figure 4.5 (a) and an example for a successful task execution with human input is shown in Fiure 4.4. In Mode B the subjects only get the options to reject actions presented by the robot and to provide feedback (positive, negative, or 'I don't know') after action execution. The interface for the input channels in Mode B is shown in Figure 4.5 (b). Before the experiments, all subjects were introduced to the overall setting. They were told that they are supposed to help the robot to successfully complete the task at hand. In addition, they received written as well as verbal instructions that precisely explained the task goal (pour lemon juice and sugar in the bowl to complete the cocktail). We randomize the order in which Mode A and B are presented to the subjects to eliminate ordering effects. In each mode, the subjects interacted for 10 episodes with the robot. An episode is finished either by a task success, a failure, or after a maximum number of steps has been reached, which we defined as 30 in our experiments. In each mode and after each episode the participants got feedback whether the episode was finished successfully or unsuccessfully. After each episode, the initial positions of the objects were changed by the experiment supervisor, in the same order of initial locations for each subject. When a mode was completed, participants were asked to fill out experiment notes. These were blank spaces in which the subjects were asked to subjectively report how they interacted with the robot during a given mode and whether they noticed anything specific during the interaction or something stood out to them. Afterward, subjects were asked to fill out a questionnaire (Questionnaire 1) which contained questions about the participants' attitude, experience and impression of the task, the robot, and their interaction and contribution to the task. For this, participants indicated on a five-point Likert scale how much they agreed with statements about how well they thought they could communicate with the robot, how helpful they felt in its success and learning as well as whether they felt like they had control over the robot. Lastly, after completing both modes for the task, they were given a final questionnaire (Questionnaire 2) that directly compared the two modes. In it, subjects indicated in which mode they could communicate best with the robot, in which mode they felt they gave the most useful input and which mode they would prefer overall in future applications. For this, they were also able to refer back to their experiment notes so they could remember each mode correctly.

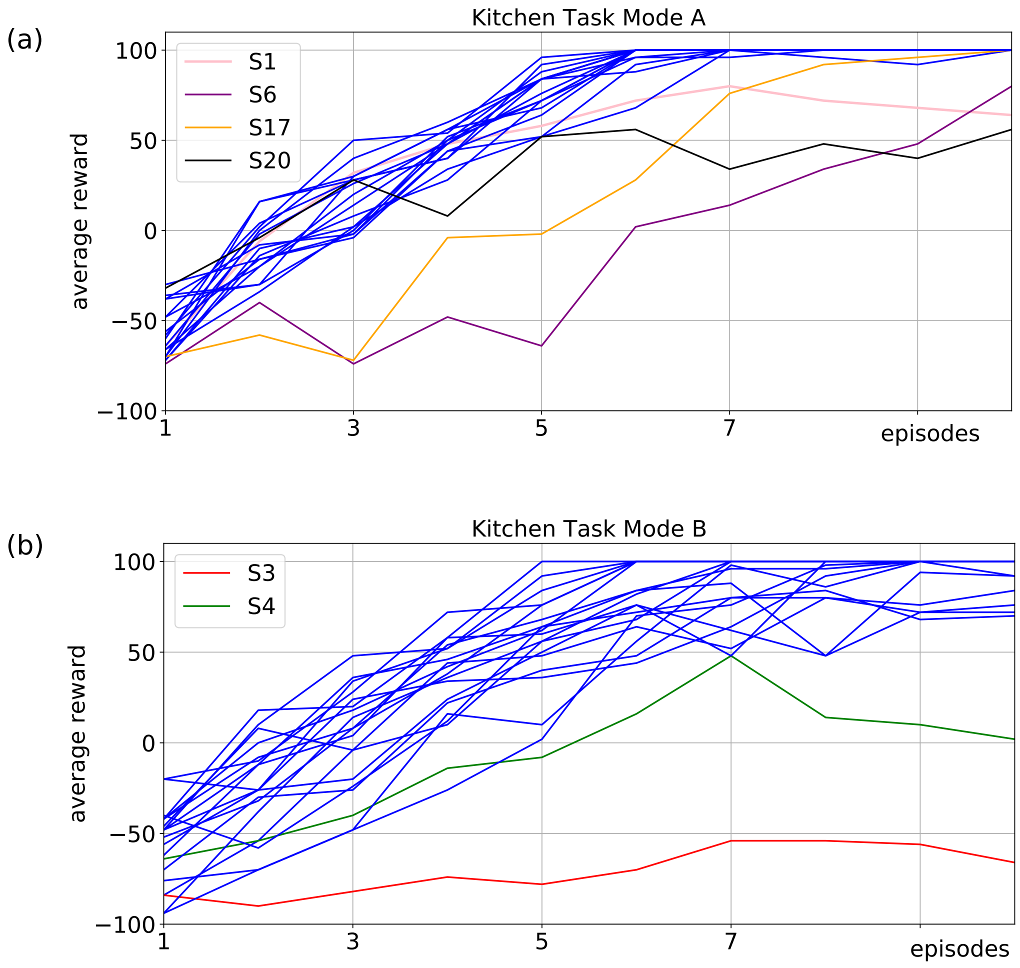


Figure 4.6.: We compare two different interaction modes in experiments with 20 inexperienced subjects on the robotic kitchen task. In Mode A, subjects can prevent action execution and actively suggest own actions. In Mode B, subjects can prevent action execution and give feedback after the action execution. We show the average reward over 10 episodes of interaction, where we plot the mean over 50 evaluation runs per episode for each subject. (a) shows the results for Mode A (pink, purple, orange, and black for highlighted subjects, blue for all other subjects). (b) shows the results for Mode B (red, green for highlighted subjects, blue for all other subjects).

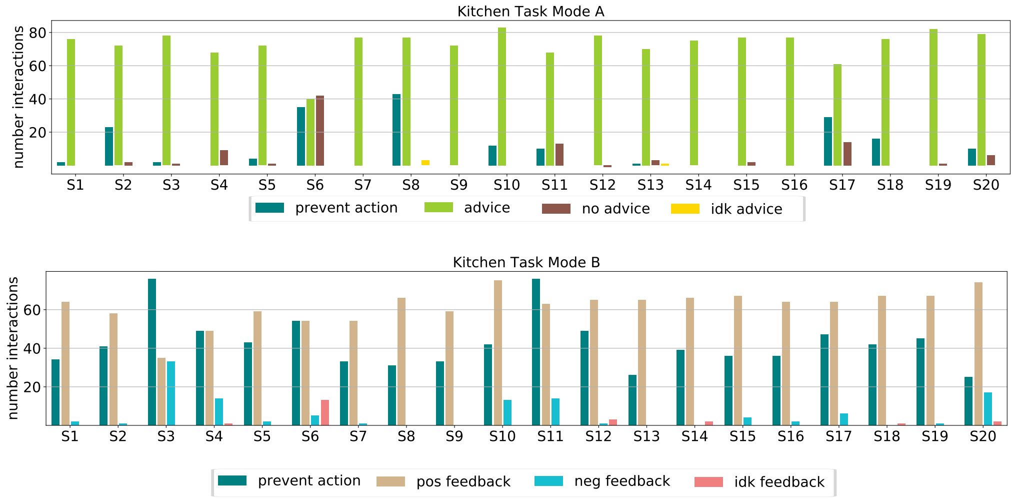


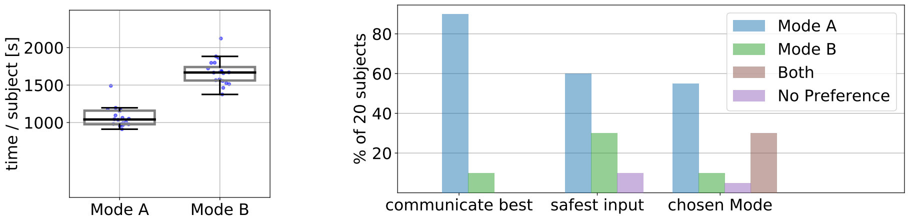
Figure 4.7.: We evaluated how the subject interacted with the robot in Mode A and Mode B. The top row shows the interactions for Mode A split into the different input forms, namely preventing actions, actively suggesting actions (advice) not giving input (no advice) and the 'I don't know' option (Idk advice). Most subjects actively gave advice. Notable is the behavior of subject 6, who let the robot explore a lot on its own, which resulted also in slower learning. The bottom row shows the interactions for Mode B split into the different input forms, namely preventing actions, giving positive, negative, or 'I don't know' feedback. It shows that most subjects rather prevented actions than gave negative feedback. Notable is subject 3, which prevented many actions and gave a lot of negative feedback resulting in worse learning.

This way, subjects could directly compare the modes. Figure 4.6 (a) shows the mean average rewards for all subjects in Mode A and Figure 4.6 (b) shows the mean average rewards for all subjects for Mode B. To obtain a mean value of the performance after each episode, we averaged the policy taught up to that point over 50 simulated evaluations runs for each subject for each episode. The plots show that for most of the subjects the interactive learning could already reach an average maximum reward of 100 after only 10 episodes of interaction. Compared to pure RL without human interaction (Figure 4.3), this results in a speedup of 20 times. We investigate whether human input can significantly decrease the number of episodes it takes to reach an average reward of 80 percent of the maximum reward. One subject in Mode A and two subjects in Mode B never reached 80 percent of the maximum reward and we excluded these subjects from the following statistical analysis. For the remaining subjects it shows that compared to learning with no human input (*Mdn* = 111) in average the learning was faster in both, Mode A (*Mdn* = 5) and Mode B (*Mdn* = 5.5). A Mann-Whitney-U test shows that the differences in comparison to learning without human input are significant in both, Mode A: $U = 0$, $z = 6.38$, $p < .001$, $n_1 = 19$, $n_2 = 50$ and Mode B: $U = 0$, $z = 6.25$, $p < .001$, $n_1 = 18$, $n_2 = 50$.

When comparing Mode A and B, it shows that for 9 subjects Mode A results in faster learning, for 6 subjects learning in both modes was equally fast and for 2 subjects learning in Mode A was slower than in Mode B. However, a Wilcoxon signed-rank test to evaluate if an average performance of 80 percent can be reached in fewer episodes in Mode A than in Mode B shows no significant difference, $T = 18.5$, $z = -1.32$, $p = .188$, $n = 11$.

We highlight the subjects for which the learning did not work so well with different colors in Figure 4.6 to allow for connections to Figure 4.7. Figure 4.7 shows how the users interacted with the robot in Mode A (upper row) and Mode B (lower row). It shows that in Mode A most of the users actively suggest actions, except for user 6 that preferred to give less advice and just let the robot decide in most cases. This results in a slower but still steady learning process, as visualized in Figure 4.6 (a) with the purple line. In Mode B the results show that the subjects use the reject option a lot, mostly in combination with positive feedback. That means most subjects used the action rejection not only to prevent possible disaster actions but to reject any action but the one they want the robot to execute and then give positive feedback for this. Subject 3 used the negative feedback in a way, that whenever the first suggestion of the robot was not correct (and needed to be changed by the subject) the subject still gave negative feedback once the correct action was chosen by the robot. As shown in Figure 4.6 (b) with the red line, this resulted in problems with the learning process. In general, sometimes negative feedback was also used by the subjects to not rate the action choice but the execution of the movement e.g. grasp position at the objects.

Figure 4.8 (a) shows the average amount of time in seconds that it took for the 10 episodes in Mode A and B over all 20 subjects and the underlying data points. We found that Mode A (*Mdn*=1041) was on average less time consuming than Mode B (*Mdn*=1668). A Wilcoxon signed-rank test indicated that this difference was significant, $T = 0$, $z = -3.92$, $p < .001$, $n = 20$. A possible reason for this could be that in Mode A the users directly suggested the actions they wanted instead of excessive rejections until the desired action was presented by the robot (Figure 4.7).



(a)

(b)

Figure 4.8.: (a) We found that Mode A (*Mdn*=1041) was on average less time consuming than Mode B (*Mdn*=1668). A Wilcoxon signed-rank test indicated that this difference was significant, $T = 0$, $z = -3.92$, $p < .001$, $n = 20$. The plots show the median, interquartile range (box), 1.5 whiskers and the underlying data points. (b) In Questionnaire 2 the users answered subjective comparison questions about in which mode they felt they could communicate best with the robot, in which mode they felt most safe their input was useful and which mode they would choose for future interaction. In the last question, they could also choose if they prefer a combination of both modes.

Some users also reported in the experiment notes that they particularly preferred Mode A because of this difference in interaction time. We also noticed during the experiments that some subjects became more distracted and bored if interactions started to become slower and when they could not actively propose actions (but just passively judge or prevent them). After the experiments, the subjects answered questionnaires on how they perceived the individual modes (Questionnaire 1) and for the direct comparison of the modes (Questionnaire 2). Figure 4.8 (b) shows the result of the direct comparison questions. Here a clear majority of 18 users reported they could communicate best with the robot in Mode A. When comparing, in which mode they felt safer to give useful input, most users (12 of 20) choose Mode A. Two users chose Mode B, with one user reporting in the experiment notes that Mode B required less active thinking than Mode A. For further interactions, 55 percent of the users would prefer to use Mode A, 10 percent Mode B and 30 percent a combination of both. Only one user found none of the modes suitable for future use. The answers to the subjective questions in Questionnaire 1 on the individual modes are shown in Figure 4.14. Here, a Wilcoxon signed-rank test shows that the subjects felt they controlled the robot (Q3) significantly less in Mode B than in Mode A, $T = 18.5$, $z = -2.18$, $p = .029$, $n = 14$). Furthermore, they indicated that they could communicate what they wanted to the robot (Q7) significantly better in Mode A compared to Mode B, $T = 22$, $z = -2.46$, $p = .014$, $n = 16$.

4.3.2. Robotic Sorting Task

While we assumed human input to be mostly correct and helpful in the first task, in real applications this might not always be the case. Their input might only be beneficial for solving parts of a task or if the user does not fully understand the task, their input might even be incorrect. We consider such cases to be important and introduced our concept of self-confidence in Section 4.2.5 such that the robot is still able to learn and solve the task at hand eventually. For experimental evaluation of this concept, we designed a second robotic task to investigate how humans react in a situation when the robot starts questioning their (potentially incorrect) input.

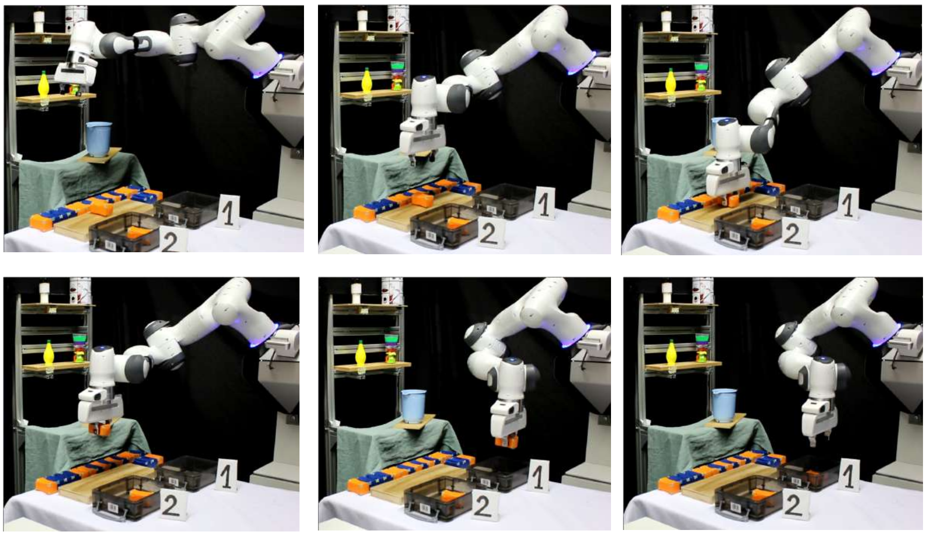


Figure 4.9.: Example of an episode of the sorting task. The figure shows the execution of the sequence of the actions GO-TO-OBJECT, PICK-UP, GO-TO-BOX1, DROP. The objects should be sorted according to their weight, which the robot can measure when it lifts them. The colors of the objects do not correlate with their weights.

In this second sequential robotic task, the robot is supposed to learn how to sort objects into two different boxes. In the beginning, neither the robot nor the human knows anything about the sorting criteria, but they get feedback at the end of each episode (that is sorting of a single item) about whether the episode was successful. The crucial part of this task is that the sorting criterion is defined by the weight of the objects, which can be either low (should go in box 1) or high (should go in box 2). The weights can be calculated by the robot when it lifts an object according to its joint torque sensors but the object weights are not accessible and do not get communicated to the human. However, since the objects also have different colors, as shown in Figure 4.2 (b), which do not correlate with the weights, humans could assume the objects need to be sorted by colors since this is the only accessible criterion for them. This definition of the task results in a situation where even though the human has more prior knowledge about the general structure of the task (i.e. first go to the object, then pick up the object, then bring it to a box), they have no full understanding of the task and might give incorrect input about which box to choose. If the robots questions the human's input based on its self-confidence, on the GUI a message box opens and displays the following: 'I am not sure if it is a good idea to do <USER-ACTION>. I think it is better to do <ACTIONS-ACCORDING-TO-Q-FUNCTION>. Is that okay? ' (Translation from German message by the authors). In case the robots Q-function had more than one optimal action the robot presents all of those in the message box and asks the user whether they want to choose one of them.
The state space of the MDP is formally defined by the weight in the robotic arm which can be EMPTY, HIGH, or LOW and its position which can be AT-HOME, AT-OBJECT, AT-BOX1 or AT-BOX2. This definition of the state space results in 12 states. The actions are defined as GO-TO-OBJECT, GO-TO-BOX1, GO-TO-BOX2, PICK-UP, and DROP which results in 6 actions. Not all actions are available in all states, e.g. the robot can only pick-up the object when it is close to the object and can only drop the object if he has grasped it before. The task is successfully finished if the object is dropped in the correct box. The task is finished unsuccessfully if the object is dropped in the incorrect box or is dropped when the robot is not at a box. The reward is defined as

$$r = \begin{cases} 10 & \text{if episode ends successful} \\ -10 & \text{if episode ends not successful} \\ 0 & \text{if episode ends after maximum number of steps} \\ 0 & \text{in all other states that do not end the episode} \end{cases}$$

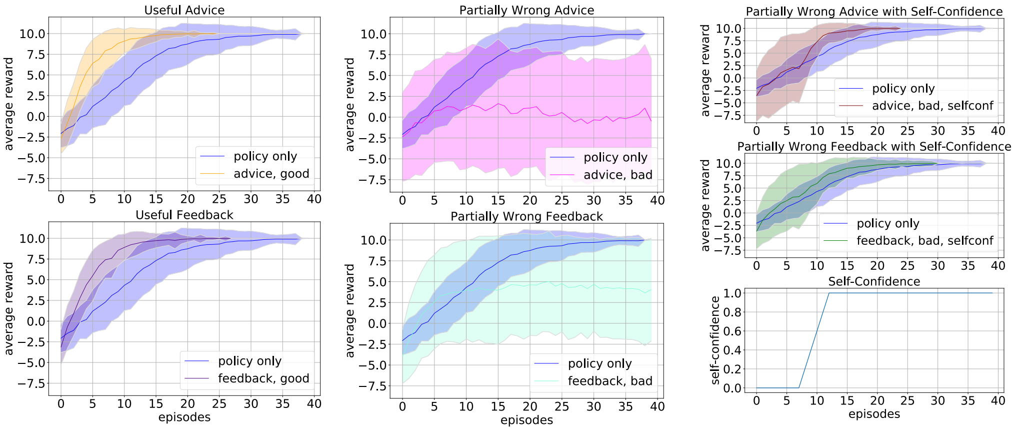
In our experiments, we used two different colors for the objects, orange and blue, however, these colors do not correlate with the weights of the objects. The scenario of the sorting task is shown in Figure 4.2 (b). In the experiments we used the heuristic increase for the self-confidence as explained in Section 4.2.6, with $\delta_\beta = 0.2$ and $I_{\text{min}} = 7$. These values where hand-tuned with respect to the average amount of episodes it took for the RL-module to learn a reasonable policy for the sorting task.

Simulated Human Input

As for the robotic kitchen task we again first evaluate the influence of simulated human input on the learning process of the sorting task. Since for the human the sorting criterion is not obvious nor accessible we assume that there can be no correct human input on which box to choose. However, the human still has a broader picture of how 'sorting' works and can provide help in structuring the task, e.g., in the beginning, the robot should always first go to the object and then pick the object up. We consider human feedback that only provides this structural information but lets the robot explore the rest of the task, i.e, which box to choose. Figure 4.10 (a) shows how such useful feedback can speed up the learning. However, in this task, we can not assume that the human only gives such useful feedback but maybe also starts giving feedback on which box to choose according to his or her own hypothesis about how the sorting works. We simulate this by random input on which box to choose. Figure 4.10 (b) shows that such incorrect input of the human can harm the learning process if it is not counteracted by the robot. Using the concept of self-confidence as introduced in our framework can, therefore, be beneficial in such tasks as illustrated in Figure 4.10 (c). Here, the robot after a defined number of initial training episodes stops fully trusting the human and also rejects human suggestions, with an overtime increasing probability. This results in a learning curve that is even able to slightly outperform the learning without human input for both, action advice and feedback after action execution. However, it should be noted that here we assume the human accepts all rejections of action advice, which might not always be the case with real humans.

Real Human Input

We evaluated our approach with the same 20 inexperienced users on the sorting task. The subjects were only told that the task goal is to sort objects correctly into the boxes. However, they did not receive any information about a sorting criterion.



(a) (b) (c)

Figure 4.10.: We evaluate different forms of simulated human input on the sorting task, namely action advice, and feedback after action execution. To obtain a mean value of the performance after each episode, we averaged the policy learned up to that point over 20 evaluations runs and repeated this for 100 experiments with different random seeds. The plots show the mean (solid line) and standard deviation (shaded area) of the average reward over the number of episodes. We assume that since the human does not know the correct sorting criterion the most useful input they can give is about the task structure (first go to object, then pick up the object). The positive influence of such optimal useful human input on the learning is shown in (a) for action advice in the top row and feedback after action execution in the bottom row. However, the human might also give suboptimal input, e.g. on which box to choose, that we simulate here by random suggestions. (b) shows the negative influence of this on the learning for action advice (top row) and feedback (bottom row). Using the self-confidence module (c) the robot becomes able to also deal with such potentially incorrect input and the learning curves for action advice (first row) and feedback (bottom row) even slightly outperform the learning without human input when using an increasing self-confidence after an initial number of training episodes (bottom row).

All subjects interacted with the robot two times (once in Mode A and once in Mode B) for 20 episodes. In each episode, one object was supposed to be sorted by the robot and at the end of an episode, the human and robot received feedback on whether the sorting was correct. We randomized the order in which the subjects faced Mode A and B to eliminate ordering effects. Overall, the study process in the sorting task, including the experiment notes and questionnaires was identical to the one described in the kitchen task. Figure 4.11 shows the average rewards of the robot's policy over the number of episodes in the first experiment (a) and the second experiment (b). To obtain a mean value of the performance after each episode, we averaged the policy taught up to that point over 50 simulated evaluations runs for each subject for each episode. The plots are also separated between the modes that the subject used in each round and show the corresponding self-confidence of the robot in the bottom row. The results show that for all subjects our approach converged to the maximum average reward in less than 20 episodes.

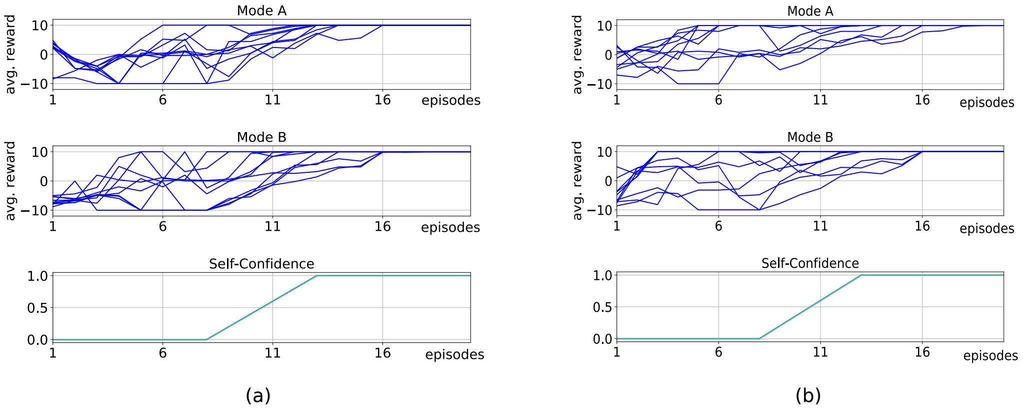
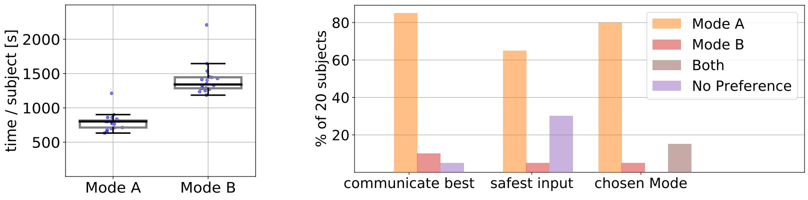


Figure 4.11.: We compare the two experiment rounds (a) and (b) and the two different interaction modes across the 20 inexperienced subjects on the robotic sorting task. In Mode A (top rows) subjects can prevent action execution and actively suggest own actions. In Mode B (middle rows) subjects can prevent action execution and give feedback after the action execution. We show the average reward over 10 episodes of interaction, where we plot the mean over 50 evaluation runs per episode for each subject. The Self-Confidence is shown in the bottom row.



(a) (b)

Figure 4.12.: (a) We found that the average interaction time in Mode A (Mdn=802.05) was on average lower than in Mode B (*Mdn* = 1340.96). A Wilcoxon signed-rank test indicated that this difference was significant, $T = 0$, $z = -3.92$, $p < .001$, $n = 20$. The plots show the median, interquartile range (box), 1.5 whiskers and the underlying data points. (b) In Questionnaire 2 the users answered subjective comparison questions about in which mode they felt they could communicate best with the robot, in which mode they felt most safe their input was useful and which mode they would choose for future interaction. In the last question, they could also choose if they prefer a combination of both modes.

Compared to the learning without human input, this is a speedup of approximately 25 percent. We investigate if human input can significantly decrease the number of episodes it takes to reach an average reward of 100 percent of the maximum reward. We compared learning without human input (*Mdn* = 16.5) to learning with human input for both Mode A (*Mdn* = 12) and Mode B (*Mdn* = 11). A Mann-Whitney-U test shows that the differences in comparison to learning without human input are significant in both, Mode A: $U = 332.5$, $z = 4.7$, $p < .001$, $n_1 = 20$, $n_2 = 100$ and Mode B: $U = 346.5$, $z = 4.6$, $p < .001$, $n_1 = 20$, $n_2 = 100$. However, Wilcoxon-signed rank test shows that there is no significant difference in learning speed between Mode A and Mode B, $T = 84.5$, $z = -0.42$, $p = .67$, $n = 19$. When comparing the results of the two rounds it shows that for some subjects the speed of learning improved in the second round, however, we think the sample size of 10 subjects is too small to perform meaningful statistical tests. When examining how the subjects gave input in the two rounds, we see that some of them changed their behavior and gave less input after they realized in the first round that they did not understand the sorting criterion. This is indicated by the average amount of input on the choice of the box subjects gave in Round 1 and Round 2. Figure 4.13 (a) shows that subjects reduced (potentially incorrect) input on which box to choose in Round 2 (*Mdn* = 21) compared to Round 1 (*Mdn* = 29). In addition, on average they gave more explicit 'I don't know' input for the choice of the box in Round 2 (*Mdn* = 11.5) compared to Round 1 (*Mdn* = 5). A Wilcoxon signed-rank test indicates that these differences are not significant, $T = 53$, $z = -1.94$, $p = .052$, $n = 20$ for input of boxes and $T = 16.5$, $z = -1.47$, $p = .141$, $n = 11$ for the 'I don't know' input. However, since the test reveals an almost significant difference between the amount of input on which box to pick from Round 1 to Round 2 this might show that there is a shift in the users' perception on the robot's abilities and how much they trust it to choose the correct box by itself.

We also noticed in the experiments that some users would not change their behavior even if they noticed they did not understand the sorting process.

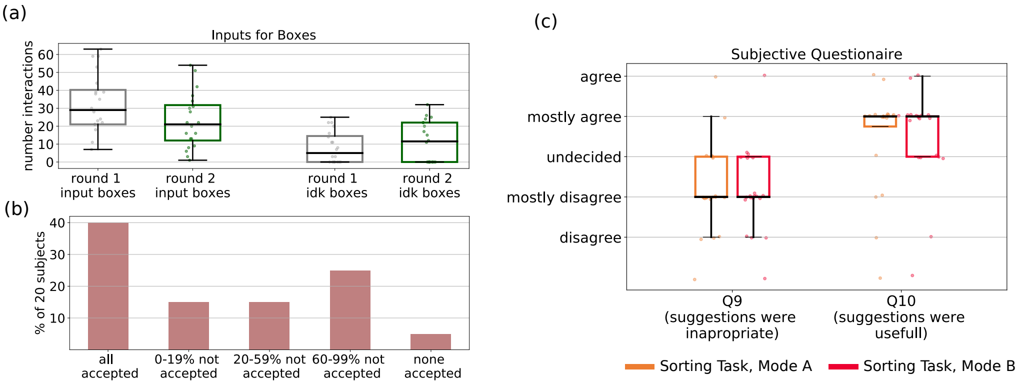


Figure 4.13.: (a) After the first round, most users adapted their behavior and the subjects on average gave less input on which box to choose after they realized they did not understand the sorting. They also gave on average more 'I don't know' input on the choice of the box. However, a Wilcoxon signed rank test indicates that these differences are not significant, $T = 53$, $z = -1.94$, $p = .052$, $n = 20$ for input of boxes and $T = 16.5$, $z = -1.47$, $p = .141$, $n = 11$ for the 'I don't know' input. (b) Not all subjects accepted the suggestions of the robot. Only 40 percent of the subjects accepted all suggestions and one subject even accepted none of the suggestions. (c) Most subjects agreed that the suggestions were mostly useful and on average the subjects mostly disagreed that the suggestions were inappropriate.

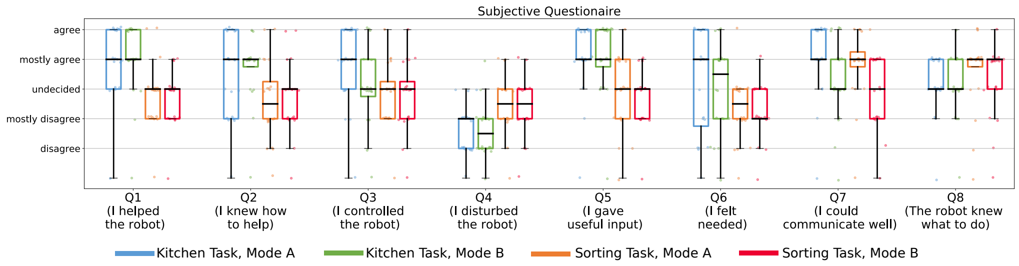


Figure 4.14.: After the different modes, the subjects answered subjective questionnaires in each task on a five-point Likert scale.

Figure 4.12 (a) shows the test results for total interaction times. As in the kitchen task, the total interaction time in Mode A (*Mdn* = 802.05) was on average lower than in Mode B (*Mdn* = 1340.96) and a Wilcoxon signed-rank test indicated that this difference was significant, $T = 0$, $z = -3.92$, $p < .001$, $n = 20$. Figure 4.12 (b) shows that similar to the kitchen task in the direct comparison in Questionnaire 2 most users (16 of 20) reported that they would prefer to use Mode A for future applications while one user would use Mode B and 3 users would prefer to use a combination of both modes.

We also evaluated how users reacted to suggestions and rejections of their input by the robot, that occurred once the robot's self-confidence started to rise. Usually, these suggestions started around episode 10. At this point, most users had already noticed that the sorting criterion was not obvious to them. However, Figure 4.13 (b) shows, that only 8 out of 20 users accepted all rejections and suggestions of the robot. Some users rejected a suggestion once or twice, to see if it would have been right and afterward started to trust new suggestions and accept them. However, 6 users refused more than 60 % of the robot's suggestions and one of them even rejected all of them. Figure 4.13 (c) visualizes the answers of the users to subjective questions on the robot suggestions. It shows that while on average most subjects thought the suggestions were mostly useful and appropriate, there were also subjects that perceived them as inappropriate and not useful. A Wilcoxon signed-rank test showed no significant difference between Mode A and Mode B for inappropriateness of the suggestions (Q9), $T = 20$, $z = -0.33$, $p = .739$, $n = 9$, and usefulness of the suggestions (Q10), $T = 22.5$, $z = 0$, $p = 1.0$, $n = 9$. In the experiment notes, subjects reported that the robot should have given them more reasons why it suggested certain actions and explain its decisions to the users. Subjects also reported that it would have helped them to ask the robot about its abilities, e.g. whether it can see colors or if it knows the boxes weights, to understand on which basis the robot made its suggestions. Another factor that might influence the subject's perception of the robot's suggestions is that once the self-confidence rises the robot also would start to explore actions not taken before, which could sometimes seem random to the users and might created distrust in action suggestions in general. Such rejections of the robot's suggestions can cause problems if the users were able to actively suggest own actions (e.g. in Mode A), because even if the robot learned the optimal policy humans would still interfere and cause incorrect sorting of the objects.

4.3.3. Discussion

In both described tasks human input in combination with our approach accelerated the learning process in almost all cases. For both tasks, users reported at the end, when directly asked in Questionnaire 2 that they preferred giving input in Mode A compared to Mode B. We see different reasons for this. The subjects reported in the subjective questionnaire (Questionnaire 1) and their experiment notes that they could communicate their own suggestions better in Mode A (Questionnaire 1-Q7, Wilcoxon signed-rank test, kitchen task: $T = 22$, $z = -2.45$, $p = .014$, $n = 16$; sorting task: $T = 0$, $z = -2.86$, $p = .004$, $n = 10$). Compared to the kitchen task, in the sorting task the subjects felt less helpful (Questionnaire 1-Q1, Wilcoxon signed-rank test, Mode A: $T = 19.5$, $z = -2.75$, $p = .006$, $n = 17$; Mode B: $T = 3$, $z = -3.33$, $p = .001$, $n = 15$) and less needed (Questionnaire 1-Q6, Wilcoxon signed-rank test, Mode A: $T = 20$, $z = -2.07$, $p = .038$, $n = 14$; Mode B: $T = 19.5$, $z = -2.37$, $p = .018$, $n = 15$). In general, Mode A was significantly less time-consuming (Wilcoxon signed-rank test, kitchen task: $T = 0$, $z = -3.92$, $p < .001$, $n = 20$; sorting task $T = 0$, $z = -3.92$, $p < .001$, $n = 20$. We also noticed that while action suggestion was clear to most users, the concept of feedback was harder to understand. Some users would start rating how actions were executed instead of rating which action was chosen, or judge based on other factors such as, how long it took the robot to suggest the correct action as the basis for their feedback. We believe that in the future more differentiated and clear ways for feedback would be beneficial. Figure 4.14 shows the results of the subjective questions on the two tasks and different modes. It shows that in the kitchen task, in particular, subjects considered their input to be useful and helpful and felt more needed than in the sorting task. In general, they felt they could communicate better and controlled the robot more in Mode A than in Mode B. However, one subject reported in the experiment notes that when actively suggesting actions the robot would not really ‘learn’ but only ‘replicate’. The mixed reactions of the users to suggestions of the robot or rejections of human input showed that the internal state of the robot should be more understandable to humans. It also showed that, in general, there is a difference in how users perceive the robot’s suggestions which ranged from ‘*Me and the robot are a real team now*’ to ‘*I feel, you want to force your opinion on me*’ and ‘*I want an option to tell the robot don’t do what you think but what I want*’ (quotes translated from German by authors). Also, the answers to the questionnaires show that humans felt less needed and less useful when the robot started making own suggestions in the sorting task. We think it would be important to communicate to the users, which forms of input (e.g. on task structure) are still beneficial in such a task. This way, they would potentially feel like they contribute more to the robot’s learning process. We consider these aspects of interactive learning very important for future research and applications to increase the acceptance and benefits of such interactive learning systems. We believe both tasks are easily reproducible on other robot systems and could be used as benchmark tasks for future experiments with interactive reinforcement learning systems.

4.4. Conclusion

In this chapter, we presented an approach to incorporate multiple human input channels in a reinforcement learning framework for sequential robotic tasks. Our approach also includes a concept for self-confidence such that the robot can try to reject human input after an initial training phase if it contradicts the learned policy. Experimental evaluations with 20 inexperienced users on two robotic tasks showed that human input could be incorporated beneficially to speed up the learning, even if it was partially incorrect. Direct comparison of different communication modes for the human subjects showed that most subjects preferred active participation in the learning process, e.g. through action suggestion or prohibition. However, the evaluations also showed that not all subjects would accept suggestions of the robot once the robot’s self-confidence was high enough to question the human input. This was particularly prominent when they did not understand the reasons behind the robot’s suggestions or the robot’s learning process. We think these results align well with findings from Li et al. (2016), who report that sharing metrics such as the robot’s uncertainty with users can increase engagement during learning and with Thomaz and Breazeal (2008) who also mention the importance of communicating the robot’s uncertainty to humans.

For future work, we think it is therefore important to include a more transparent communication of the robot's internal learning state into our approach. In particular, we want to investigate how communication about reasons for suggestions could help to increase acceptance of the robot's suggestions by users. The general question of how to deal with wrong human input in interactive RL systems requires further research as well. While the tasks evaluated in this paper provided already valuable insights on the interaction of humans with our interactive RL-framework, in future work we plan for additional evaluation on more realistic tasks, in particular in the context of assistant robotics, including more complex implementations for the human-advice and RL-module. To tackle more realistic and complex problems with larger state and action spaces we consider it necessary to change the current simplistic tabular representation of the advice and the RL module into more complex function approximators, with capabilities to generalize across similar states, such as a linear model of e.g. Radial-Basis-Function-features. Moreover, we think the human advice module should be extended in a way that it also can adapt to the learning process that might happen for the human during the interaction. E.g. if the human suggests an action and then gives negative feedback after realizing unexpected effects after execution the human advice module should take this correction into account. To this end, we additionally want to consider incorporation of a way to track how recent feedback or advice on nearby states has been received as proposed in the form of their eligibility module by **?**. Further, the current choice for decreasing learning and exploration rate of the RL-module should be reconsidered in future work, since even though it is a common choice in classical RL in the HRI context we see the need for adaptation to learning effects of the human and therefore the necessity to investigate different strategies of computation for learning and adaptation rates. Moreover, we consider it important to include more principled concepts for computation of the robot's self-confidence, which could also be state-dependent. One option would be to consider the convergence of the Q-function in different regions of the state space rather than just increase the self-confidence after an initial training phase as implemented for our experiments. Lastly, subjects also reported that they would have liked to communicate over different modalities besides the tablet with the robot e.g. natural language. Also incorporating rule-based input forms or options to teach sequences of coupled actions were suggested by the users and could be potentially incorporated in our approach in future work.

5. Conclusion

In this thesis, we introduced three main contributions towards the development of intelligent assistive robots. Hereby, the thesis contributes to the areas of incremental learning from demonstrations, online adaptation of Probabilistic Movement Primitives in shared workspaces and interactive Reinforcement Learning for sequential tasks. In this chapter, we briefly summarize the core points of those contributions. Subsequently, we discuss possible directions for future work that emerge out of, and around the topics of this thesis.

5.1. Summary

In this thesis, we focussed on three main contributions to the development of intelligent assistive robots. First, we presented a novel approach to incrementally learn a library for cooperative skills from demonstrations. In particular, we combine a Mixture of Experts approach and incremental Gaussian Mixture Models to incrementally learn a library of cooperative Interaction Primitives. We use Probabilistic Movement Primitives as a skill representation and thereby the skill library is able to capture variance in the demonstrations as well as inherent coupling between human and robot motions. We showed that our approach converges to similar solutions as existing approaches that learn a model from batch data. However, in contrast to those methods, our approach does not require all demonstrations during training time but can update the learned skill library online from new incoming data points. This is in particular desirable in a setting where the total number of tasks is unknown in the beginning, as it might be often the case for assistive robots. We evaluated the approach on a robotic scenario where a robot assists a human in a cooperative kitchen task. Here, we also evaluated how skill-libraries learned on subjects transfer to other subjects. The experimental evaluation also showed that not all skills transfer equally well and that further improvement of the gating model might be desirable to improve this performance in the future.

Second, we introduced two novel methods for online adaptation of Probabilistic

Movement Primitives in shared workspaces, namely spatial deformation, and temporal scaling. We additionally presented a goal-directed motion model that we used in combination with the two adaptation methods to achieve learning of intention aware adaptation of Probabilistic Movement Primitives to human subjects in a shared workspace setting. We evaluated the two adaptation methods with human subjects regarding subjective metrics such as perceived level of safety and comfort and objective metrics such as task performance and idle times on a robotic pick-and-place task. The results show that the subjective level of perceived safety was higher when the robot adapted its motions to the humans and humans felt less disturbed by the robot, in particular in the spatial adaptation mode. However, different adaptation strategies are not perceived equally by all subjects and subjects reported that is in particular important that the motions of the robot are predictable and understandable for them.

Third, we presented a framework for interactive Reinforcement Learning of sequential tasks. Here, we incorporate different input channels for humans such as feedback and action advice or prevention of action execution for the learning of sequencing of skills. We also introduced a concept of self-confidence that allows the robot to question and reject human advice if it contradicts the robot's policy after an initial training phase. We evaluated the proposed framework on two sequential robotic tasks. In the second task, we investigated human reactions after the robot's rejection of their input. The experimental evaluations showed, that while human input in both tasks was able to speed up the learning it is in particular important to provide better means of understanding and insights to the learning procedure and progress of the robot for humans in order to make them accept a robot's suggestions.

To summarize, this thesis contributed methods to enable assistive robots to incrementally learn a library for cooperative skills from demonstrations, adapt such learned skills to humans in shared workspaces, and a framework that includes multiple forms of human input to accelerate and improve Reinforcement Learning based sequencing of single skills. In all three parts of the thesis, our goal was always not only to contribute new theoretical methods but also to evaluate them on robotic tasks with real human interaction. We think such experimental evaluations, in particular with non-expert users are an important part to gain valuable insights on the interaction of human subjects with intelligent robotic systems and bring us closer towards the goal of intelligent and assistive robots. We believe insights gained throughout our experiments can hereby, also be useful for future research in the direction of interactive Machine Learning for robotics.

5.2. Future Work

Several directions for interesting future work emerge out of and around the topics of this thesis. In the following, we discuss what we consider most promising and important directions for follow up work on the main contributions of the thesis and what we consider in general important further research topics towards the development of intelligent assistive robots.

Skill Transfer Across Subjects, Tasks and Robots

In our experiments, we noticed that not all learned cooperative skills transferred equally well in between subjects. We think that one important point for future work on incremental learning of skill libraries is to investigate skill representations that allow for better transfer of learned skills in between subjects and maybe also in between different task settings. One important aspect is hereby how to represent the same skill consistently if the demonstrations are e.g. multimodal or if they differ significantly between subjects. We think a promising approach for this can be to investigate the use of dimensionality reduction methods and projections of demonstrations to lower-dimensional spaces or a hierarchical structure for skills that also considers multimodality. A long term goal in the development of skill libraries could also be the transfer of learned skills in between robots. In such a scenario robots should profit from skill databases learned across different households or application areas.

Preference-Based Adaptation

In our experiments, we noticed that the form of skill adaptation that humans prefer differs between subjects and might also differ in between different task settings. We believe that it is a promising direction to investigate how to classify different user types or situations and chose different adaptation strategies according to this classification. In particular, incorporating facial expressions or gaze directions as well as natural language processing to detect discomfort and the perceived level of safety of humans can be beneficial for this. Such multimodal cues could be subsequently used as objectives in a Reinforcement Learning setting so optimize adaptation strategies of the robot. We also think it would be beneficial to include more expressive means for the robots to communicate its adaptation strategies better to humans since one of the most crucial parts for acceptance in our experiments was how much the behavior of the robot was perceived as predictable by the humans. Hereby, including head motions or also spoken messages by the robot could be interesting aspects to combine with pure motion adaptation strategies.

Predicting Human Behavior and Intentions

One crucial aspect of the adaptation of learned skills as well as in the choice of skill activation in cooperative tasks is the correct prediction of human behavior and intentions. Hereby, more complex approaches than the currently used goal-directed motion can potentially be integrated well with our adaptation strategies for ProMPs. Additionally, we consider it important to develop new models that also consider the bidirectional influence between the robot's adaptation to human motions and resulting changes in human behavior. Here it can be beneficial to develop new approaches that model this influence inside a prediction model for human behavior. Beside pure prediction of human motions also the prediction of more high-level human intentions or different prediction models for varying situational contexts can be valuable for choosing better adaptation strategies for the robot and should be taken into account in future work.

Multimodal Communication and Interaction Channels

An important aspect to facilitate more natural interaction for skill learning but also in interactive Reinforcement Learning is to enable multimodal channels for communication, which was also mentioned by the participants in our experiments. In particular, in elderly assistance, this multimodality can become an important aspect since some elderly persons might have health limitations that limit the use of some senses such as vision or hearing and make alternatives necessary. Therefore, the integration of multimodal channels should further be investigated for skill learning as well as for skill improvement. Richer forms of demonstrations could be included in the skills and in the gating model of a cooperative skill library. For the learning of skills, we think it would be in particular beneficial to include information about interaction and manipulation forces or object relations. For the gating model on human intentions, we think multimodal information such as natural language commands or gaze direction can help to distinguish better between skills that have maybe similar gestures related to them. In the interactive Reinforcement Learning natural language as well as physical correction of robot motions and non-binary feedback, possibilities should be further investigated.

Explainability and Transparency of the Robot's Internal State

A major issue during our evaluations of the interactive Reinforcement Learning approach was that users wanted to understand how the robot came to its decisions and would refuse to trust the robot if this was not possible. To overcome such limitations for applications of interactive Reinforcement Learning we belief future work needs to focus on two points. The first is how to make the learning process accessible and understandable to humans. Therefore, exploration strategies, as well as the learning progress of the robot, should be communicated to the users. It is necessary to also consider how to include transparency about state-dependent uncertainty of the robot. We think that this can help to increase the trust of the users and is a basic component to enable more successful and acceptable interaction. On the algorithmic side, we believe that a more principled concept of how to compute and update the robot's self-confidence is necessary. Hereby, one possibility would be to include state-dependent convergence of the learned policy into the computation of the self-confidence. Another promising direction for future work can be how the robot could directly communicate which types of input would be most useful in different situations and actively request help depending on its uncertainty.

Intelligent Assistive Robots in the Wild

Besides the development and improvement of interactive Machine Learning algorithms, we believe a crucial part to bring us closer to the goal of intelligent assistive robots is evaluating the systems with humans in real application scenarios. In our experiments, we noticed that there are a lot of aspects in algorithmic and interface design as well as in the choice of communication channels that only become obvious or problematic during interaction with real human subjects. To understand the needs, preferences, and problems of users interacting with intelligent robots it is, therefore, crucial to further aim for research that does not only investigate interactions with robots in lab settings but aims for interactions in the wild, that is people's homes and workplaces. In such experimental evaluations, we also think it is important to have a variety of different participants in terms of age, gender and background experience with robots. Testing algorithms and intelligent robots in such more realistic contexts and evaluating how inexperienced users respond to different design and algorithmic choices can then contribute to bridge the gap between algorithmic development and its application for the use of humans and society.

A. Publication List

A.1. Journal Articles

Koert, D.; Kircher, M.; Salikutluk, V.; D'Eramo, C.; Peters, J. (2020). Multi-Channel Interactive Reinforcement Learning for Sequential Tasks. Frontiers in Robotics and AI, 7, 97

Koert, D.; Trick, S.; Ewerton, M.; Lutter, M.; Peters, J. (2020). Incremental Learning of an Open-Ended Collaborative Skill Library. International Journal of Humanoid Robotics, 17(01), 2050001

Koert, D.; Pajarinen, J.; Schotschneider, A.; Trick, S.; Rothkopf, C.; Peters, J. (2019). Learning Intention Aware Online Adaptation of Movement Primitives. IEEE Robotics and Automation Letters, 4(4), 3719-3726

Ewerton, M., Arenz, O., Maeda, G., **Koert, D.**, Kolev, Z., Takahashi, M., ; Peters, J. (2019). Learning Trajectory Distributions for Assisted Teleoperation and Path Planning. Frontiers in Robotics and AI, 6, 89

Maeda, G., Ewerton, M., **Koert, D.**, ; Peters, J. (2016). Acquiring and generalizing the embodiment mapping from human observations to robot skills. IEEE Robotics and Automation Letters, 1(2), 784-791

A.2. Articles in Conference Proceedings

Koert, D.; Trick, S.; Ewerton, M.; Lutter, M.; Peters, J. (2018). Online Learning of an Open-Ended Skill Library for Collaborative Tasks, Proceedings of the International Conference on Humanoid Robots (HUMANOIDS)

Koert, D.; Maeda, G.; Neumann, G.; Peters, J. (2018). Learning Coupled Forward-Inverse Models with Combined Prediction Errors, Proceedings of the International Conference on Robotics and Automation (ICRA)

Hoelscher, J.*; **Koert, D.***; Peters, J.; Pajarinen, J. (2018). Utilizing Human Feedback in POMDP Execution and Specification, Proceedings of the International Conference on Humanoid Robots (HUMANOIDS)

Koert, D.; Maeda, G.J.; Lioutikov, R.; Neumann, G.; Peters, J. (2016). Demonstration Based Trajectory Optimization for Generalizable Robot Motions, Proceedings of the International Conference on Humanoid Robots (HUMANOIDS)

Trick, S., **Koert, D.**, Peters, J., ; Rothkopf, C. (2019). Multimodal Uncertainty Reduction for Intention Recognition in Human-Robot Interaction, IEEE International Conference on Intelligent Robots and Systems (IROS)

Ewerton, M., Maeda, G., **Koert, D.**, Kolev, Z., Takahashi, M., ; Peters, J. (2019). Reinforcement learning of trajectory distributions: Applications in assisted teleoperation and motion planning. In IEEE International Conference on Intelligent Robots and Systems (IROS)

A.3. Workshop Papers

Koert, D., Hoelscher, J., Peters, J., and Pajarinen, J., Human-robot interaction under partial observability, IEEE ICRA Workshop WORKMATE 2018: the WORKplace is better with intelligent, collaborative, robot MATE", 2018.

B. Curiculum Vitae

Dorothea Carola Koert
Technische Universität Darmstadt
Hochschulstr. 10
64289 Darmstadt

Current Position

06/2016 - now	**Technische Universität Darmstadt** PhD student at the research group Intelligent Autonomous Systems

Education

10/2013 – 04/2016	**Technische Universität Darmstadt** M.Sc. Computational Engineering Major field of study: Computational Robotics
	Technische Universität Darmstadt M.Sc. Autonomous Systems
10/2010 – 10/2013	**Technische Universität Darmstadt** B.Sc. Computational Engineering, specialisation in Computational Mechanical and Process Engineering
07/2009 – 09/2010	**Voluntary Social Year at ambulance service** **“DRK Mittelhessen“** finished as certified paramedic

08/2001 – 06/2009	**Elisabethschule Marburg** University-entrance diploma (Abitur) Major subjects: Mathematics and Chemistry

Honors and Awards

12/2019	KI-Newcomer Award of German assosiation for Computer Science (Gesellschaft für Informatik)
10/2016	“Dr. Kanako Miura Award” of the IEEE-RAS International Conference on Humanoid Robots
2013, 2014, 2015	Winner of “Best in Class Autonomy”-award in the Robot RescueLeague (2013, 2014, 2015) and Robot Rescue League with Team Hector (2014)

Work Experience in Projects and Teaching

06/2018 – now	**KoBo34 (BMBF project)** Development of an assistive robot for the elderly
02/2017 - now	**Coordinator of the Softwaregroup of the aDDa 4 Students Team** Development of an autonomous car at TU Darmstadt
2019	**Teaching Assistant** “Statistical Machine Learning”
2018	**Teaching Assistant** “Projektseminar Computational Engineering”

Student Supervision

2019	Knaust, M., Master Thesis "Intuitive imitation learning for one-handed and bimanual tasks using ProMPs"
2019	Kircher, M., Master Thesis "Learning from Human Feedback: A Comparison of Interactive Reinforcement Learning Algorithms"
2019	Pal, S., Master Thesis "Deep Robot Reinforcement Learning for Assisting a Human"
2019	Jukonyte, L., Master Thesis "Probabilistic Motion and Intention Prediction for Autonomous Vehicles"
2019	Hoffmann, D., Master Thesis "Pedestrian Detection, Tracking and Intention Prediction in the Context of autonomous Driving"
2018	Trick, S., Master Thesis "Multimodal Uncertainty Reduction for Intention Recognition in a Human-Robot Environment"
2018	Bug,D., Master Thesis “Comparison and Evaluation of Concepts for SLAM in the Context of Autonomous Driving”
2017	Hoelscher, J., Master Thesis, "Interactive Planning Under Uncertainty"
2019	Kirschner, M., Bachelor Thesis "Integration of LIDAR SLAM for an automous vehicle"
2019	Lang, M., Bachelor Thesis "Imitation Learning for Highlevel Robot behavior in the Context of Elderly Assistance"
2019	Divo, F., Bachelor Thesis "Trajectory Based Upper Body Gesture Recognition for an Assistive Robot"
2019	Kaiser, F., Bachelor Thesis "Towards a Robot Skill Library Using Hierarchy, Composition and Adaptation"
2018	Schotschneider, A., Bachelor Thesis "Collision Avoidance in Uncertain Environments for Autonomous Vehicles using POMDPs"
2017	Szelag, S., Bachelor Thesis "Transferring Insights on Mental Training to Robot Motor Skill Learning"
2017	Rother, D., Bachelor Thesis "Transferring Insights on Biological Sleep to Robot Motor Skill Learning"

Bibliography

Abel, D., Salvatier, J., Stuhlmüller, A., and Evans, O. (2017). Agent-agnostic human-in-the-loop reinforcement learning. *arXiv preprint arXiv:1701.04079*.

Ahmed, A. and Xing, E. (2008). Dynamic non-parametric mixture models and the recurrent chinese restaurant process: with applications to evolutionary clustering. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 219–230. SIAM.

Akgun, B. and Subramanian, K. (2011). Robot learning from demonstration: kinesthetic teaching vs. teleoperation. *Unpublished manuscript*.

Alenljung, B., Lindblom, J., Andreasson, R., and Ziemke, T. (2017). User experience in social human-robot interaction. *International Journal of Ambient Computing and Intelligence (IJACI)*, 8(2):12–31.

Allmendinger, J. and Ebner, C. (2006). Arbeitsmarkt und demografischer wandel: Die zukunft der beschäftigung in deutschland. *Zeitschrift für Arbeits-und Organisationspsychologie A&O*, 50(4):227–239.

Amershi, S., Cakmak, M., Knox, W. B., and Kulesza, T. (2014). Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120.

Amor, H. B., Neumann, G., Kamthe, S., Kroemer, O., and Peters, J. (2014). Interaction primitives for human-robot cooperation tasks. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2831–2837. IEEE.

Amor, H. B., Vogt, D., Ewerton, M., Berger, E., Jung, B., and Peters, J. (2013). Learning responsive robot behavior by imitation. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3257–3264. IEEE.

Arai, T., Kato, R., and Fujita, M. (2010). Assessment of operator stress induced by robot collaboration in assembly. *CIRP annals*, 59(1):5–8.

Arandjelovic, O. and Cipolla, R. (2006). Incremental learning of temporally-coherent gaussian mixture models. *Society of Manufacturing Engineers (SME) Technical Papers*, pages 1–1.

Argall, B. D., Browning, B., and Veloso, M. (2008). Learning robot motion control with demonstration and advice-operators. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 399–404. IEEE.

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483.

Atkeson, C. G. and Schaal, S. (1997). Robot learning from demonstration. In *ICML*, volume 97, pages 12–20. Citeseer.

Bai, H., Cai, S., Ye, N., Hsu, D., and Lee, W. S. (2015). Intention-aware online POMDP planning for autonomous driving in a crowd. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 454–460.

Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). Robot programming by demonstration. In *Springer handbook of robotics*, pages 1371–1394. Springer.

Blumberg, B., Downie, M., Ivanov, Y., Berlin, M., Johnson, M. P., and Tomlinson, B. (2002). Integrated learning for interactive synthetic characters. In *ACM transactions on graphics (TOG)*, volume 21, pages 417–426. ACM.

Bundesagentur für Arbeit, D. (2019). Arbeitsmarktsituation im pflegebereich. berichte: Blickpunkt arbeitsmarkt.

Busch, B., Maeda, G., Mollard, Y., Demangeat, M., and Lopes, M. (2017). Postural optimization for an ergonomic human-robot interaction. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 2778–2785. IEEE.

Calinon, S. and Billard, A. (2007). 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. ACM.

Calinon, S., Guenter, F., and Billard, A. (2007). On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2):286–298.

Celemin, C., Maeda, G., Ruiz-del Solar, J., Peters, J., and Kober, J. (2019). Reinforcement learning of motor skills using policy search and human corrective advice. *The International Journal of Robotics Research*, 38(14):1560–1580.

Celemin, C. and Ruiz-del Solar, J. (2016). Teaching agents with corrective human feedback for challenging problems. In *2016 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–6. IEEE.

Celemin, C. and Ruiz-del Solar, J. (2019). An interactive framework for learning continuous actions policies based on corrective feedback. *Journal of Intelligent & Robotic Systems*, 95(1):77–97.

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

Colomé, A. and Torras, C. (2017). free contextualized probabilistic movement primitives, further enhanced with obstacle avoidance. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference*, pages 3190–3195. IEEE.

Cruz, F., Magg, S., Nagai, Y., and Wermter, S. (2018). Improving interactive reinforcement learning: What makes a good teacher? *Connection Science*, 30(3):306–325.

Cruz, F., Magg, S., Weber, C., and Wermter, S. (2016). Training agents with interactive reinforcement learning and contextual affordances. *IEEE Transactions on Cognitive and Developmental Systems*, 8(4):271–284.

Declercq, A. and Piater, J. H. (2008). Online learning of gaussian mixture models-a two-level approach. In *VISAPP (1)*, pages 605–611.

Dermy, O., Paraschos, A., Ewerton, M., Peters, J., Charpillet, F., and Ivaldi, S. (2017). Prediction of intention during interaction with icub with probabilistic movement primitives. *Frontiers in Robotics and AI*, 4:45.

Doblhammer, G. and Ziegler, U. (2006). Future elderly living conditions in europe: demographic insights. In *Gender, Health and Ageing*, pages 267–292. Springer.

Dragan, A. D., Lee, K. C., and Srinivasa, S. S. (2013). Legibility and predictability of robot motion. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, pages 301–308. IEEE.

Engel, P. M. and Heinen, M. R. (2010). Incremental learning of multivariate gaussian mixture models. In *Brazilian Symposium on Artificial Intelligence*, pages 82–91. Springer.

Ewerton, M., Neumann, G., Lioutikov, R., Amor, H. B., Peters, J., and Maeda, G. (2015). Learning multiple collaborative tasks with a mixture of interaction primitives. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1535–1542. IEEE.

Griffith, S., Subramanian, K., Scholz, J., Isbell, C. L., and Thomaz, A. L. (2013). Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in neural information processing systems*, pages 2625–2633.

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

Hämmerer, D. and Eppinger, B. (2012). Dopaminergic and prefrontal contributions to reward-based learning and outcome monitoring during child development and aging. *Developmental psychology*, 48(3):862.

Havoutis, I., Tanwani, A. K., and Calinon, S. (2016). Online incremental learning of manipulation tasks for semi-autonomous teleoperation.

Hayes, B. and Scassellati, B. (2013). Challenges in shared-environment human-robot collaboration. *learning*, 8(9).

Hayne, R., Luo, R., and Berenson, D. (2016). Considering avoidance and consistency in motion planning for human-robot manipulation in a shared workspace. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 3948–3954. IEEE.

Hoffmann, H., Pastor, P., Park, D.-H., and Schaal, S. (2009). Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2587–2592. IEEE.

Hoyos, J., Prieto, F., Alenya, G., and Torras, C. (2016). Incremental learning of skills in a task-parameterized gaussian mixture model. *Journal of Intelligent & Robotic Systems*, 82(1):81–99.

Huang, Y., Rozo, L., Silvério, J., and Caldwell, D. G. (2019). Kernelized movement primitives. *The International Journal of Robotics Research*, 38(7):833–852.

Huber, M., Su, Y.-H., Krüger, M., Faschian, K., Glasauer, S., and Hermsdörfer, J. (2014). Adjustments of speed and path when avoiding collisions with another pedestrian. *PloS one*, 9(2):e89589.

Hussein, A., Gaber, M. M., Elyan, E., and Jayne, C. (2017). Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):21.

Ijspeert, A. J., Nakanishi, J., Hoffmann, H., Pastor, P., and Schaal, S. (2013). Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373.

Ijspeert, A. J., Nakanishi, J., and Schaal, S. (2002). Movement imitation with nonlinear dynamical systems in humanoid robots. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 2, pages 1398–1403. IEEE.

Ijspeert, A. J., Nakanishi, J., and Schaal, S. (2003). Learning attractor landscapes for learning motor primitives. In *Advances in neural information processing systems*, pages 1547–1554.

Ikeda, T., Chigodo, Y., Rea, D., Zanlungo, F., Shiomi, M., and Kanda, T. (2013). Modeling and Prediction of Pedestrian Behavior based on the Sub-goal Concept. *Robotics*, 10.

Inamura, T., Kojo, N., and Inaba, M. (2006). Situation recognition and behavior induction based on geometric symbol representation of multimodal sensorimotor patterns. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5147–5152. IEEE.

Isbell, C. L., Kearns, M., Singh, S., Shelton, C. R., Stone, P., and Kormann, D. (2006). Cobot in lambdamoo: An adaptive social statistics agent. *Autonomous Agents and Multi-Agent Systems*, 13(3):327–354.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Jenkins, O. C. and Mataric, M. J. (2002). Deriving action and behavior primitives from human motion data. In *IEEE/RSJ international conference on intelligent robots and systems*, volume 3, pages 2551–2556. IEEE.

Judah, K., Roy, S., Fern, A., and Dietterich, T. G. (2010). Reinforcement learning via practice and critique advice. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Kaochar, T., Peralta, R. T., Morrison, C. T., Fasel, I. R., Walsh, T. J., and Cohen, P. R. (2011). Towards understanding how humans teach robots. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 347–352. Springer.

Kaplan, F., Oudeyer, P.-Y., Kubinyi, E., and Miklósi, A. (2002). Robotic clicker training. *Robotics and Autonomous Systems*, 38(3-4):197–206.

Karamouzas, I. and Overmars, M. (2010). Simulating human collision avoidance using a velocity-based approach.

Khansari-Zadeh, S. M. and Billard, A. (2012). A dynamical system approach to realtime obstacle avoidance. *Autonomous Robots*, 32(4):433–454.

Knox, W. B. and Stone, P. (2008). Tamer: Training an agent manually via evaluative reinforcement. In *2008 7th IEEE International Conference on Development and Learning*, pages 292–297. IEEE.

Knox, W. B. and Stone, P. (2010). Combining manual feedback with subsequent mdp reward signals for reinforcement learning. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 5–12. International Foundation for Autonomous Agents and Multiagent Systems.

Knox, W. B. and Stone, P. (2011). Augmenting reinforcement learning with human feedback. In *ICML 2011 Workshop on New Developments in Imitation Learning (July 2011)*, volume 855, page 3.

Knox, W. B. and Stone, P. (2012). Reinforcement learning from simultaneous human and mdp reward. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 475–482. International Foundation for Autonomous Agents and Multiagent Systems.

Knox, W. B., Stone, P., and Breazeal, C. (2013). Training a robot via human feedback: A case study. In *International Conference on Social Robotics*, pages 460–470. Springer.

Kober, J., Bagnell, J. A., and Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274.

Koert, D., Kircher, M., Salikutluk, V., D'Eramo, C., and Peters, J. (2020). Multi-channel interactive reinforcement learning for sequential tasks. *Frontiers in Robotics and AI*, 7:97.

Koert, D., Maeda, G., Lioutikov, R., Neumann, G., and Peters, J. (2016). Demonstration based trajectory optimization for generalizable robot motions. In *Humanoids*, pages 515–522.

Koert, D., Pajarinen, J., Schotschneider, A., Trick, S., Rothkopf, C., and Peters, J. (2019a). Learning intention aware online adaptation of movement primitives. *IEEE Robotics and Automation Letters*, 4(4):3719–3726.

Koert, D., Trick, S., Ewerton, M., Lutter, M., and Peters, J. (2018). Online learning of an open-ended skill library for collaborative tasks. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–9. IEEE.

Koert, D., Trick, S., Ewerton, M., Lutter, M., and Peters, J. (2019b). Incremental learning of an open-ended collaborative skill library. *International Journal of Humanoid Robotics*.

Konidaris, G., Kuindersma, S., Grupen, R., and Barto, A. (2012). Robot learning from demonstration by constructing skill trees. *The International Journal of Robotics Research*, 31(3):360–375.

Koppenborg, M., Nickel, P., Naber, B., Lungfiel, A., and Huelke, M. (2017). Effects of movement speed and predictability in human–robot collaboration. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 27(4):197–209.

Kormushev, P., Calinon, S., and Caldwell, D. (2013). Reinforcement learning in robotics: Applications and real-world challenges. *Robotics*, 2(3):122–148.

Kruse, T., Basili, P., Glasauer, S., and Kirsch, A. (2012). Legible robot navigation in the proximity of moving humans. In *Advanced Robotics and its Social Impacts (ARSO), 2012 IEEE Workshop on*, pages 83–88. IEEE.

Kuhlmann, G., Stone, P., Mooney, R., and Shavlik, J. (2004). Guiding a reinforcement learner with natural language advice: Initial results in robocup soccer. In *The AAAI-2004 workshop on supervisory control of learning and adaptive systems*. San Jose, CA.

Kulić, D. and Croft, E. A. (2005). Safe planning for human-robot interaction. *Journal of Robotic Systems*, 22(7):383–396.

Kulić, D., Ott, C., Lee, D., Ishikawa, J., and Nakamura, Y. (2012). Incremental learning of full body motion primitives and their sequencing through human motion observation. *The International Journal of Robotics Research*, 31(3):330–345.

Kulvicius, T., Ning, K., Tamosiunaite, M., and Worgötter, F. (2011). Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting. *IEEE Transactions on Robotics*, 28(1):145–157.

Kuniyoshi, T. (1989). Teaching by showing: Generating robot programs by visual observation of human performance. In *Proc. 20th Int. Symp. Ind. robots, 1989*, pages 119–126.

Lasota, P. A. and Shah, J. A. (2015). Analyzing the effects of human-aware motion planning on close-proximity human–robot collaboration. *Human factors*, 57(1):21–33.

Lee, D., Ott, C., and Nakamura, Y. (2010). Mimetic communication model with compliant physical contact in human-humanoid interaction. *The International Journal of Robotics Research*, 29(13):1684–1704.

Lemme, A., Reinhart, R. F., and Steil, J. J. (2014). Self-supervised bootstrapping of a movement primitive library from complex trajectories. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 726–732. IEEE.

Levas, A. and Selfridge, M. (1984). A user-friendly high-level robot teaching system. In *Proceedings. 1984 IEEE International Conference on Robotics and Automation*, volume 1, pages 413–416. IEEE.

Li, G., Whiteson, S., Knox, W. B., and Hung, H. (2016). Using informative behavior to increase engagement while learning from human reward. *Autonomous agents and multi-agent systems*, 30(5):826–848.

Linz, K. and Stula, S. (2010). Demographic change in europe-an overview. *Observatory for Sociopolitical Developements in Europe*, 4(1):2–10.

Loftin, R., Peng, B., MacGlashan, J., Littman, M. L., Taylor, M. E., Huang, J., and Roberts, D. L. (2014). Learning something from nothing: Leveraging implicit human feedback strategies. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pages 607–612. IEEE.

Lopez, F. G., Abbenseth, J., Henkel, C., and Dörr, S. (2017). A predictive online path planning and optimization approach for cooperative mobile service robot navigation in industrial applications. In *Mobile Robots (ECMR), 2017 European Conference on*, pages 1–6. IEEE.

Losey, D. P. and O'Malley, M. K. (2018). Trajectory deformations from physical human–robot interaction. *IEEE Transactions on Robotics*, 34(1):126–138.

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

Luo, R., Hayne, R., and Berenson, D. (2018). Unsupervised early prediction of human reaching for human–robot collaboration in shared workspaces. *Autonomous Robots*, 42(3):631–648.

Maclin, R. and Shavlik, J. W. (1996). Creating advice-taking reinforcement learners. *Machine Learning*, 22(1-3):251–281.

Maeda, G., Ewerton, M., Lioutikov, R., Amor, H. B., Peters, J., and Neumann, G. (2014). Learning interaction for collaborative tasks with probabilistic movement primitives. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 527–534. IEEE.

Maeda, G., Ewerton, M., Osa, T., Busch, B., and Peters, J. (2017a). Active incremental learning of robot movement primitives. In *Conference on Robot Learning (CORL)*.

Maeda, G. J., Neumann, G., Ewerton, M., Lioutikov, R., Kroemer, O., and Peters, J. (2017b). Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks. *Autonomous Robots*, 41(3):593–612.

Maeda, G. J., Neumann, G., Ewerton, M., Lioutikov, R., Kroemer, O., and Peters, J. (2017c). Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks. *Autonomous Robots*, 41(3):593–612.

Mainprice, J. and Berenson, D. (2013). Human-robot collaborative manipulation planning using early prediction of human motion. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference*, pages 299–306. IEEE.

Mainprice, J., Hayne, R., and Berenson, D. (2015). Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative replanning. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 885–892. IEEE.

Meltzoff, A. N. and Moore, M. K. (1977). Imitation of facial and manual gestures by human neonates. *Science*, 198(4312):75–78.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.

Morato, C., Kaipa, K. N., Zhao, B., and Gupta, S. K. (2014). Toward safe human robot collaboration by using multiple kinects based real-time human tracking. *Journal of Computing and Information Science in Engineering*, 14(1):011006.

Moreno, D. L., Regueiro, C. V., Iglesias, R., and Barro, S. (2004). Using prior knowledge to improve reinforcement learning in mobile robotics. *Proc. Towards Autonomous Robotics Systems. Univ. of Essex, UK*.

Muelling, K., Kober, J., and Peters, J. (2010). Learning table tennis with a mixture of motor primitives. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 411–416. IEEE.

Muench, S., Kreuziger, J., Kaiser, M., and Dillman, R. (1994). Robot programming by demonstration (rpd)-using machine learning and user interaction methods for the development of easy and comfortable robot programming systems. In *Proceedings of the International Symposium on Industrial Robots*, volume 25, pages 685–685. INTERNATIONAL FEDERATION OF ROBOTICS, & ROBOTIC INDUSTRIES.

Nahavandi, S. (2019). Industry 5.0—a human-centric solution. *Sustainability*, 11(16):4371.

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

Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., Peters, J., et al. (2018). An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179.

Østergaard, E. H. (2018). Welcome to industry 5.0.

Paraschos, A., Daniel, C., Peters, J., and Neumann, G. (2018). Using probabilistic movement primitives in robotics. *Autonomous Robots*, 42(3):529–551.

Park, D.-H., Hoffmann, H., Pastor, P., and Schaal, S. (2008). Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 91–98. IEEE.

Pastor, P., Hoffmann, H., Asfour, T., and Schaal, S. (2009). Learning and generalization of motor skills by learning from demonstration. In *2009 IEEE International Conference on Robotics and Automation*, pages 763–768. IEEE.

Pavlichenko, D. and Behnke, S. (2017). Efficient stochastic multicriteria arm trajectory optimization. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 4018–4025. IEEE.

Pérez-D'Arpino, C. and Shah, J. A. (2015). Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6175–6182. IEEE.

Pinto, R. C. and Engel, P. M. (2015). A fast incremental gaussian mixture model. *PloS one*, 10(10):e0139931.

Ravichandar, H. and Dani, A. (2015). Human intention inference through interacting multiple model filtering. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2015 IEEE International Conference on*, pages 220–225. IEEE.

Rosenbaum, D. A. (2009). *Human motor control*. Academic press.

Saveriano, M., Hirt, F., and Lee, D. (2017). Human-aware motion reshaping using dynamical systems. *Pattern Recognition Letters*, 99:96–104.

Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242.

Sisbot, E. A. and Alami, R. (2012). A human-aware manipulation planner. *IEEE Transactions on Robotics*, 28(5):1045–1057.

Sisbot, E. A., Marin-Urias, L. F., Broquere, X., Sidobre, D., and Alami, R. (2010). Synthesizing robot motions adapted to human presence. *International Journal of Social Robotics*, 2(3):329–343.

Suay, H. B. and Chernova, S. (2011). Effect of human guidance and state space size on interactive reinforcement learning. In *2011 Ro-Man*, pages 1–6. IEEE.

Sutton, R. S. and Barto, A. G. (2011). Reinforcement learning: An introduction.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S., Barto, A. G., et al. (1998). *Introduction to reinforcement learning*, volume 2. MIT press Cambridge.

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

Thomaz, A. L., Breazeal, C., et al. (2006). Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *Aaai*, volume 6, pages 1000–1005. Boston, MA.

Thomaz, A. L., Hoffman, G., and Breazeal, C. (2005). Real-time interactive reinforcement learning for robots. In *AAAI 2005 workshop on human comprehensible machine learning*.

Torrey, L. and Taylor, M. (2013). Teaching on a budget: Agents advising agents in reinforcement learning. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1053–1060. International Foundation for Autonomous Agents and Multiagent Systems.

Torrey, L. and Taylor, M. E. (2012). Help an agent out: Student/teacher learning in sequential decision tasks. In *Proceedings of the Adaptive and Learning Agents workshop (at AAMAS-12)*.

Trautman, P. and Krause, A. (2010). Unfreezing the robot: Navigation in dense, interacting crowds. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 797–803. IEEE.

Tsarouchi, P., Makris, S., and Chryssolouris, G. (2016). Human–robot interaction review and challenges on task planning and programming. *International Journal of Computer Integrated Manufacturing*, 29(8):916–931.

Unhelkar, V. V., Lasota, P. A., Tyroller, Q., Buhai, R.-D., Marceau, L., Deml, B., and Shah, J. A. (2018). Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time. *IEEE Robotics and Automation Letters*, 3(3):2394–2401.

Vasic, M. and Billard, A. (2013). Safety issues in human-robot interactions. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 197–204. IEEE.

Vogt, D., Lorenz, B., Grehl, S., and Jung, B. (2015). Behavior generation for interactive virtual humans using context-dependent interaction meshes and automated constraint extraction. *Computer Animation and Virtual Worlds*, (3-4):227–235.

Vogt, D., Stepputtis, S., Grehl, S., Jung, B., and Amor, H. B. (2017). A system for learning continuous human-robot interactions from human-human demonstrations. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2882–2889. IEEE.

Wang, Y., Huber, M., Papudesi, V. N., and Cook, D. J. (2003). User-guided reinforcement learning of robot assistive tasks for an intelligent environment. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 1, pages 424–429. IEEE.

Weiss, A., Igelsbock, J., Calinon, S., Billard, A., and Tscheligi, M. (2009). Teaching a humanoid: A user study on learning by demonstration with hoap-3. In *RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication*, pages 147–152. IEEE.

Wiewiora, E., Cottrell, G. W., and Elkan, C. (2003). Principled methods for advising reinforcement learning agents. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 792–799.

Zoliner, R., Pardowitz, M., Knoop, S., and Dillmann, R. (2005). Towards cognitive robots: Building hierarchical task representations of manipulations from human demonstration. In *Proceedings of the 2005 IEEE International Conference On Robotics and Automation*, pages 1535–1540. IEEE.