Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

Teaching Robots With Interactive Reinforcement Learning

Dissertation

Dissertation submitted to the University of Hamburg
with the aim of achieving a doctoral degree at the
Faculty of Mathematics, Informatics and Natural Sciences,
Department of Informatics.

Francisco Cruz

Hamburg 2017

Submitted:
June 2nd, 2017

Day of oral defence:
July 10th, 2017

The following evaluators recommend the admission of the dissertation:

Prof. Dr. Víctor Uc-Cetina (reviewer)
Department of Informatics,
University of Hamburg, Germany

Prof. Dr. Frank Steinicke (chair)
Department of Informatics,
University of Hamburg, Germany

Prof. Dr. Stefan Wermter (advisor)
Department of Informatics,
University of Hamburg, Germany

To Manuel, Celia, Alejandra, Mari, and Nahuel.
You are my best reward.

Abstract

Intelligent assistive robots have recently taken their first steps toward entering domestic scenarios. It is thus expected that they perform tasks which are often considered rather simple for humans. However, for a robot to reach human-like performance diverse subtasks need to be accomplished in order to satisfactorily complete a given task. These subtasks include perception, understanding of the environment, learning strategies, knowledge representation, awareness of its own state, and manipulation of the environment.

An open challenging issue is the time required by a robot to autonomously learn a new task. A strategy to speed up this apprenticeship period for autonomous robots is the integration of parent-like trainers to scaffold the learning. In this regard, a trainer guides the robot to enhance the task performance in the same manner as caregivers may support infants in the accomplishment of a given task. In this thesis, we focus on these learning approaches, specifically on interactive reinforcement learning to perform a domestic task. We use parent-like advice to explore two set-ups: agent-agent and human-agent interaction.

First, we investigate agent-agent interactive reinforcement learning. We use an artificial agent as a parent-like trainer. The artificial agent is previously trained by autonomous reinforcement learning and afterward becomes the trainer of other agents. This interactive scenario allows us to experiment with the interplay of parameters like the probability of receiving feedback and the consistency of feedback. We show that the consistency of feedback deserves special attention since small variations on this parameter may considerably affect the learner's performance. Moreover, we introduce the concept of contextual affordances which allows to reduce the state-action space by avoiding failed-states, i.e., to avoid a group of states from which it is not possible to reach the goal-state of a task. By avoiding

failed-states, the learner-agent is able to collect significantly more reward. The experiments also focus on the internal representation of knowledge in trainer-agents to improve the understanding of what the properties of a good teacher are. We show that using a polymath agent, i.e., an agent with more distributed knowledge among the states, it is possible to offer better advice to learner-agents compared to specialized agents.

Thereafter, we study human-agent interactive reinforcement learning. Initially, experiments are performed with human parent-like advice using uni-modal speech guidance. The experimental set-up considers the use of different auditory sensors to compare how they affect the consistency of advice and the learning performance. We observe that an impoverished speech recognition system may still help interactive reinforcement learning agents, although not to the same extent as in the ideal case of agent-agent interaction. Afterward, we perform an experiment including audiovisual parent-like advice. The set-up takes into account the integration of multi-modal cues in order to combine them into a single piece of consistent advice for the learner-agent. Additionally, we utilize contextual affordances to modulate the advice given to the robot to avoid failed-states and to effectively speed up the learning process. Multi-modal feedback produces more confident levels of advice allowing learner-agents to benefit from this in order to obtain more reward and to gain it faster.

This thesis contributes to knowledge in terms of studying the interplay of multi-modal interactive feedback and contextual affordances. Overall, we investigate which parameters influence the interactive reinforcement learning process and show that the apprenticeship of reinforcement learning agents can be sped up by means of interactive parent-like advice, multi-modal feedback, and affordances-driven environmental models.

Zusammenfassung

Intelligente Assistenzroboter werden vermehrt in häuslichen Umgebungen eingesetzt, wo sie entsprechende Aufgaben übernehmen, die für Menschen einfach umzusetzen sind. Um eine ähnliche Performanz mit einem Roboter zu erreichen, ist es häufig nötig, Teilaufgaben zu definieren. Diese beinhalten die Perzeption, sowie das Wissen und Verstehen der Umwelt, Lernstrategien, Wissensrepräsentationen, das Bewusstsein über den eigenen Zustand und Handlungsmöglichkeiten in der jeweiligen Umgebung.

Das Erlernen autonomen Handelns hinsichtlich einer speziellen Aufgabe durch einen Roboter ist bis heute ein nicht vollständig gelöstes Problem. Eine mögliche unterstützende Lernstrategie für autonome Roboter ist das Bereitstellen eines sogenannten "Lehrers" oder "Trainers", dessen Rolle es ist, den Roboter in der Ausführung einer Aufgabe anzuleiten, ähnlich wie Eltern ihren Kindern beim Erlernen von Fähigkeiten helfen. In der vorliegenden Dissertation konzentrieren wir uns daher auf genau solche Lernszenarien, insbesondere auf das interaktive, verstärkende Lernen ("interactive reinforcement learning", IRL) zur Ausführung von häuslichen Aufgaben. Wir verwenden das o.g. Lehrerprinzip zur Untersuchung von zwei Fallstudien: die Agenten-Agenten-Interaktion und Mensch-Agenten-Interaktion.

Als Erstes untersuchen wir die Agenten-Agenten-Interaktion mit der verstärkenden Lernstrategie (IRL). Ein künstlicher Agent dient dabei als Lehrer, welcher zuvor mit der "reinforcement"-Methode trainiert wurde um autonome Aufgaben erfüllen zu können. Dieses Wissen wird dann auf den anderen Agenten übertragen. Diese Art der Interaktion erlaubt die Untersuchung des Zusammenspiels von Parametern wie z.B. der Wahrscheinlichkeit ein Feedback zu erhalten oder dessen Zuverlässigkeit. Wir zeigen, dass die Beständigkeit bzw. Zulässigkeit von Feedback

eine entscheidende Rolle spielt, da schon kleine Variationen Einfluss auf die Lernperformanz haben. Wir führen außerdem das Konzept von kontextuellen Affordanzen ein, die es erlauben den Zustands-Aktions-Raum durch das Vermeiden von sogenannten “failed states” zu minimieren. Dies sind Zustände von denen aus es unmöglich ist, weitere sinnvolle Handlungen zu generieren. Diese Reduktion des Aktionsraumes hat einen signifikant positiven Einfluss auf die verwendete Lernmethode für den Lehrer. Unsere Experimente konzentrieren sich auch auf die internen Repräsentationen des Agenten um ein verbessertes Verständnis über die wichtigen Eigenschaften eines guten Lehrers zu gewinnen. Wir zeigen, dass das Einsetzen eines sogenannten “polymath”-Agenten, d.h. ein Agent mit verteiltem Wissen über seinen Zustandsraum, zu einer Verbesserung von Hinweisen in Lernszenarien spezialisierter Agenten führt.

Desweiteren erforschen wir die IRL Strategie für die Mensch-Agenten Interaktion. Die Experimente beinhalten das Erteilen von Ratschlägen, wie es für uns Menschen üblich ist, wobei uni-modale Sprachsignale verwendet werden. Der experimentelle Aufbau enthält verschiedene auditive Sensoren, um deren Effekt auf die Zuverlässigkeit der erteilten Hinweise im Hinblick auf die Lernperformanz zu vergleichen. Unsere Beobachtungen haben dabei gezeigt, dass schon ein einfaches Spracherkennungssystem ein IRL-Szenario unterstützen kann, allerdings nicht im selben Umfang wie im idealen Fall der Agenten-Agenten-Interaktion. Darauf aufbauend zeigen wir Experimente, die audio-visuelle Hinweise verwenden. Das Szenario beschreibt die Integration von multi-modalen Stimuli zur Bereitstellung konsistenter Ratschläge für den lernenden Agenten. Wir verwenden außerdem kontextuelle Affordanzen zur Modulierung von Hinweisen für den Roboter, was zur Vermeidung von genannten “failed states” führt und damit zur Beschleunigung des Lernverfahrens. Das multi-modale Feedback führt zu einer höheren Konfidenz gegebener Ratschläge, was dafür sorgt, dass der lernende Agent seine Belohnung erhöhen und diese schneller erhalten kann.

Diese Arbeit leistet einen Beitrag zum Wissen über das Zusammenspiel zwischen multi-modalem interaktivem Feedback und kontextuellen Affordanzen. Zusammengefaßt untersuchen wir den Einfluss von Parametern im IRL und zeigen, dass das Erlernen von Fähigkeiten autonomer Agenten durch interaktives Handeln, multi-modales Feedback und mit Hilfe von durch Affordanzen beschriebenen Umgebungsmodellen erheblich verbessert werden kann.

Contents

List of Figures

List of Tables

Part I

Preamble and Basics

Chapter 1

Introduction

1.1 Motivation

There has been considerable progress in robotics in the last years allowing robots to successfully contribute to our society. We can find them from industrial contexts, where they are well established, to domestic environments, where their presence is steadily rising. A reasonable concern is then: How well prepared are assistive robots to be social actors in daily-life home environments in the near future.

Big challenges in robotics involve to work with service and assistive robots in home environments and develop plausible robot domestic applications. The underlying intention is the development of highly interactive intelligent robots to perform tasks in new and complex environments while being able to anticipate and resolve conflictual situations that may lead to mistakes or incomplete performance.

Intelligent robots operating around people should be able to know where they are located, detect users, learn and recognize faces, learn new objects, understand action-object opportunities, and furthermore, they should learn to behave cooperative in domestic scenarios. In order to accomplish these complex tasks successfully, robots have to deal with many challenges such as perception, pattern recognition, navigation, and object manipulation, all of that in varying environmental conditions. Such challenges can only be addressed if the robot constantly acquires and learns new skills, either autonomously or from parent-like trainers.

This thesis principally targets bio-inspired developmental learning and psycholog-

ically motivated learning approaches within the context of home applications for a domestic cleaning scenario. These methods are inspired by how humans develop knowledge through interactions with their environment.

1.2 Problem Statement and Research Questions

Reinforcement Learning (RL) is a learning approach supported by behavioral psychology where an agent, e.g., an infant or a robot, interacts with its environment trying to find an optimal policy to perform a particular task. In every time step, the agent performs an action reaching a new state and, sometimes, may obtain either a reward or a punishment. The agent tries to maximize the obtained reward by choosing the best action in a given state (Sutton and Barto, 1998).

One RL problem, that still remains open, is the time spent by an RL agent during learning. It often requires excessive time to find a proper policy (Knox and Stone, 2009), mainly due to a large and complex state action space which leads to excessive computational costs. To overcome this issue, sometimes an RL agent may be guided by a trainer in order to help the agent to finish the task more rapidly, like parents assisting their children. In this regard, when interacting with their caregivers, infants are subject to different environmental stimuli which can be present in various modalities. Nevertheless, when more modalities are considered, issues can also emerge regarding the interpretation and integration of multi-modal information, especially when multiple sources are conflicting or being ambiguous, e.g., yielding low confidence levels (Ozasa et al., 2012). As a consequence, the advice to follow may not be clear and may be misunderstood, and hence, may lead the apprentice agent to a decreased performance when solving a task (Cruz et al., 2016a).

In this thesis, we explore approaches aiming to speed up the RL method, such as interactive feedback using both agent-agent and human-agent interaction, complemented by the use of contextual affordances, which are a generalization of the affordance concept (Gibson, 1979), as a way to model possible actions in the environment. Therefore, the main research question can be stated as: *Can RL be sped up by using parent-like advice and affordance-driven environmental models?* A subset of supplementary research questions arise in order to answer the main one

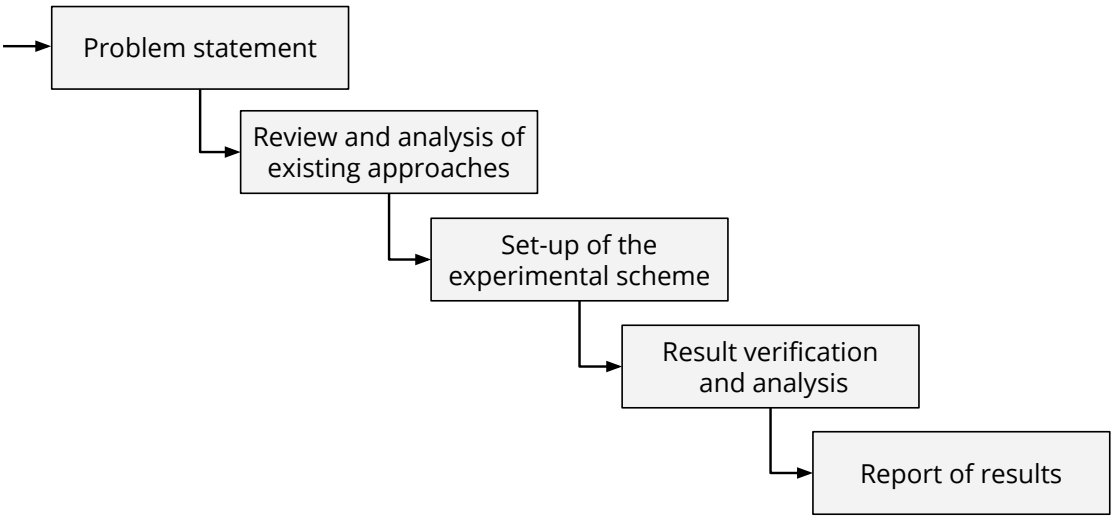


Figure 1.1: Five steps carried out into the scientific method.

and to obtain a better understanding of interactive reinforcement learning (IRL):

- How can an affordance-based model of the environment support the IRL framework?
- What constitutes a good teacher-agent when considering internal knowledge representation and interaction parameters?
- How beneficial is uni- and multi-modal advice during the apprenticeship process?

These questions will be addressed in this document one by one with the aim of answering the main research question. In the context of a robot learning a new task with an advisor suggesting actions in order to complete the task successfully, we hypothesize that a concrete range of advice level is needed to obtain a good performance by the robot. The advice level is measured in terms of the probability of feedback and the robot performance in terms of the collected reward and number of actions to finish the task.

1.3 Research Methodology

The presented research can be divided into five main steps based on the scientific method (see Fig. 1.1), as described by Nola and Sankey (2014):

- Problem statement: As stated in the previous section, RL requires excessive time to find a proper policy. Moreover, by using IRL, if more modalities are considered, issues can also emerge regarding the interpretation and integration of multi-modal information, especially when multiple sources are in conflict or ambiguous.
- Review and analysis of existing approaches: A comprehensive review of the theoretical framework and recent research has been carried out. Since this was a four-year research project, new approaches have emerged during the time that this thesis has been developed in and have also been surveyed. As a result, a detailed overview of useful approaches and their biological and psychological representations has been obtained describing all methods which are used in our project.
- Set up the experimental scheme: The methods found in the previous step have been integrated into a common robotic scenario, including parent-like advice to speed up the acquisition of the knowledge on how to perform a domestic task. In this regard, different kinds of parent-like trainers have been used to evaluate the learner-agent performance.
- Results verification and analysis: The results on achieving the goal of completing the domestic task have been evaluated systematically. To this end, the collected rewards of different learner-agents have been used to assess the convergence point and speed of convergence.
- Report results: All the obtained results have been reported through different scientific publications in high-impact conferences and journals. Additionally, this thesis itself also represents a way to report the final obtained results. In terms of code, all the routines developed during this research project are available in a git repository. For further details, refer to https://git.informatik.uni-hamburg.de/cruz/IRL.

1.4 Novelty and Contribution of the Work

This work presents methods, experimental set-ups, and novel results on interactive reinforcement learning. The main contribution to the state of the art of IRL can

be summarized in the following points:

- **Study of interaction parameters.** Learning is dissimilarly affected when trainers with different interaction characteristics are used. We study the probability of receiving feedback, consistency of feedback, and learner-agent's obedience. The consistency of feedback deserves special attention, given that even very few mistakes in the advice given by trainers may lead to a considerably worse learning process.

- **Investigation of impact of different internal representations on IRL.** We contribute to a better understanding of the impact of different internal representations of the knowledge on the performance of IRL. Results suggest that using polymath agents (agents with more distributed knowledge among the states) as trainers benefits the learning process leading to greater collected reward and faster convergence in comparison to specialized agents.

- **Extension toward contextual affordances.** The classic idea of affordances relates objects, actions, and effects. We have introduced the concept of contextual affordances to model the actions in the environment taking into consideration an additional variable for the state of the agent leading to a more accurate representation of affordances.

- **Interplay of interactive feedback and contextual affordances.** By using IRL along with contextual affordances, learners take advantage of parent-like trainer knowledge and a better understanding of the environment. Thus, the learner is able to collect a greater reward and for this converges more rapidly. Both approaches have not been utilized altogether in the RL framework.

- **Analysis of effects of uni- and multi-modal advice on IRL.** Results show that multi-modal stimuli benefit learners using RL in comparison to uni-modal signals. Moreover, multi-modal advice modulated by contextual affordances enables to collect greater and faster reward in comparison to autonomous RL and non-affordances IRL.

Finally, from a more general view, the main contribution of this work is to show that learning of RL agents can be sped up by using parent-like advice, multi-modal feedback, and affordance-driven environmental models. All the aforementioned ap-

proaches help individually, but, the combined use of them leads to greater benefits on the performance of IRL. All these results are described and explained during this thesis by means of different experimental set-ups.

1.5 Structure of the Thesis

The present document is organized into four main parts, each one of them is described as follows:

I. Preamble and Basics: After a brief introduction to the problem and the way to address it, we present the state of the art and a robotic scenario which will be utilized in the course of this work.

   1. Introduction: This is the current chapter which briefly describes what motivates this thesis, states the problem along with defining the main research questions, and shows the methodology utilized to address the problem. It also presents a brief description of the main novelties and contributions.

   2. Theoretical Framework and Related Approaches: The state of the art is presented from four different perspectives, all of them related and used during the development of the work. Initially, we present the RL framework and its components as well as the learning techniques utilized to solve Markovian decision processes. Subsequently, we show the main elements of artificial neural networks, including learning and training methods. Consecutively, we present the affordance concept from the classic perspective to the current use in robotics and agent control. Finally, we survey the main methods in IRL in autonomous agents showing the main problems of the classic RL approach and defining different kinds of IRL.

   3. Robotic Cleaning-table Scenario: This chapter defines a domestic scenario for a robotic agent. The scenario consists of a robot standing in front of a table with the aim of cleaning it. The proposed scenario is described as a Markovian decision process, and actions, states, transitions, and a reward function are defined. The scenario description is

an important section in this document since all proposed methods are assessed throughout this scenario. The task is initially learned by an agent autonomously and afterward, a second agent learns the same task assisted by an external trainer, either artificial or human.

II. Agent-Agent Interactive Reinforcement Learning: The second part of the document presents a general proof of concept for the proposed methods in the sense of an artificial agent trained autonomously to after becoming itself into a parent-like trainer. An artificial trainer-agent enables to better control some experimental variables as well as repeat the apprenticeship process more quickly. Moreover, it presents the basis to subsequently introduce a human parent-like trainer in the next part.

4. Interactive Feedback and Contextual Affordances: It is introduced the concept of contextual affordance to model the actions in the environment. This is linked with the first research question: *How can an affordance-based model of the environment support the IRL framework?* Contextual affordances are implemented by an artificial neural network and then combined with IRL using an artificial parent-like trainer. Furthermore, we allow a decreasing frequency of feedback over time in order to mimic human-agent interactive scenarios. Our results show that IRL using affordances benefits the learner-agent performance in terms of collected reward and executed actions on each episode.

5. Influence of Different Trainer Types on Learner-Agents: This chapter is directly related to the second research question: *What constitutes a good teacher-agent?* We investigate what characteristics are relevant for an agent to become a good teacher. To this end, the frequency of feedback and the consistency of feedback, as well as the learner-agent's obedience are analyzed. The obtained results show that even using a polymath trainer-agent with a low probability of feedback and high consistency of feedback as an advisor, a learner-agent may learn in few episodes.

III. Human-Agent Interactive Reinforcement Learning: In the third part of the thesis, the IRL is presented as an approach using human parent-like trainers this time, at first with uni-modal auditory guidance only and then with multi-modal audiovisual feedback. In this regard, this part of the document

shifts our approach closer to naturalistic scenarios, considering multi-modal stimuli complemented by an affordance-driven approach later.

6. Speech Guidance Using a Domain-specific Language: We show the IRL approach working with human parent-like trainers. To deliver instructions or guidance we use an automatic speech recognition system through different kinds of microphones in order to evaluate how the hardware configuration affects the speech recognition and consequently the guidance for a learner-agent. We also perform experiments with environmental noise created by keeping an arbitrary distance from the input sensors. Our results show that the speech-driven IRL approach improves the learner-agent performance in terms of the performed actions over each episode.

7. Multi-modal Feedback Using Audiovisual Sensory Inputs: We extend the speech-driven IRL approach in order to incorporate multi-modal guidance which is related to the third and last posed research question: *How beneficial is uni- and multi-modal advice during the apprenticeship process?* We use audiovisual feedback identifying the advice associated with the sensory input incorporating a confidence value. When using multi-modal signals, it is necessary to deal with inconsistencies of the inputs, therefore, we propose a mathematical transformation to relate the likeness level considering congruent and incongruent sensory inputs. Afterward, we complement this multi-modal integration model with an affordance-driven approach to modulate the advice sent to the learner-agent. Our best results are obtained by using multi-modal information with contextual affordances during the apprenticeship process.

IV. Closing: The fourth and last part of the document presents the final conclusions as well as appendices with additional material which is related to this research but not directly utilized to address the posed research questions.

8. Conclusions: In this chapter, we summarize the main ideas, insights, and methods described throughout the thesis. After analyzing the obtained results, we develop the main conclusions and the contributed knowledge to the state of the art in IRL. Moreover, this chapter discusses the open issues, describes limitations of the proposed model, and

gives the main directions in order to address future improvements.

A. Contextual Affordances with an Associative Neural Architecture: This first appendix shows an alternative method to learn contextual affordances using an associative neural network. The robotic scenario is based on the aforementioned domestic scenario with slight adjustments. The obtained results show that the self-organized architecture is able to learn the contextual affordances in the proposed scenario rapidly by mapping the network inputs into a complex-domain output.

B. State Transitions of the Cleaning-table Scenario: The second appendix shows more details of the search space in the robotic cleaning-table task. States and transitions are shown by means of nodes and edges respectively in a state machine.

C. Published Contributions Originating from this Thesis: This appendix lists the scientific publications produced during the research for the present thesis. Publications include journal articles, conference, and workshops papers.

Chapter 2

Theoretical Framework and Related Approaches

2.1 Reinforcement Learning and Interaction

Learning is the process of acquiring knowledge, abilities, behavior, or principles through study, experience, or education, and as such, it is one of the foundations of intelligence, either human or artificial (Russell and Norvig, 1995). To use an approach that includes learning is appropriate when full knowledge of the environment is not available at the moment of designing a solution (Mitchell, 1997). It is by learning that systems are provided with autonomy.

Reinforcement Learning (RL) (Sutton and Barto, 1998) is a kind of learning that allows autonomous agents to learn using feedback received from the environment (Szepesvári, 2010; Busoniu et al., 2010; Rieser and Lemon, 2011). The basic idea is inspired by nature itself, based on the manner that people and animals learn (Niv, 2009). RL is based on trying actions and observing what happens in the environment. If actions lead to better situations, there is the tendency of applying such behavior again, otherwise, the tendency is to avoid such behavior in the future. Therefore, the problem is reduced to learn how to select optimal actions to be performed in each situation to reach a given goal (Rieser and Lemon, 2011).

Aims may be expressed by a function (of reward) which assigns a numerical value to each action performed by the agent from a particular situation. Positive values

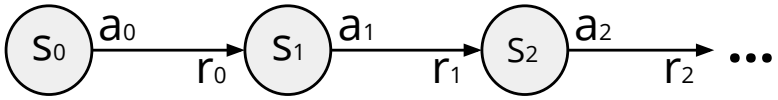


Figure 2.1: An RL agent must associate what actions to select in each state in order to maximize the collected reward.

indicate to the agent that the just performed action is good and negative values indicate a bad action (Mitchell, 1997). Moreover, each performed action leads the agent to a variation of the current state.

RL implies to acquire new knowledge to improve the performance of an agent interacting with its environment. However, the agent is not told what actions to take. The agent has to discover by itself what actions lead to more reward by trial and error (Marsland, 2015). Hence, the agent has to associate situations (or states) with actions which maximize:

$$r_0 + \lambda \cdot r_1 + \lambda^2 \cdot r_2 + ... \tag{2.1}$$

where r_i is the reward in episode i and $\lambda \in [0, 1)$ the discount factor, a parameter that indicates how influential future actions are. Fig. 2.1 depicts such a situation for the three first episodes.

2.1.1 First Insights

One of the first ideas which are related to RL is what Aristotle called the contiguity law. The philosopher expressed his idea as "things that occur near each other in time or space are readily associated". The contiguity law is one of the laws of association proposed by Aristotle around the year 350 B.C. (Warren, 1916).

One other important idea for the conception of RL is the classic conditioning also known as Pavlovian conditioning or stimulus-response learning (Pavlov, 1927). Pavlov observed that when putting food in front of dogs, they started to salivate, but also observed a similar response to other stimuli as seeing the person who brought the food. Therefore, he experimented by ringing a bell each time he fed the dogs. Afterward, Pavlov rang the bell without feeding the dogs. The dogs

started to salivate regardless of the presence of food. Thus, dogs were giving a response (salivation) to a stimulus (the bell). Learning by conditioning is based on stimulus-response rules, which means that Pavlov's dogs made no decisions, they simply salivated because the ring of the bell reminded them of the food.

If we take into consideration that taken actions have consequences, the learning is not only through stimulus-response associations anymore. This is known as instrumental or operational conditioning (Thorndike, 1911). Thorndike examined cats trying to escape from a box. The needed time to get out was monitored as the learning metric and showed a decreasing learning curve. With his experiments, Thorndike was able to establish that animals cannot only learn stimulus-response relations, but also arbitrary behavior based on such stimuli.

Later on, Rescorla and Wagner (1972) introduced the error-driven learning principle, i.e., the update of an association value is proportional to the difference between the prediction and observed values. Let s_t be a state and $V(s_t)$ the association value in the state s_t at time t, then we may call s_{t+1} the next state and $V(s_{t+1})$ the predicted value associated to the next state. The update of the predicted value can be described as:

$$V'(s_t) \leftarrow V(s_t) + \alpha[V(s_{t+1}) - V(s_t)] \tag{2.2}$$

with α being a small positive value called learning rate, $V(s_{t+1}) - V(s_t)$ the prediction error, and $V'(s_t)$ the updated association value in the state s_t at time t. Eq. (2.2) constitutes an example of a temporal-difference learning method given that the update is done based on the difference $V(s_{t+1}) - V(s_t)$ corresponding to two estimations at different time steps.

2.1.2 Elements of Reinforcement Learning

Additionally to the agent itself and the environment, four main elements in RL tasks can be identified (Sutton and Barto, 1998; Rieser and Lemon, 2011):

- The control policy.
- The reward function.
- The value function.

- Optionally, a model of the environment.

Each of these elements will be explained in the following subsections.

2.1.2.1 Policy

The control policy defines the way the agent behaves at every moment. It is a correspondence between the state the agent is in and the actions that can be taken in such a state. Moreover, it resembles the stimulus-response association from psychology.

In some occasions, the policy may be a function or a table, in other occasions more complex approaches are necessary, such as artificial neural networks (Szepesvári, 2010). The policy is the core of RL in the sense that it is enough to determine the agent's behavior.

2.1.2.2 Reward Function

The reward function defines the objective of an RL problem. It establishes a correspondence between each state of the environment (or state-action pair) and a value which indicates the desirability of every state. The only aim of an agent during the learning process is to maximize the overall received reward. In other words, the reward function defines what events are good or bad for the agent in terms of the aim, being the only way to indicate it (Mitchell, 1997).

In biological systems, the reward may be related to pleasure and pain which are also associated with the level of the dopamine neurotransmitter in the brain (Niv, 2009). Obviously, the function is external to the agent and therefore it cannot be modified by it.

2.1.2.3 Value Function

Alternatively, to the policy, the agent may also learn a function which indicates how good each state is with respect to the aim, the so-called value function. On the one hand, the reward function says what it is good in an immediate sense, on

the other hand, the value function indicates what it is good for the whole task execution (Busoniu et al., 2010).

The value of a state is the total amount of reward that an agent can expect to accumulate in the future starting from that state. Rewards are given by the environment whereas values must be estimated from sequences of observations that an agent accumulates through the operation.

2.1.2.4 Model of the Environment

The model is something that imitates or mimics the behavior of the environment. For instance, given a state and an action, the model might predict the result of the next state and the next reward. Models are used to plan since the action to perform can be decided considering possible future situations before they have actually occurred.

One way to model the environment is by the use of affordances (Jamone et al., 2017; Min et al., 2016). In fact, affordances allow anticipating the effect of an action which is performed by the agent in the environment. This model will be presented further in Sec. 2.2 since it represents a fundamental part of the performed experiments.

2.1.3 The Reinforcement Learning Framework

RL is a learning method which allows an apprentice agent to learn from interactions with the environment to reach an aim. The interaction is continuous, namely, the agent selects actions and the environment responds to these actions presenting new situations to the agent. Furthermore, the environment sends numerical rewards that the agent attempts to maximize over time (Russell and Norvig, 1995).

At each instant t, the apprentice agent receives some representation of the state of the environment $s_t \in S$, where S is the set of possible states. In that state s_t, the agent selects an action $a_t \in A(s_t)$, where $A(s_t)$ is the set of available actions in s_t. Afterward, as consequence of the performed action, the agent receives a numeric reward $r_{t+1} \in \mathbb{R}$ and transits to a new state s_{t+1} (Sutton and Barto, 1998). Fig. 2.2 shows the classic RL framework where an agent in s_t performs an action a_t

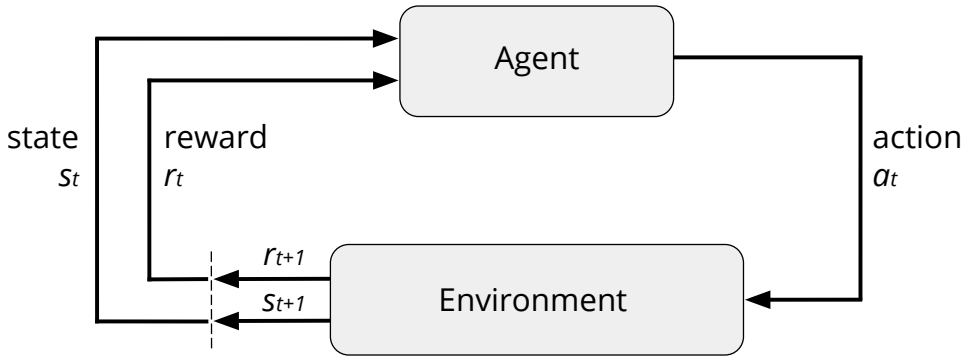


Figure 2.2: The classic reinforcement learning loop between the agent and the environment. Figure adapted from (Sutton and Barto, 1998).

in the environment which takes the agent to a new state s_{t+1} besides obtaining a reward r_{t+1}.

Each time, the agent updates the association between states and selection probabilities of every possible action. This association is named policy and denoted by π with $\pi_t(s_t, a_t)$ being the probability of performing action a_t in state s_t. RL methods specify how the agent should change the policy as a result of its experience. Basically, the problem is to approximate a function $\pi : S \rightarrow A$ where S is the set of states and A the set of actions. The agent aims to maximize the amount of total reward obtained during the execution.

2.1.4 Markov Decision Processes

Markov Decision Processes (MDPs) are the base of RL tasks. In an MDP, transitions and rewards depend only on the current state and the selected action by the agent (Puterman, 1994). In other words, a Markov state contains all the information related to the dynamics of a task, i.e., once the current state is known, the history of transitions that led the agent to that position is irrelevant in terms of the decision-making problem.

An MDP is characterized by the 4-tuple $< S, A, \delta, r >$ where:

- S is a finite set of states,
- A is a set of actions,

- δ is the transition function $\delta : S \times A \rightarrow S$, and,
- r is the reward function $r : S \times A \rightarrow \mathbb{R}$.

As aforementioned, at each time t, the agent perceives the current state $s_t \in S$ and selects the action $a_t \in A$ to perform it. The environment returns the reward $r_t = r(s_t, a_t)$ and the agent transits to the state $s_{t+1} = \delta(s_t, a_t)$. The functions r and δ depend only on the current state and action, i.e., it is a process with no memory.

To formalize the problem we should consider that the agent wants to learn the policy $\pi : S \rightarrow A$ which, from a state s_t, produces the greatest accumulated reward over time (Rieser and Lemon, 2011). Therefore, we can extend Eq. (2.2) as follows:

$$r_t + \gamma \cdot r_{t+1} + \gamma^2 \cdot r_{t+2} + ... = \sum_{i=0}^{\infty} \lambda^i \cdot r_{t+1} = V^{\pi}(s_t) \tag{2.3}$$

where $V^{\pi}(s_t)$ is the accumulated reward by following the policy π from an initial state s_t and λ is a constant ($0 \leq \lambda < 1$) which determines the relative importance of immediate rewards with respect to the future rewards. If $\lambda = 0$, then the agent is short-sighted and maximizes only the immediate rewards. If $\lambda \rightarrow 1$ the agent is more foresighted and takes more the future rewards into account.

2.1.5 Action Selection Methods

An agent choosing actions usually has to deal with the exploration/exploitation trade-off problem, that is, the available information depends on the previously performed actions and as such the agent has to explore the action space offsetting the already explored good actions with others that it never tried (Marsland, 2015).

The agent needs a strategy to choose actions to perform in a given state. In the following, we review different alternatives to implement such action selection strategies.

2.1.5.1 Greedy Method

The greedy method selects always the action a which reports the greatest value from a state s_t. However, it is risky because by exploiting good actions, identified at the beginning of the learning process, the agent could get stuck in local minima and not consider potentially better actions (Szepesvári, 2010). Formally, the probability $P(s_t, a)$ of selecting an action a in a state s_t is defined as follows:

$$P(s_t, a) = \begin{cases} 1 & \text{if } a = \underset{a_i \in A(s_t)}{argmax}\ Q(s_t, a_i) \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

2.1.5.2 ϵ-Greedy Method

The ϵ-greedy method explores more in comparison to a greedy policy. To achieve this, it utilizes an exploration factor ϵ, randomly chosen from a uniform distribution. Thus, the probability $P(s_t, a)$ of selection action a in state s_t can be formally defined as:

$$P(s_t, a) = \begin{cases} 1 - \epsilon & \text{if } a = \underset{a_i \in A(s_t)}{argmax}\ Q(s_t, a_i) \\ \epsilon & \text{otherwise} \end{cases} \tag{2.5}$$

However, a drawback of this method is that if $Q(s, a_1) >> Q(s, a_2)$ then actions a_1 and a_2 have the same probability of being chosen at the moment of exploration (Szepesvári, 2010). When comparing the greedy strategy with ϵ-greedy strategies, it is observed that the greedy strategy may quickly get stuck in a local minimum while the ϵ-greedy strategies in general converge to greater reward (Sutton and Barto, 1998).

2.1.5.3 Softmax Method

The softmax method uses a parameter T (so-called temperature) to determine the level of exploration. On the one hand, if $T \to \infty$, then all the available actions are equally likely. On the other hand, if $T \to 0$, then the softmax method becomes

greedy (Szepesvári, 2010). The probability $P(s_t, a)$ of selecting action a from state s_t is formally defined as follows:

$$P(s_t, a) = \frac{e^{Q(s_t,a)/T}}{\sum_{a_i \in A} e^{Q(s_t,a_i)/T}} \tag{2.6}$$

Generally, T is reduced over time to benefit the convergence. Nevertheless, not always it is easy to define T because it depends on the order of magnitude of $Q(s, a)$. Moreover, it is difficult to state whether ϵ-greedy or softmax performs better since this may depend on other task-related factors and the set parameters (Sutton and Barto, 1998).

2.1.6 Temporal-Difference Learning

Actions are selected according to a policy π, which in psychology is called a set of stimulus-response rules or associations Kornblum et al. (1990). Thus, the value of taking an action a in a state s under a policy π is denoted $q^\pi(s, a)$ which is also called the action-value function for a policy π.

In essence, to solve an RL problem means to find a policy that collects the highest reward possible over the long run (Mitchell, 1997). If there exists at least one policy which is better or equal than all others this is called an optimal policy. Optimal policies are denoted by π^* and share the same optimal action-value function which is denoted by q^* and defined as:

$$q^*(s, a) = \max_\pi q^\pi(s, a) \tag{2.7}$$

This optimal action-value function can be solved through the Bellman optimality equation for q^* as follows:

$$q^*(s, a) = \sum_{s'} p(s'|s, a)[r(s, a, s') + \gamma \max_{a'} q^*(s', a')] \tag{2.8}$$

where s is the current state, a is the taken action, s' is the next state reached by performing action a in the state s, and a' are possible actions that could be taken in s'. In the equation, p represents the probability of reaching the state s' given

that the current state is s and the selected action is a, and r is the received reward for performing action a in the state s to reach the state s'.

For solving Eq. (2.8) diverse learning methods exist. Algorithm 2.1 shows a general learning method with an iterative update of $Q(s, a)$ based on temporal-difference learning (Busoniu et al., 2010). Following, we revise two of these iterative methods.

Algorithm 2.1. General algorithm of temporal-difference learning.

```
1: Initialize Q(s, a) arbitrarily
2: for (each episode) do
3:     Choose an action a_t
4:     repeat
5:         Take action a_t
6:         Observe reward r_{t+1} and next state s_{t+1}
7:         Choose an action a_{t+1}
8:         Update Q(s_t, a_t)
9:         s_t ← s_{t+1}
10:        a_t ← a_{t+1}
11:    until s is terminal
12: end for
```

2.1.6.1 On-policy Method SARSA

In the SARSA method, the update of $Q(s_t, a_t)$ depends on the 5-tuple $< s_t, a_t, r_t, s_{t+1}, a_{t+1} >$, which gave rise to the name SARSA (state, action, reward, state, action). SARSA is an on-policy algorithm because it learns and follows the action selection policy (based on the values $Q(s_t, a_t)$) at the same time. Furthermore, the value $Q(s_t, a_t)$ is updated using the value $Q(s_{t+1}, a_{t+1})$ of the next action a_{t+1} that the agent will perform in the next iteration (Mitchell, 1997).

The on-policy method SARSA solves the Eq. (2.8) considering transitions from state-action pair to state-action pair instead of transitions from state to state only (Rummery and Niranjan, 1994). Every state-action value can be updated using the following Eq. (2.9):

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{2.9}$$

2.1.6.2 Off-policy Method Q-learning

The Q-learning method is an off-policy algorithm because it learns the action selection policy independently of the actions performed by the agent. The update of the value $Q(s_t, a_t)$ is carried out utilizing the value $\max_{a \in A(s_{t+1})} Q(s_{t+1}, a)$, although the agent might perform a different action in the next iteration (Mitchell, 1997). Therefore, state-action values are updated according to the Eq. (2.10) (Watkins, 1989; Watkins and Dayan, 1992):

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_{a \in A(s_{t+1})} Q(s_{t+1}, a) - Q(s_t, a_t)] \tag{2.10}$$

Sutton and Barto (1998) carried out a task-oriented comparison between SARSA and Q-learning. They used a grid world called cliff walking. The task consisted of reaching a goal position, going through intermediate states receiving a negative reward of -1 for each. In case that the agent stepped into a forbidden region (the so-called cliff), the agent received a negative reward of -100 and it must restart the task. Results showed the RL agent using SARSA to learn the longer but safer path, keeping itself away from the cliff, while the RL agent using Q-learning learned the shorter and riskier path.

2.1.7 Learning and Behavior

To autonomously explore the environment is one of the first developing behaviors for a human. An infant is constantly exploring its surroundings and learning from it most of the time without the need of a trainer to instruct it on how to perform a task.

Learning in humans and animals has been widely studied in neuroscience yielding a better understanding of how the brain can acquire new cognitive skills. We currently know that RL is associated with cognitive memory and decision-making in animals' and humans' brains in terms of how behavior is generated (Niv, 2009). Fig. 2.3 shows how the brain interacts with the world and processes the sensory inputs to generate motor actions. In general, computational neuroscience has interpreted data and used abstract and formal theories to help to understand about functions in the brain.

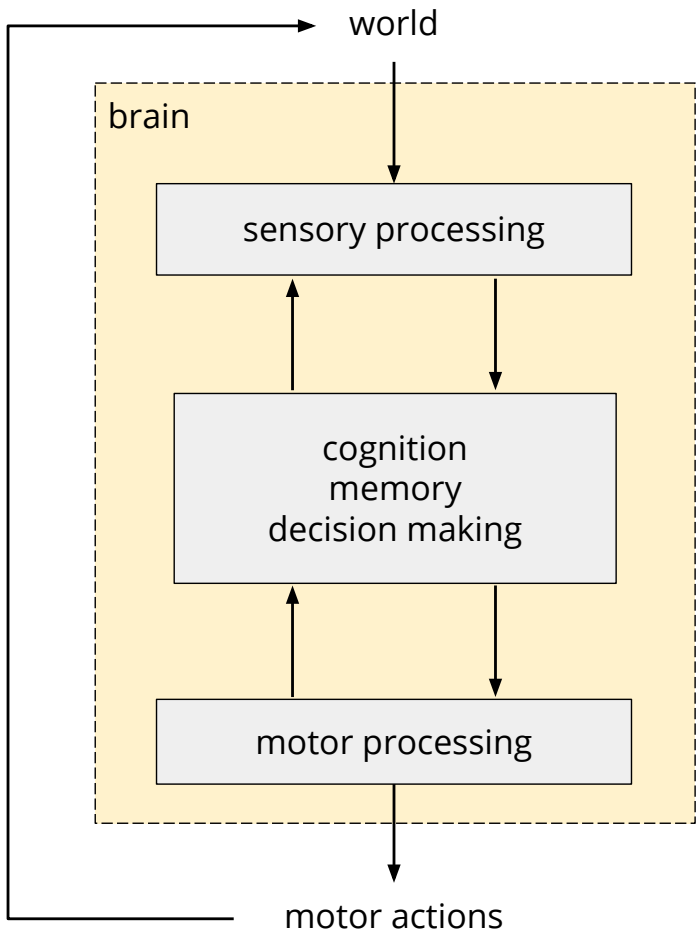


Figure 2.3: The brain-world interactive framework. The brain processes sensory information from the world using cognitive memory and decision-making to perform motor actions which are previously processed in the brain.

RL is, therefore, a method used to address optimal decision-making, attempting to maximize collected reward and minimize the punishment over time. It is a mechanism utilized by humans and in robotic agents. In developmental learning, it plays an important role since it allows infants to learn through exploration of the environment and connects experiences with pleasant feelings which are associated with higher levels of dopamine in the brain (Wise et al., 1978; Gershman and Niv, 2015).

The frontal cortex is known to play an important role in planning and decision-making (Payzan-LeNestour et al., 2013). Moreover, neurophysiology has shown the role of the basal ganglia and the frontal cortex in mammalian reinforcement

learning (Wimmer et al., 2012). Based on neuroscience evidence, the basal ganglia can be modeled by an actor-critic version of temporal difference learning (Rivest et al., 2004). RL has been shown in infant studies (Hämmerer and Eppinger, 2012; Deak et al., 2014) and in robotics (Kober et al., 2013; Kormushev et al., 2013) to be successful in terms of acquiring new skills, mapping situations to actions (Cangelosi and Schlesinger, 2015).

In developmental robotics (Cangelosi and Schlesinger, 2015) different tasks such as navigation, grasping, vision, speech recognition, and pattern recognition among others, can be tackled by different machine learning paradigms, like supervised, unsupervised or reinforcement learning (Bishop, 2011; Rieser and Lemon, 2011). In this thesis, we focus mainly on cognitive memory and decision-making which is the central part in Fig. 2.3, but, we also include some ideas about sensory processing to complement the decision-making process. In our approach, the autonomous agents are provided with no previous knowledge on how to perform tasks and they can learn only by making decisions when interacting with the environment and through the reward obtained. Therefore, the learning process is carried out with RL.

2.1.8 Interactive Reinforcement Learning in Autonomous Agents

As aforementioned, RL is a plausible method to develop goal-directed action strategies. During an episode, an agent explores the state space within the environment, selecting random actions which bring the agent into a new state. Over time, the agent learns the value of the states in terms of future reward, or reward proximity, and how to get to states with higher values to reach the target by performing actions (Weber et al., 2008).

To learn a task autonomously, an RL agent has to interact with its environment in order to collect enough knowledge about the intended task. RL has demonstrated to be a very useful learning approach; nevertheless, on some occasions, it is impractical to leave the agent to only learn autonomously, mainly due to time restrictions or in other words, the excessive time spent during the learning process (Knox and Stone, 2009), mainly due to large and complex state spaces which lead to excessive

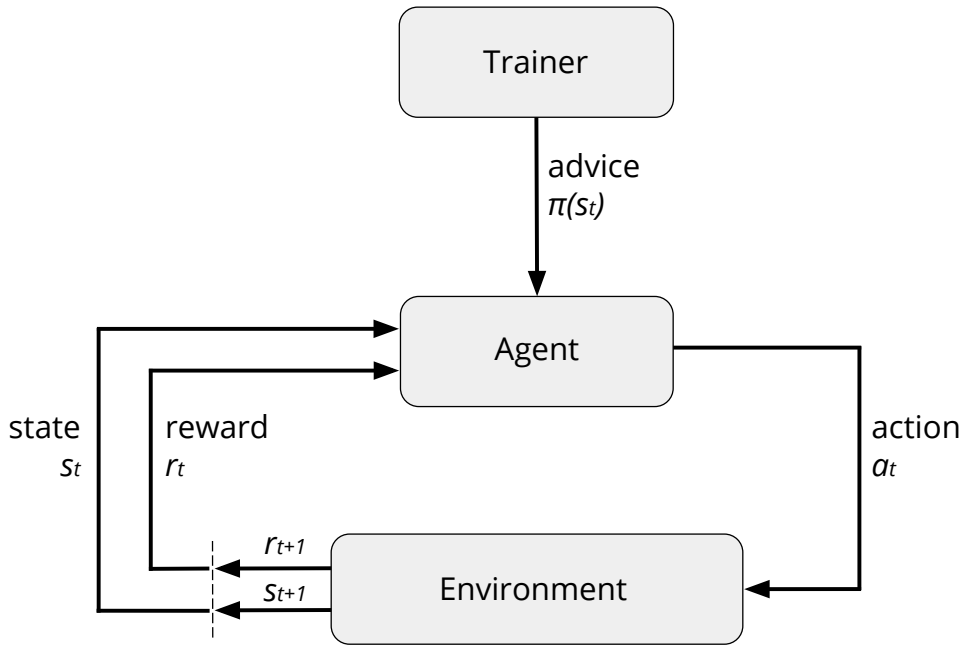


Figure 2.4: Interactive reinforcement learning extension including an external trainer. The trainer provides interactive feedback over the policy to the agent.

computational costs to find a suitable policy (Ammar et al., 2012). Therefore, we aim to find a way to accelerate the learning process. There are different approaches that attempt to speed up RL. Among them, interactive reinforcement learning (IRL) involves an external trainer who provides some instructions on how to improve the decision-making (Suay and Chernova, 2011; Grizou et al., 2013).

Fig. 2.4 shows a general view of the IRL approach where an external trainer is added to the learning process to communicate feedback to the learner-agent. Fig. 2.5 shows a typical human-robot interaction where a robot is assisted in its learning by a human parent-like trainer who sometimes delivers advice on what action to perform in order to complete the task faster.

In domestic and natural environments, adaptive agent behavior is needed, utilizing approaches used by humans and animals. IRL allows to speed up the apprenticeship process by using a parent-like advisor to support the learning by delivering useful advice in selected episodes. This allows to reduce the search space and thus to learn the task faster in comparison to an agent exploring fully autonomously (Suay and Chernova, 2011; Cruz et al., 2015). In this regard, the parent-like teacher guides the learning robot, enhancing its performance in the same manner as ex-

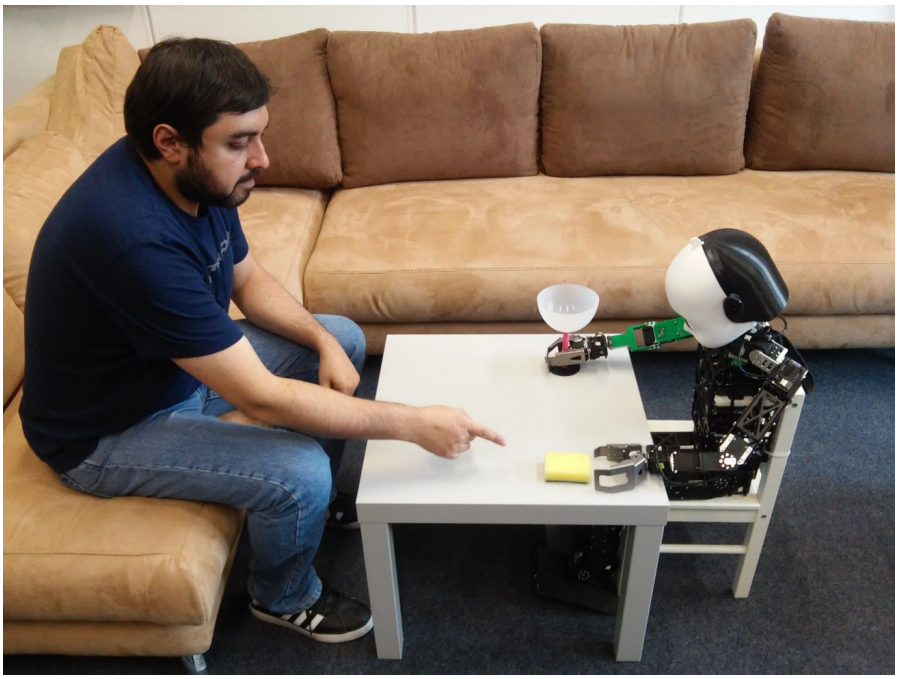


Figure 2.5: A scenario with human-robot interaction where the apprentice robot is supported by a parent-like trainer to complete the task.

ternal caregivers may support infants in the accomplishment of a given task, with the provided support frequently decreasing over time. This teaching technique has become known as parental scaffolding (Breazeal and Velásquez, 1998; Ugur et al., 2015).

When working autonomously, the next action is selected by choosing the best known action at the moment, represented by the highest state-action pair, but IRL speeds up the learning process by including the external advice in the apprenticeship loop. When using IRL, an action is interactively encouraged by a trainer with a priori knowledge about the desired goal (Thomaz et al., 2005; Thomaz and Breazeal, 2006; Knox et al., 2013b).

Early research on IRL (Lin, 1991) shows that external guidance plays an important role in learning tasks, performed by both humans and robots, leading to a decrease of the time needed for learning. Furthermore, in large spaces where a complete search through the whole search space is not possible, the trainer may lead the apprentice to explore more promising areas at early stages as well as help to avoid getting stuck in suboptimal solutions.

The external guidance can be implemented through different strategies of interaction between an agent and an external trainer for developing joint tasks, such as learning by imitation (Bandera et al., 2012), demonstration (Konidaris et al., 2012; Rozo et al., 2013; Peters et al., 2013), and feedback (Thomaz and Breazeal, 2006; Thomaz et al., 2005; Knox et al., 2013a).

In particular for learning by feedback two main approaches are distinguished: policy and reward shaping. Whereas in reward shaping an external trainer is able to evaluate how good or bad performed actions by the RL agent are (Thomaz et al., 2005; Knox and Stone, 2012), in policy shaping the action proposed by the RL agent can be replaced by a more suitable action chosen by the external trainer before it is executed (Cederborg et al., 2015; Amir et al., 2016). When the external trainer does not give feedback, acceptance of the action a or reward r is assumed. In both cases, an external trainer gives interactive feedback to the apprentice agent to encourage it to perform certain actions in certain states to reach a better policy leading to faster performance. Novel strategies can emerge from mixing both, namely, the advice on performing the action a and manipulating the received reward r as well.

Pilarski and Sutton (2012) propose that human training and direction methods can be projected to a two-dimensional space in terms of the explicitness and the bandwidth of the feedback signal. Explicitness refers to the content of explicit semantics in the signal with the reward (reward shaping) in one extreme and the instruction (policy shaping) in the other extreme. Bandwidth describes the complexity of the signal being the case of reward the simplest one and the case of instruction the most complex one including multisensory cues and real-time operation.

Fig. 2.6 shows the policy shaping approach in IRL through feedback, where interaction from an external trainer is given during the robot's action selection. Manipulating actions is a way to tell the agent that what it is currently doing is wrong and should be corrected in the future (Thomaz and Breazeal, 2007). The reward shaping approach is shown in Fig. 2.7. In this case, the external trainer may modify the reward r and send its own reward to the agent specifying how good or how bad the latest performed action a was. Examples of this approach were developed by Thomaz and Breazeal (2006) and Knox and Stone (2012).

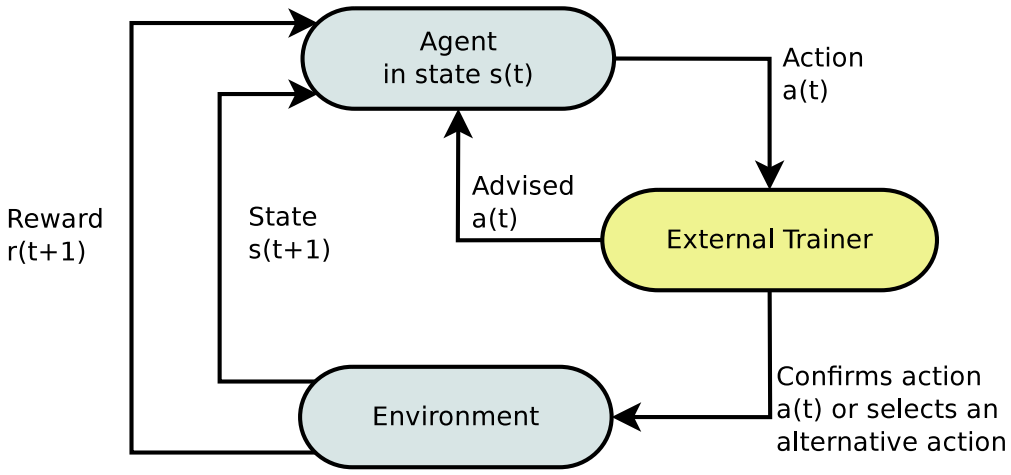


Figure 2.6: Policy shaping feedback approach for interaction between a robotic agent and an external trainer. In this case, the external agent is able to change a selected action to be performed in the environment.

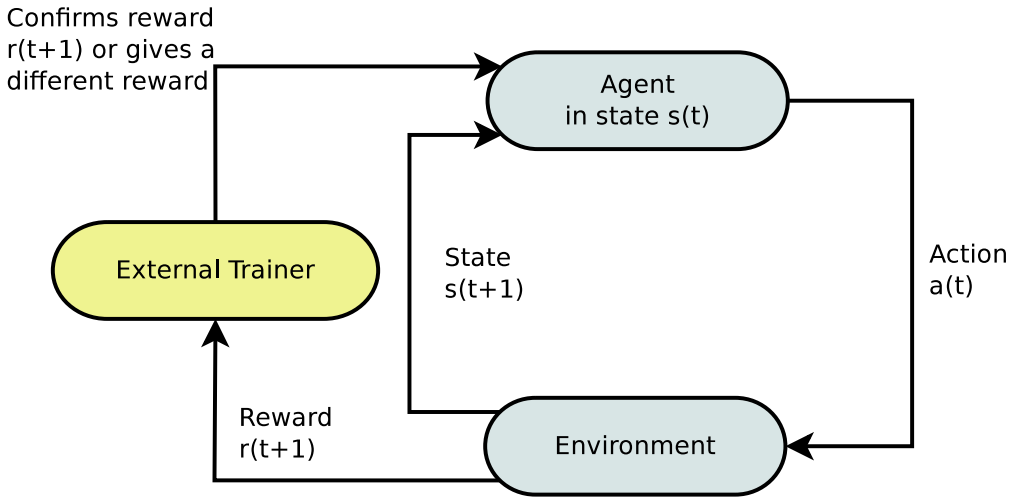


Figure 2.7: Reward shaping feedback approach for interaction between a robotic agent and an external trainer. In this case, the external agent is able to modify the proposed reward.

In an IRL scenario it is desired to keep the rate of interaction with an external trainer as low as possible; otherwise, with a high rate of interaction, RL becomes supervised learning. Also, the consistency or quality of the feedback should be considered to determine whether learning is still improving given that the external trainer could also make mistakes (Griffith et al., 2013). Supportive advice can be obtained from diverse sources like expert and non-expert humans, artificial agents

with perfect knowledge about the task, or previously trained artificial agents with certain knowledge about the task. In this thesis, we use both human and artificial trainer-agents. The artificial trainers are themselves previously trained through autonomous RL and afterward, they are used to provide advice, which has been formerly used in other works. For instance, in (Cruz et al., 2014, 2016a) advice is given based on an interaction probability and consistency of feedback. In Taylor's works, the interaction is based on a maximal budget of advice and they studied which moment is better to give advice during the training (Torrey and Taylor, 2013; Taylor et al., 2014).

In the following section, we will review affordances as an alternative method which enables to speed up RL. We will introduce it into the IRL framework in order to allow a learner-agent to speed up the learning process working with both interactive feedback and affordances.

2.2 Affordances

A promising alternative method to improve RL convergence speed by modeling the actions in the environment is the use of affordances (Wang et al., 2013), where cognitive agents favor specific actions to be performed with specific objects. Affordances represent neither agent nor object characteristics, but rather the characteristics of the relationship between them (Gibson, 1979). Affordances limit the number of meaningful actions in some states and can reduce the computational complexity of RL.

2.2.1 Gibson's Proposal

Affordances are often seen as opportunities for action of an agent (a person, an animal, a robot, or an organism). The original concept comes from cognitive psychology and was proposed by Gibson (1966, 1979) as:

> *"When the constant properties of constant objects are perceived (the shape, size, color, texture, composition, motion, animation, and position relative to other objects), the observer can go on to detect their*

> *affordances. I have coined this word as a substitute for values, a term which carries an old burden of philosophical meaning. I mean simply what things furnish, for good or ill. What they afford the observer, after all, depends on their properties."*

For instance, a soccer ball and a skateboard are objects which afford different actions. An agent interacting with these objects may kick the soccer ball or ride the skateboard, whereas the agent may not do the opposite. Let us consider another example: a cup and a sofa afford different actions to a person who is able to grasp the cup and sit down on the sofa but cannot do it the other way around. Thus, an agent is able to determine some object affordances, e.g., the caused effect of performing a specific action with an object.

In Gibson's book, many diverse examples are given but no concrete, formal definition is provided. Even nowadays, we find marked differences among cognitive psychologists about the formal definition of affordances (Horton et al., 2012; Chemero, 2011) and these discrepancies could even be stronger between them and artificial intelligence (AI) scientists (Şahin et al., 2007; Chemero and Turvey, 2007).

Horton et al. (2012) distinguish three essential characteristics of an affordance:

- The existence of an affordance is associated with the capabilities of an agent;
- An affordance exists regardless whether the agent is able to perceive it or not;
- Affordances do not change, unlike necessities or goals of an agent.

2.2.2 Developmental Robotics Perspective

In developmental robotics, affordances are aligned with basic cognitive skills which are acquired on top of previous skills by interacting with the environment (Moldovan et al., 2012). It is expected that domestic service robots learn, recognize, and apply some social norms in the same way as humans do. Commonly these social rules are learned by interaction and socialization with other agents of the group. In this regard, an object can be used in a restricted manner not considering all its action opportunities but only socially accepted actions. These constraints of use are usually shaped by the group norms and are called functional affordances

(Awaad et al., 2015) which also lead to a reduced action space. Such a human-like behavior is an important issue in developmental robotics (Sigaud and Droniou, 2016).

In the literature we can find different approaches for learning affordances in robotics; for instance, Lopez et. al address the imitation learning problem using affordance-based action sequences (Lopes et al., 2007). Moldovan et al. (2012) extend the affordance model allowing the robot to work with a second object using an enlarged Bayesian network to represent affordances.

Jamone et al. (2017) recently studied affordances taking into consideration three main aspects from different areas: psychology, neuroscience, and robotics. Furthermore, Min et al. (2016) surveyed affordance research particularly in the field of developmental robotics showing interesting insights on how affordances could serve as a basis to develop architectures and algorithmic principles within artificial cognitive systems. In the following subsection, we present a formal computational definition based on the original concept of Gibson.

2.2.3 Formalization of the Model

Affordances have been particularly useful to establish relationships between actions performed by an agent with available objects. They have been utilized in a way to represent object/action information (Cruz et al., 2016a). Montesano et al. (2008) define an affordance as the relationship between an object, an action, and an effect as the 3-tuple which can be written as:

$$affordance := < object, action, effect > \tag{2.11}$$

Hence, it is possible to predict the effect using objects and actions as domain variables, i.e.:

$$effect = f(object, action) \tag{2.12}$$

Fig. 2.8 shows the relationship between the previous components, where objects are entities which the agent is able to interact with; actions represent the behavior

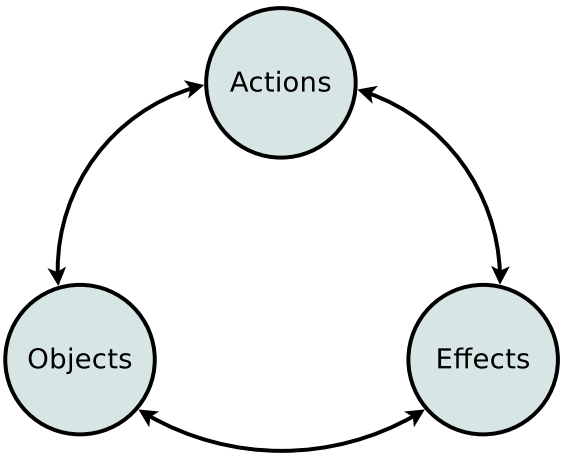


Figure 2.8: Affordances as relations between objects, actions, and effects. Objects are entities which the agent is able to interact with, actions represent the behavior that can be performed with the objects, and effects are the results caused by applying an action (Montesano et al., 2008).

Table 2.1: Uses of learned affordances by utilizing bi-directional mapping.

Input	**Output**	**Functionality**
(object, action)	effect	Predict effect
(object, effect)	action	Action planning and recognition
(action, effect)	object	Object selection and recognition

or motor skills that can be performed with the objects; and the effects are the results of an action involving an object (Montesano et al., 2008; Atıl et al., 2010).

It is also important to note that the object in Eq. (2.12) can also be a place or a location, for instance, a hill affords climbing. From here onwards, we employ the term *object* to refer to the affordance component but we consider also locations. Once affordances are recognized and learned through the components of the 3-tuple, it is possible to establish bi-directional mappings with different purposes (Lopes et al., 2007), as shown in Table 2.1.

Figure 2.9: All the objects in the picture afford to grasp. Nevertheless, if a human agent has no free hand in a particular moment, then the affordance of graspability is temporally unavailable in that state until the agent releases an object.

2.2.4 Implications for Agent Control

The use of affordances in robotics allows to address much more interesting problems by reducing the action space due to retrieved relevant information from the world or environment, allowing to identify what actions are possible for a robot to perform (Şahin et al., 2007). Nevertheless, although the aforementioned formalized model has been shown to be suitable for many scenarios, it does not include context information which allows anticipating effects in all situations properly (Kammer et al., 2011).

For instance, let us consider the following scenario in which we are given a set of objects which afford to grasp (as do the ones shown in Fig. 2.9). In case we have an agent with both hands already occupied with objects, then the agent cannot grasp a new object. In other words, the affordance of graspability is temporarily unavailable until the agent places one object back on the table which in turn modifies the context.

The fact of not considering the context leads the agent to face issues at the moment of deciding what actions to take. Therefore, to control the agent, in terms of the performed actions with an object and trying to predict the caused effect, presents

a more complex task to the learner-agent. Later on in this thesis, we will define an extension of the proposed model by Montesano et al. (2008) to take environmental variables into account and enable the agent to better decide between available actions and the corresponding effects.

2.3 Discussion

In this chapter, we have reviewed the main approaches related to our work, which will be used throughout this thesis. First, we have reviewed the RL basics and the IRL framework. In this thesis, we address a scenario modeled as a Markov decision process using RL agents with temporal-difference learning inspired by behavioral approaches. Autonomous agents are provided with interactive advice using the IRL framework.

Afterward, we have reviewed affordances as an alternative to model the actions in the environment. One alternative to implementing affordances is to use artificial neural networks in which actions and objects may be used as inputs to anticipate the effect of such an action. With this, we conclude the review of the theoretical framework and the main related works.

In the following chapter, we introduce a robotic scenario as a Markov decision process in which we test the proposed methods in different experimental set-ups. The set-ups are shown in an agent-agent IRL framework and a human-agent IRL framework.

Chapter 3

Robotic Cleaning-table Scenario

3.1 Introduction

With the aim of evaluating the proposed methods and answering the research questions, we have designed a simulated domestic scenario focused on cleaning a table. We model the problem as a Markov decision process as described in Sec. 2.1.4. Therefore, the task can be learned by an agent if a definition of the states, the actions, the transition function, and the reward function is given. In this scenario, we have included objects, locations, and actions. Initially, a robotic agent has a *sponge* and is standing in front of a table, in particular in front of a specific area of the table which is desired to be cleaned.

We implement the robotic home scenario with an agent interacting with a parent-like trainer to perform the cleaning task. The task of cleaning the table is performed with the use of the robot's right arm. To successfully complete the cleaning task, it is necessary to carry out additional subtasks such as interacting with objects on the table. The trainer is able to advise the learner robot on what action to perform next.

Initially, we implement the proposed scenario by using agent-agent interaction and later on we extend it to human-agent interaction. By using agent-agent interaction we attempt to mimic a real human-agent interaction as much as possible and

at the same time, having an artificial trainer-agent allows to better control the experimental variables. By using human-agent interaction we first developed the IRL scenario with automatic speech recognition only to guide an apprentice robot in the achievement of a task and, afterward, we extend the approach to incorporate visual information and integrate it with audio as a more robust guidance during the apprenticeship process.

3.2 Domestic Scenario

The scenario is motivated by the increasing presence of robots in home environments. For robots to reach human-like performance on challenging tasks, they still need to be boosted in order to satisfactorily complete a given task. Therefore, the proposed cleaning-table scenario shows a daily-based task which can be performed by robots assisted by human trainers as advisors.

The proposed scenario comprises three locations, two objects, and seven actions. The robot is assumed to be placed in front of a table in order to clean it. The three locations defined in the cleaning-table scenario are:

i. *left*, the *left* section of the table;

ii. *right*, the *right* section of the table;

iii. *home*, an additional position that is the initial and final position of the robot's arm.

The scenario comprises two objects that the robot can manipulate using its gripper. The two objects are:

i. *sponge*, used to clean both sections of the table. The *sponge* is placed at the *home* position while not being used by the robot;

ii. *goblet*, initially placed in one of the sections of the table and, therefore, it must be moved from one section to the other during cleaning in order to end the task successfully.

In this thesis, the experiments are conducted using artificial agents and simulated robots. Fig. 3.1 shows an example of the domestic scenario in a simulated envi-

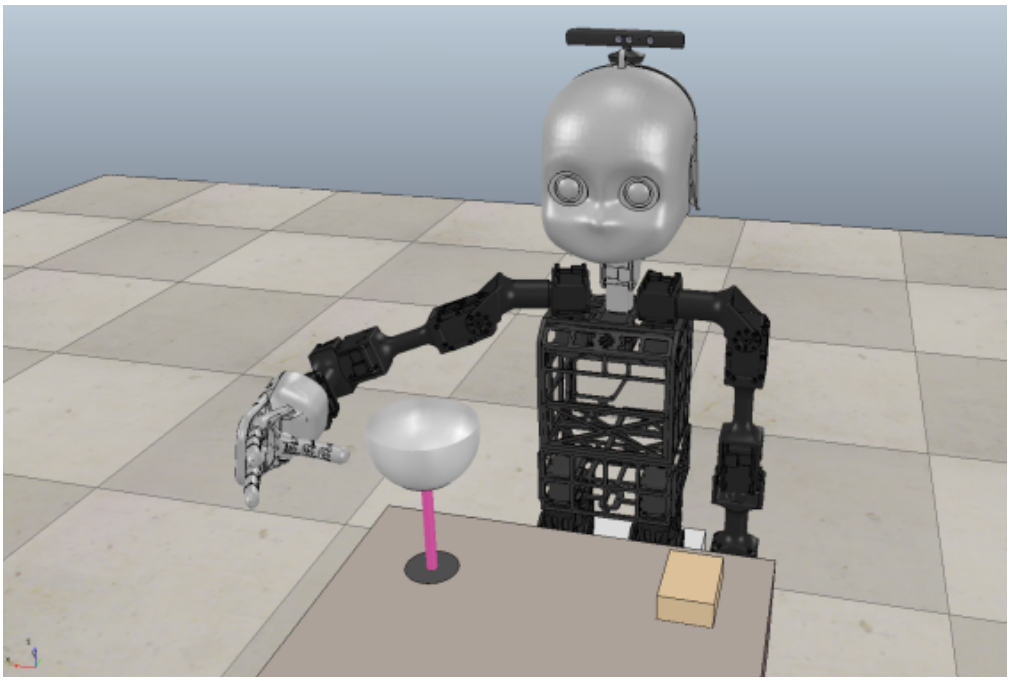


Figure 3.1: The simulated domestic scenario with the NICO robot. Our scheme is composed of two objects, 3 locations, and 7 action classes.

ronment. In the figures, it is possible to observe the robot in front of the table with two objects: the *goblet* and the *sponge*.

3.3 Markov Decision Process Definition

In our domestic task, transitions and rewards depend only on the current state and the chosen action by the agent, hence, we model the task as an MDP problem. The following section defines the robot scenario as an MDP. Following the actions, the states, the transitions, and the reward function are presented.

3.3.1 Actions

The robot can perform seven different actions. These actions may be chosen by the robot autonomously or through advice given by either an artificial trainer-agent or a parent-like trainer using multi-modal feedback for audiovisual advice.

Available actions are as follows:

Table 3.1: List of defined objects, locations, and actions for the cleaning-table scenario.

Objects	**Locations**	**Actions**	
sponge	left	go <location>	
goblet	right	get	drop
	home	clean	abort

(i) `GET:` allows the robot to pick up the object which is placed in the same location as its hand.

(ii) `DROP:` allows the robot to put down the object held in its hand. The object is placed in the location where the hand is.

(iii) `GO HOME:` moves the hand to the *home* position.

(iv) `GO LEFT:` moves the hand to the *left* position.

(v) `GO RIGHT:` moves the hand to the *right* position.

(vi) `CLEAN:` allows the robot to *clean* the section of the table at the current hand position if holding the *sponge*.

(vii) `ABORT:` enables to cancel the execution of the cleaning task at any time and return to the initial state.

Table 3.1 shows a summary of objects, locations, and actions defined for this domestic cleaning scenario. For instance, let us now suppose that the *goblet* is located on the *left* side of the table at the beginning. The initial position of the robot's hand is the location *home*, and we want to finish with the hand *free* and above *home* with both sides of the table clean. In this context, an episode is defined as one attempt to reach the goal. The following example shows the shortest episode to complete this task successfully: *get, go right, clean, go home, drop, go left, get, go right, drop, go home, get, go left, clean, go home, drop.* The example shows, for one initial situation, the shortest sequence of actions to reach the final state; therefore, the minimum number of actions to complete the cleaning-table task is $|A_{min}| = 15$.

3.3.2 States

To implement the described scenario we developed a state machine with two final states; each state is represented by a state vector of four variables. These variables are:

i. `handObject`: the object which is currently in the robot's hand, i.e., *sponge,* *goblet,* or *free*;

ii. `handPosition`: the position of the robot's arm, i.e., *left, right,* or *home*;

iii. `gobletPosition`: the position of the *goblet* on the table, i.e., *left, right,* or *home*;

iv. `sideCondition[]`: a 2-tuple with the current condition of every side of the table surface, that is, whether the table location has already been cleaned or not.

Therefore, the state vector at any time t is characterized as follows:

$$s_t =< handObject, handPosition, gobletPosition, sideCondition[] > \tag{3.1}$$

At the beginning of each training episode, the robot's hand is *free* at the *home* location, the *sponge* is also placed at the *home* position, while the *goblet* is at either the *left* or the *right* location, and both table sections are *dirty.* Therefore, the initial state s_0 may be represented as:

$$s_0 =< free, home, left|right, [dirty, dirty] > \tag{3.2}$$

To complete the task, the robot must clean both sections of the table by moving the *goblet* from one section to the other during the process of cleaning. After the robot has cleaned both sections, the task is finished when the *sponge* is placed at the *home* position and the robot is with the hand *free.* Therefore, the final state s_f can be represented as:

$$s_f =< free, home, left|right, [clean, clean] > \tag{3.3}$$

From certain states, the agent can perform non-reversible actions which lead to a failed-state from where it is not possible anymore to complete the task. These actions include getting an object when the robot's hand is occupied, to lose either the *goblet* or the *sponge* due to an incorrect *drop*, or cleaning a section of the table where the *goblet* is also placed on.

Definition 3.1. Failed-state: Let us consider the set of states S and the set of actions A. Given one sequence of states $\psi_s = \{s_t, s_{t+1}, s_{t+2}, ..., s_n\}$ with $s_i \in S$ and one sequence of actions $\psi_a = \{a_t, a_{t+1}, a_{t+2}, ..., a_n\}$ which leads to ψ_s with $a_i \in A$. Then $s_i = f(s_{i-1}, a_{i-1})$ for $0 < i \leq n$. Now, let $\Psi_A(s_t)$ be the set of all possible ψ_a from a state $s_t \in S$. If $\nexists\ \psi_a \in \Psi_A(s_t)$ which produces a $\psi_s \mid s_f \in \psi_s$ with s_f the final state $\implies$ s_t is a failed-state.

For instance, let us assume that the robot's hand and the *goblet* are at the *left* location and the robot has just performed the action *get* from the state s_t = `<free, left, left, [dirty, dirty]>` transiting to the next state s_{t+1} = `<goblet, left, left, [dirty, dirty]>` according to the four previous state variables; therefore now the *goblet* is held in its hand over the *left* position. If the robot then cleans the *left* section of the table with the *goblet* in its hand instead of a *sponge*, it may shatter the *goblet*; hence, it is not feasible to finish the cleaning task from the following state s_{t+2}.

The scenario consists of 53 regular states, i.e.: the agent is still able to complete the task successfully from these states. Some regular states are shown in Table 3.2 where the initial state is labeled as state number 1 and the final states are the numbers 52 and 53. The states between number 16 and 30, and the states between 31 and 45 are not shown since they are the same as the first 15 states apart from the *sideCondition* which are $[clean, dirty]$ and $[dirty, clean]$ respectively (for more details, refer to Apendix B).

3.3.3 Transition Function

From the initial state, the state vector is updated every time after performing an action according to the state transition table as shown in Table 3.3. The table shows the seven possible actions. For the action *go <pos>* there are three different

Table 3.2: Regular states defined for the cleaning-table scenario. States labeled as 52 and 53 represent final states once the task is completed.

Number	**handObject**	**handPosition**	**gobletPosition**	**sideCondition[]**
1	free	home	left	[dirty, dirty]
2	sponge			
3	free	left		
4	sponge			
5	goblet			
6	free	right		
7	sponge			
8	free	home	right	
9	sponge			
10	free	left		
11	sponge			
12	free	right		
13	sponge			
14	goblet			
15	goblet	home	home	
...	...	...	...	...
46	sponge	right	left	[clean, clean]
47	sponge	left		
48	sponge	home		
49	sponge	left	right	
50	sponge	right		
51	sponge	home		
52	free	home	left	
53	free	home	right	

Table 3.3: State vector transitions. After performing an action the agent reaches either a new state or a failed-state, if the latter, the agent starts another training episode from the initial state s_0. In the proposed scenario, the action *go <pos>* includes three different possibilities, i.e., *go home, go left,* and *go right.*

Action	**State vector update**
Get	**if** handPos == *home* **&&** handObj == *goblet* **then** FAILED
	if handPos == gobletPos **&&** handObj == *sponge* **then** FAILED
	if handPos == *home* **then** handObj = *sponge*
	if handPos == gobletPos **then** handObj = *goblet*
Drop	**if** handPos == *home* **&&** handObj == *goblet* **then** FAILED
	if handPos != *home* **&&** handObj == *sponge* **then** FAILED
	otherwise handObj = *free*
Go <pos>	handPos = pos
	if handObj == *goblet* **then** gobletPos = pos
Clean	**if** handPos == gobletPos **then** FAILED
	if handPos == *home* **then** FAILED
	if handObj == *sponge* **then** sideCond[handPos] = *clean*
Abort	handPos = *home*
	handObj = *free*
	gobletPos = random(pos)
	sideCond = [dirty]*|pos|

possibilities, i.e., *go home, go left,* and *go right.* In the current scenario, considering the state vector features, the 53 regular states represent two divergent paths to two final states.

Assuming that the *goblet* is on the *left* side of the table at the beginning, there are two feasible paths for reaching a final state from the initial state. The upper path (Path A) consists of cleaning first the empty side of the table and then moving the *goblet* in order to clean the second side. The lower path (Path B) would consist

of moving the *goblet* first to the *right* side and cleaning the *left*, and after that returning the *goblet* to its original side for cleaning the second one. In each case, a final state is reached involving different numbers of intermediate states. The ending sequence of each path contains four states in which the robot returns its hand to the *home* location and drops the *sponge*.

Fig. 3.2 depicts a summarized illustration of the state transitions to reach a final state assuming the *goblet* to be at the *left* position initially. It can be observed at the initial state that both sides of the table are *dirty* and from there on the two possible paths to finish the task. Internally, in our algorithm we do not use *left* or *right*, but rather *side1* and *side2*, where *side1* is the side of the table where the *goblet* is at the beginning of an episode, and it is feasible to start cleaning any side of the table because both paths can lead to a final, successful state. The figure also shows the number of intermediate states involved in each path. In order to visualize the whole search space, all the state transitions can be seen in Appendix B. Each path leads to a different number of transited states which in turn also leads to a different accumulated reward (see Sec. 3.3.4). As we already stated above, the shortest path is composed of 15 actions for reaching the final state through the path A.

As defined, the same transitions may be used in scaled-up scenarios with more locations on the table in a larger grid, since the definition of transitions includes only the object held by the robot and the hand position in reference to either the *home* location or the *goblet* position.

3.3.4 Reward Function

As long as the agent successfully finishes the task, a reward equal to 1 is given to it, whereas a reward of -1 is given if a failed-state has been reached. Furthermore, it is given a small negative reward of -0.01 for each transition in order to discourage longer paths and loops. Therefore, the reward function can be summarized as follows:

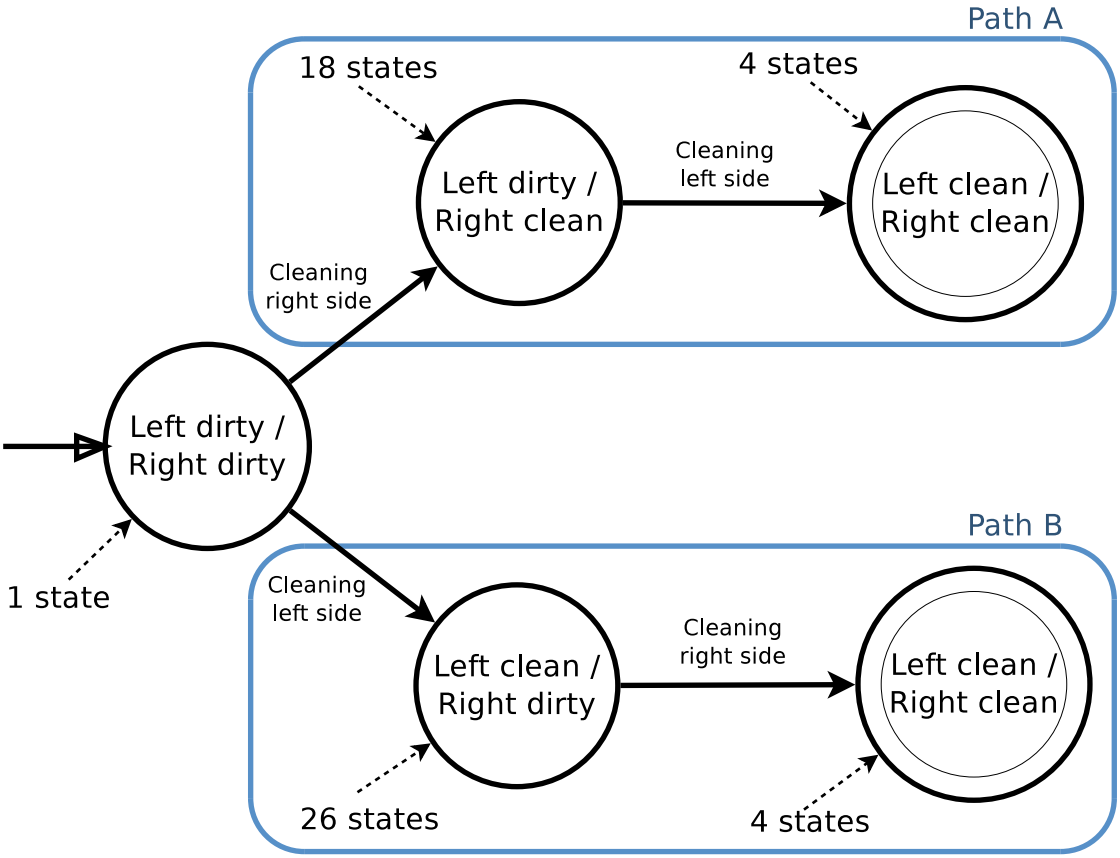


Figure 3.2: Outline of state transitions in the defined cleaning-table scenario. Two different paths are possible to reach a final state. Each path implies a different number of intermediate states which influence the total amount of collected reward during a learning episode. Path A comprises 23 states and B 31 states. See more details in Appendix B.

$$r(s) = \begin{cases} 1 & \text{if } s \text{ is the final state} \\ -1 & \text{if } s \text{ is a failed-state} \\ -0.01 & \text{otherwise} \end{cases} \tag{3.4}$$

3.4 Parent-like Advice

Although the robot is able to perform actions autonomously using RL, by using a parent-like trainer to advise the robot at specific steps about what action to perform next, we want to reduce the time required to learn the sequence of actions

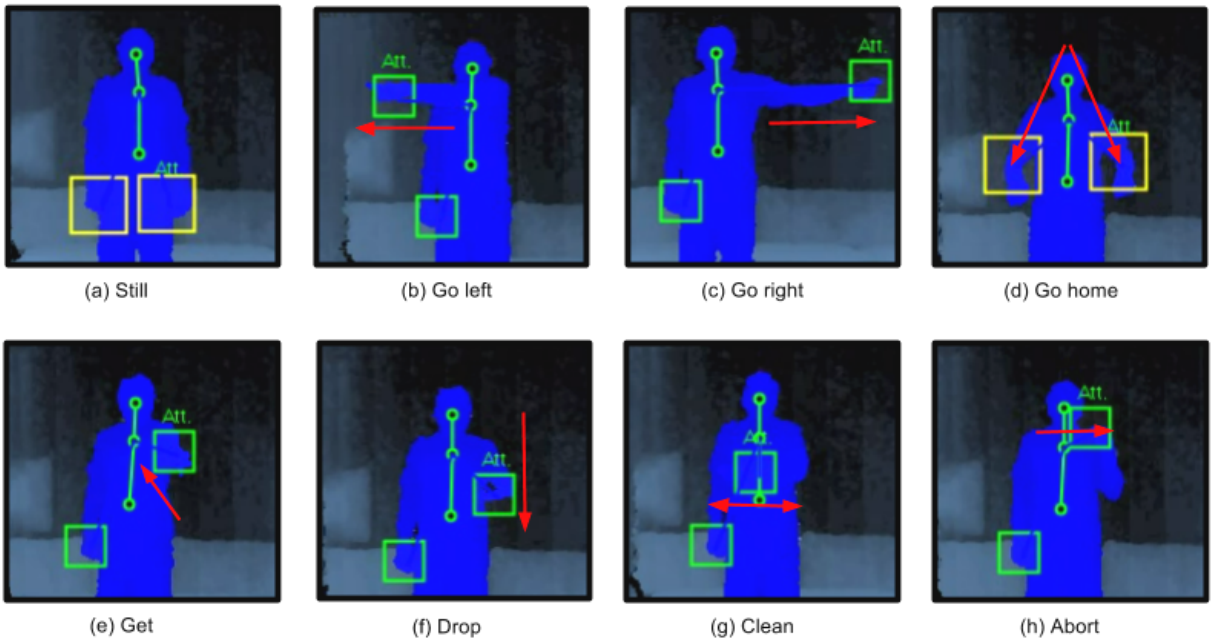


Figure 3.3: Gestures used as advice in the robotic scenario. Red arrows represent the hand movement performed to advise the robot.

for finishing the task.

For our scenario, we define a set of possible advice classes that can be given to the robot by a trainer. The trainer can be either another previously-trained artificial agent or a human with knowledge about the task. When using artificial trainer-agents, we assume the advice cues to be directly observable, however in the case of a human parent-like trainer, they may use audiovisual signals to provide interactive feedback to the learner-agent, therefore, each advice class has a spoken representation in a domain-based language and a visual representation with gestures from vision.

The advice can be delivered at any time with the following advice classes: *go left, go right, go home, get, drop, clean,* and *abort*. In the case of audio inputs, we use 33 voice commands belonging to the seven aforementioned classes. For example, the instruction *go right* could also be stated as *go to the right*, *move (to the) right*, or *change (to the) right*, but all of them would belong to the same advice class. In the case of visual inputs, we include seven gestures, each of them representing an advice class. Additionally, since gesture labels are continuously predicted from

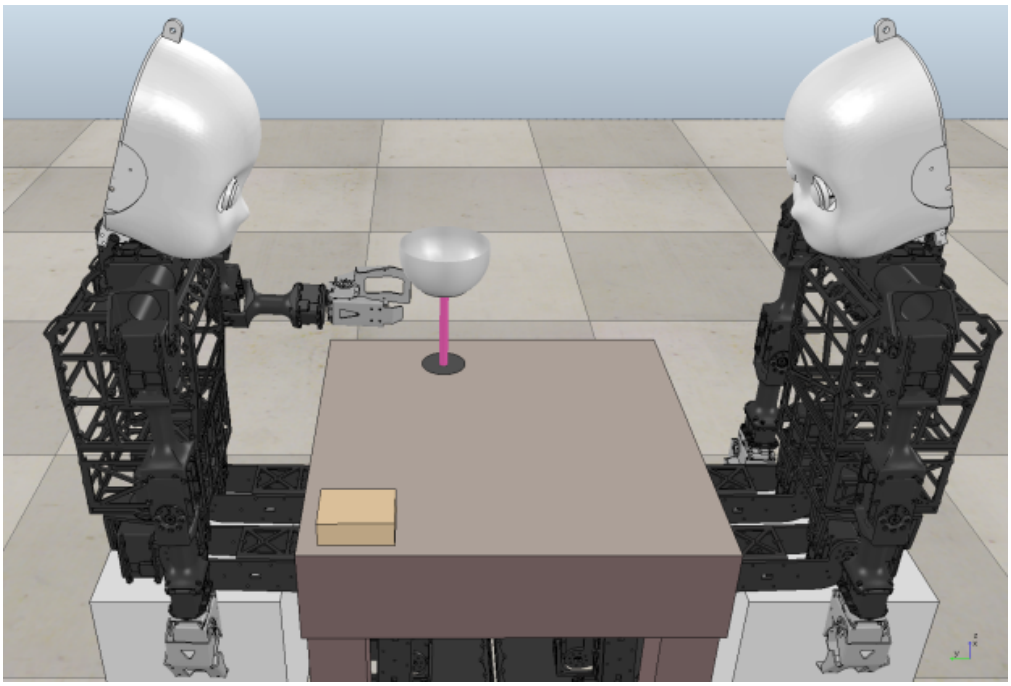


Figure 3.4: An example of the simulated home scenario where agents perform the actions in the environment which is created in a robot simulator. The trainer-agent to the left advises the learner-agent in selected states what action to perform next.

depth map video sequences, we add the label *still* to indicate no advice at that moment. Fig. 3.3 shows the gesture advice classes using RGB-D information from a depth sensor.

The cleaning-table task is carried out by a robot in a simulated environment using the V-REP simulator (Rohmer et al., 2013). All actions are performed using only one arm and one effector. Fig. 3.4 shows an example of the domestic robotic scenario with two robotic agents where one agent, which has already learned the task by using autonomous RL, becomes the trainer of a second robot. The second agent performs the same task supported by the trainer-agent with selected advice using the IRL framework. We did not focus on investigating grasping since the main aim of this work is to learn the right sequence quickly. Nevertheless, for reaching the defined locations we employed direct planning and for grasping inverse kinematics as a support for low-level control. Both approaches are available in the robot simulator (Rohmer et al., 2013).

3.5 Discussion

In this chapter, we have presented a domestic robot scenario. The scenario is defined as a Markov decision process with actions, states, the transition function, and the reward function. Moreover, in the IRL context, we have introduced the parent-like advice in terms of auditory and visual feedback. The defined simulated scenario is used in the following chapters to investigate different experimental setups in order to test our proposed methods and compare the results between the RL and IRL frameworks.

First, we will show an agent-agent IRL scenario with an artificial agent, previously trained by RL, being the trainer-agent. In this context, we introduce an affordance-driven model to avoid failed-states and we also explore the internal representation of the knowledge in trainer-agents and the effects by considering such different representations. Moreover, we study the interplay of interaction parameters as the probability of receiving feedback, the consistency of feedback, and the learner-agent's obedience.

Subsequently, we will show a human-agent IRL scenario with human parent-like trainers advising the learner-agents. We perform experiments using as advice uni- and multi-modal feedback cues in terms of audio and audiovisual signals respectively.

Part II

Agent-Agent Interactive Reinforcement Learning

Chapter 4

Interactive Feedback and Contextual Affordances

4.1 Introduction

In robotics, there has been considerable progress in the last years allowing robots to be successful in diverse scenarios, from industrial environments where they are nowadays established to domestic environments where their presence is still limited (Tadele et al., 2014). In domestic environments, tasks often require active parent-like participation in order to execute the tasks more effectively. In particular, in the simulated home scenario which we have proposed in chapter 3, the agent has to perform the task assisted by an external trainer giving different degrees of guidance.

In this chapter, we present an IRL approach for the domestic task of cleaning a table and compare three different learning methods using simulated robots: reinforcement learning (RL), RL with contextual affordances to avoid failed-states, and the previously trained robot serving as a trainer to a second apprentice robot in an interactive reinforcement learning (IRL) framework. The experimental set-up is designed and performed in order to answer the first research question of this thesis: How can an affordance-based model of the environment support the IRL framework?

Contextual affordances are a generalization of Gibson's affordance concept. The

latter has recently been used successfully in robotics (Horton et al., 2012; Min et al., 2016; Jamone et al., 2017). As mentioned before, we are interested in introducing contextual affordances into the IRL framework in order to allow learner-agents to speed up the overall learning process working with both interactive feedback and affordances. In this regard, contextual affordances allow to avoid failed-states which enable the agent to complete the task in fewer episodes. We implement contextual affordances with an artificial neural network (ANN) to estimate either the robot's next state or whether the affordance is temporally unavailable. Although the agent working autonomously must be able to complete the task with RL, the learning process is expected to be slow and with a low rate of success. We want to show that working with RL and contextual affordances, fewer actions are needed and can reach higher rates of success.

Afterward, we test different levels of interaction and consistency of feedback to show how they influence the IRL performance. For good performance with IRL, considering the level of consistency of feedback is essential, since inconsistencies can cause considerable delay in the learning process. In general, we want to demonstrate that interactive feedback provides an advantage for learner-agents in most of the learning cases.

4.2 Contextual Affordances

As stated, we are interested in reducing the needed actions of an episode to reach a reasonable performance in both approaches, RL and IRL. This becomes especially important when it is desired to work in real scenarios, because, while in simulated environments it is feasible to run many episodes in a short time, in a real environment one cannot afford to run excessive episodes until reaching a suitable policy. To this aim, we use affordances to reduce the number of actions involved in each episode.

If an affordance exists and the agent has awareness of it, the actual, next step is to determine if it is possible to utilize it considering the agent's current state. If the affordance is temporally unavailable, as shown in Fig. 2.9, this does not mean that the affordance does not exist, to the contrary, the affordance is still present but it just cannot be used by the agent in that particular situation.

Kammer et al. (2011) proposed to consider the dynamics in the environment in which the object was embedded rather than dynamic states of the agent. The awareness of this extra variable is called situated affordances (Kammer et al., 2011). Even though a formal definition was provided, neither applications nor results are shown in their work. Nevertheless, we use the same concept to address the problem when the agent's state is also dynamic. Thus, we propose a model where the current state of an agent is also considered for the effects of an action performed with an object, we call this contextual affordance. In this case, the affordances 3-tuple shown in Eq. (2.11) is now extended to:

$$contextualAffordance :=< state, object, action, effect > \tag{4.1}$$

Now, to predict the effect after performing an action, we consider the following function:

$$effect = f(state, object, action) \tag{4.2}$$

For instance, given an agent performing the same action a with the same object o, but from a different agent's state $s_1 \neq s_2$: when action a is performed, different effects $e_1 \neq e_2$ could be generated, since the initial states s_1 and s_2 are different. It is unfeasible to establish differences in the final effect when we utilize affordances to represent it, because $e_1 = (a, o)$ and $e_2 = (a, o)$. Hence, to deal with the current states $s_1 \neq s_2$, an agent must distinguish each case and learn them at the same time utilizing contextual affordances defined by $e_1 = (s_1, a, o)$ and $e_2 = (s_2, a, o)$, establishing clear differences between the final effects.

Fig. 4.1 shows the relationship between object, action, effect, and the agent's current state. We use contextual affordances to provide knowledge about actions that lead to undesirable or failed-states from which it is not possible to reach the goal. Therefore, the action space is reduced by avoiding these states. In our approach, the set of possible actions is filtered for every state that the agent is in.

At first, we ran a classic RL algorithm to confirm that failed-states are not a problem as such; they can be handled and controlled by giving punishment (or negative reward). In this case, the agent is discouraged to perform that action from the same state in the future. However, a more suitable strategy is to consider

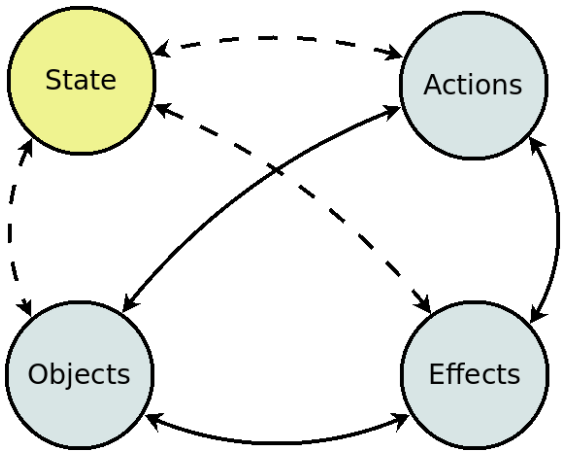


Figure 4.1: Contextual affordances as relations between state, objects, actions, and effects. The state is the agent's current condition and different effects could be produced for different occasions.

the use of affordances which have been shown to improve the convergence speed of learning algorithms (Koppula et al., 2013; Kober and Peters, 2012).

The contextual affordance model allows us to determine beforehand when it is possible to apply an affordance. Given the robotic scenario defined in Sec. 3, we are able to set the presence of four different contextual affordances which allow us to determine whether objects are *graspable, droppable, movable,* or *cleanable* according to the robot's current state.

4.3 Experimental Set-up

To carry out the experiments, first, we used the classic RL approach to train a robot for reaching the final state. Afterward, we introduced contextual affordances attempting to reduce the needed episodes to obtain a satisfactory performance in terms of collected reward and performed actions for reaching the final state. Finally, a second agent was trained using IRL and receiving feedback from the previously trained robot that sometimes showed which action to choose in a specific state. The detailed implementation of these methods is explained in this section, while the obtained results are shown in the following section.

4.3.1 Learning Contextual Affordances with a Neural Architecture

The model with contextual affordances enables us to predict the effects of specific actions by using a simplified artificial neural network that learns the relationship of the states, the actions, and the objects. Contextual affordances are used in both autonomous RL and IRL, i.e., the selected action either by the agent or the parent-like trainer may be disallowed if the effect of performing such an action leads the agent to a failed-state.

It has been shown that a multilayer feedforward neural network with only one hidden layer and a sufficient number of neurons in this layer is able to approximate any continuous non-linear function with arbitrary precision (Cybenko, 1989; Funahashi, 1989; Hornik et al., 1989). Therefore, to learn the relationship between inputs and outputs when using contextual affordances we implemented a multilayer perceptron (MLP) which is a feedforward network with one hidden layer.

We encode all the variables as presented in Table 4.1 where we show a localist data representation for objects, locations, side conditions, and actions. In side conditions, letters d and c represent the fact of being *dirty* or *clean* for each part of the table. Afterward, we use this representation to design the neural model as shown in Fig. 2.9. As input, we use vectors with 20 components in order to represent the information about the current state and the action as the affordance input. The current state is represented by the first 13 components in the input vector considering the data representation for the four variables that define a state, i.e., hand object, hand position, goblet position, and side condition. The output corresponds to the effect from contextual affordances encoded as a vector with 13 components representing the next state. If the performed action leads to a failed-state, then all components of the output vector are equal to zero. The multi-layer perceptron has 30 hidden neurons with a sigmoid transfer function and the neurons in the output layer use a linear transfer function. The number of neurons selected in the hidden layer is empirically determined related to our scenario. We use Nguyen-Widrow weight initialization (Nguyen and Widrow, 1990) and the second order training method Levenberg-Marquardt Hagan and Menhaj (1994) for 100 epochs.

Table 4.1: Representation of training data used for neural network classification.

Data Representation					
Objects		**Locations**		**Side conditions**	
free	[1 0 0]	home	[1 0 0]	dd	[1 0 0 0]
sponge	[0 1 0]	left	[0 1 0]	dc	[0 1 0 0]
goblet	[0 0 1]	right	[0 0 1]	cd	[0 0 1 0]
				cc	[0 0 0 1]
Actions					
grasp	[1 0 0 0 0 0 0]	go right	[0 0 0 0 1 0 0]		
place	[0 1 0 0 0 0 0]	clean	[0 0 0 0 0 1 0]		
go home	[0 0 1 0 0 0 0]	abort	[0 0 0 0 0 0 1]		
go left	[0 0 0 1 0 0 0]				

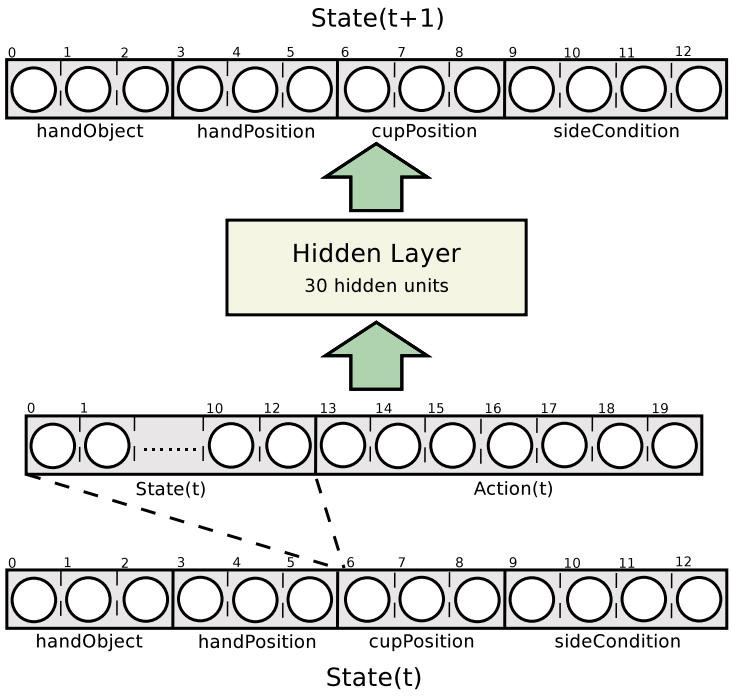


Figure 4.2: Multi-layer perceptron architecture for future state prediction. In our scenario, the next state reached by the robotic agent represents the affordance effect.

To obtain data, we initially used a previous run of autonomous RL to collect actions that lead to failed-states. The total number of data samples is 371 instances for the training of the multi-layer perceptron. The MLP is employed as an associative memory to map states, actions, and objects to the subsequent effect, as in Eq. (4.2). Therefore, the neural network is able to store not only failed-states but also transitions when an action leads to another valid state. The neural network training is carried out in an offline fashion before the IRL execution with the previously collected data. Information about failed-states may be gathered by others sources as well, like the transition function or when the trainer-agent is performing autonomous reinforcement learning. Thus, to compare fairly an affordance-driven approach to other approaches, it is necessary to have information about failed-states first, otherwise this must be learned in an online manner.

4.3.2 Interactive Reinforcement Learning Approach

Since RL is used, most of the time the robot performs actions autonomously by exploring the environment unless guidance is delivered by the previously trained robot which already has full knowledge on how to carry out the task. The apprentice robot takes advantage of this advice in these periods during a learning episode and performs the suggested actions trying to complete the task with fewer actions.

In the learning algorithm, to solve equation Eq. (2.8), we allow the robot to perform actions considering transitions from state-action pair to state-action pair rather than transitions from state to state only. Therefore, we implement the on-policy method SARSA (Rummery and Niranjan, 1994) to update every state-action value according to Eq. (4.3):

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{4.3}$$

where s_t and s_{t+1} are the current and next state respectively, a_t and a_{t+1} are the current and next action, $Q(s_t, a_t)$ is the value of the state-action pair, r_{t+1} the collected reward, α is the learning rate and γ the discount factor. The parameters used in Eq. (4.3) are empirically set to $\alpha = 0.3$ and $\gamma = 0.9$ considering values $\in (0, 1)$. Furthermore, we use the ϵ-greedy method for action selection with $\epsilon = 0.1$. Therefore, 10% of the time the agent selects an exploratory action and 90% of the

time the next action a_t is determined as shown in Eq. (4.4):

$$a_t = \underset{a \in A}{\operatorname{argmax}}\ Q(s_t, a) \tag{4.4}$$

where s_t is the current state at time t, a is an action, and A corresponds to the set of all actions. Algorithm 4.1 shows this action selection method, whereas algorithm 4.2 presents the classic RL approach where it is used as a subroutine.

Using contextual affordances we slightly modify the policy for the action selection shown in Eq. (4.4) as in the following expression:

$$a_t = \underset{a \in A_s}{\operatorname{argmax}}\ Q(s_t, a) \tag{4.5}$$

where s_t is the current state in time t, a is an action, and A_s corresponds to a subset of available actions in the current state s_t. The subset is determined based on the contextual affordances (see Eq. (4.1) and Fig. 4.1). In this regard, it is possible to anticipate the effect of performing the action with an object in a particular state. Algorithm 4.3 shows the action selection method used during RL with contextual affordances where the subset A_s is created by observing the contextual affordances in each state and populated with the actions which return a valid next state value.

Algorithm 4.1. SELECTACTION method used in the classic reinforcement learning approach

Input: Agent's current state s_t

Output: Next action a_t to perform

```
1: function SELECTACTION(s_t)
2:     if rand(0,1) < ε then
3:         a_t ← choose any random action a from A
4:     else
5:         a_t ← argmax_{a∈A} Q(s_t, a)
6:     end if
7:     return a_t
8: end function
```

We use the *advise* method parameters (Griffith et al., 2013) for interaction, i.e., probability of feedback $\mathcal{L}$ and consistency of feedback $\mathcal{C}$. Algorithm 4.4 shows the

Algorithm 4.2. Classic reinforcement learning approach with the on-policy method SARSA

Initialize $Q(s, a)$ arbitrarily
for each episode **do**
    Choose an action using $a_t \leftarrow$ selectAction(s_t)
    repeat
        Take action a_t
        Observe reward r_{t+1} and next state s_{t+1}
        Choose an action using $a_{t+1} \leftarrow$ selectAction(s_{t+1})
        $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$
        $s_t \leftarrow s_{t+1}$
        $a_t \leftarrow a_{t+1}$
    until s is terminal
end for

Algorithm 4.3. selectActionWithAffordances method used in the reinforcement learning with contextual affordances approach

Input: Agent's current state s_t

Output: Next action a_t to perform

function selectActionWithAffordances(s_t)
    Create subset A_s
    if $rand(0, 1) < \epsilon$ **then**
        $a_t \leftarrow$ choose any random action a from A_s
    else
        $a_t \leftarrow \underset{a \in A_s}{\text{argmax}}\ Q(s_t, a)$
    end if
    return a_t
end function

method used when advice is required. The higher the values of $\mathcal{C}$, the more often a good advice is given. In this context, the best advice is obtained from the subset of available actions A_s considering the highest state-action pair from the trainer-agent, whereas the worst action advised is also taken from A_s but considering the lowest state-action pair. In Algorithm 4.4, we use the worst advice instead of random advice to distinguish it with an exploratory action.

Algorithm 4.4. getAdvice method used in the interactive reinforcement learning approach with contextual affordances

Input: Agent's current state s_t

Output: Next action a_t to perform

function getAdvice(s_t)
    Create subset A_s
    if $rand(0,1) < \mathcal{C}$ **then**
        $a_t \leftarrow$ best advice from A_s
    else
        $a_t \leftarrow$ worst advice from A_s
    end if
    return a_t
end function

Algorithm 4.5. Interactive reinforcement learning approach using contextual affordances and interaction

Initialize $Q(s,a)$ arbitrarily
for (each episode) **do**
    Choose an action using $a_t \leftarrow$ selectActionWithAffordances(s_t)
    repeat
        Take action a_t
        Observe reward r_{t+1} and next state s_{t+1}
        $a_{t+1} \leftarrow$ selectActionWithAffordances(s_{t+1})
        if $rand(0,1) < \mathcal{L}$ **then**
            Change action $a_{t+1} \leftarrow$ getAdvice(s_{t+1})
        end if
        $Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)]$
        $s_t \leftarrow s_{t+1}$
        $a_t \leftarrow a_{t+1}$
    until s is terminal
end for

In our experiments we use the policy shaping method described in Sec. 2.1.8. Algorithm 4.5 shows the IRL approach using contextual affordances and interaction by means of the subroutines SELECTACTIONWITHAFFORDANCES and GETADVICE shown in algorithms 4.3 and 4.4 respectively. The conditional statement in line 8 of algorithm 4.5 represents the fact that the external trainer delivers advice and changes the next action a_{t+1} by calling the method GETADVICE where contextual affordances are used.

Each set-up was carried out 100 times using the obtained average values for the subsequent analysis. The Q-values were initialized randomly using a uniform probability distribution between 0 and 1.

4.4 Experimental Results

4.4.1 Training an Agent Using Classic RL

In this first step, simulations are performed to train the first agent with the SARSA algorithm (see Eq. (4.3)) using the reward function shown in Eq. (3.4). Due to a large number of states, and since many of them are actually failed-states, the agent needs more than 400 episodes of training to reach the final state at least once in 100 attempts. Fig. 4.3 shows the average number of actions involved in every episode until reaching the final state with green crosses. Here, the number of actions is only shown when the final state was reached; hence, in the episodes where no cross is shown, only failed-states were reached. In the first half of the training the final state was reached just a few times, but in the last part the agent becomes more successful and furthermore, in these cases close to 15 actions are being performed which in fact is the minimal number of possible actions.

The dashed green line shown in Fig. 4.3 with a different scale on the y-axis represents the percentage of successful runs. This curve is calculated by a convolution using 50 neighbors to make it smoother and we observe that the initial percentage of success is very low. Nevertheless, additional tests have shown that the curve keeps growing, although it only reaches success rates of 35%. This clearly shows the difficulty in obtaining a stable behavior by RL and the corresponding long training times. When comparing to previous work (Cruz et al., 2014), we have

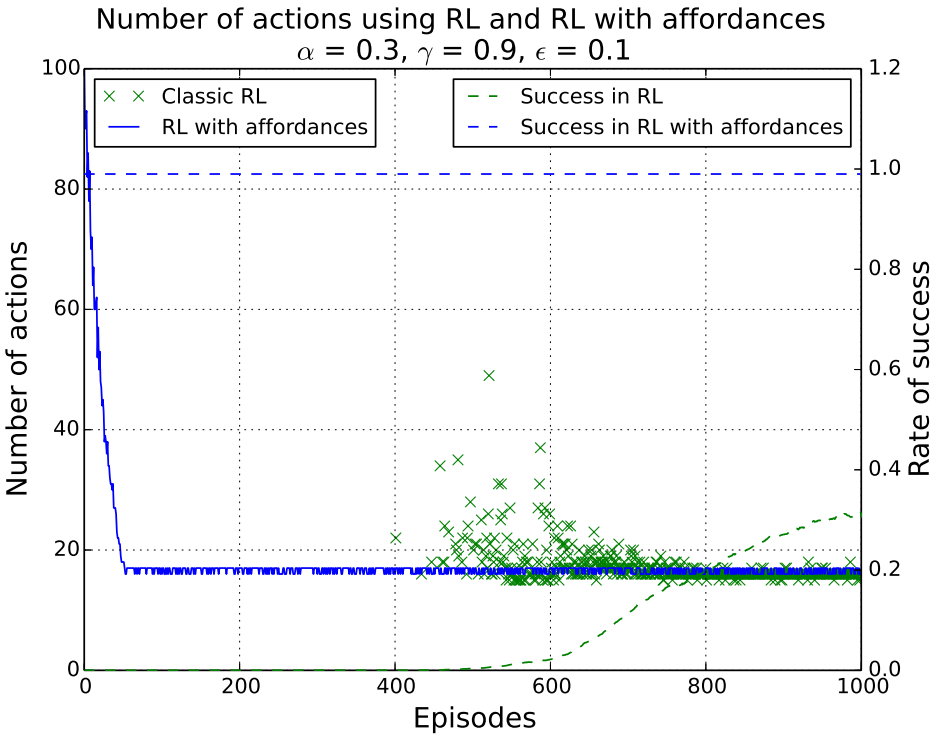


Figure 4.3: Average number of actions needed for reaching the final state for classic RL (green) and RL with contextual affordances (blue) over 100 runs in 1000 episodes. For classic RL, the average number of actions is shown as a green cross only if at least one run was successful. Dashed lines show the rate of success of runs that have reached the final state, which is always 1 in the case of RL with contextual affordances since failed-states cannot be reached anymore. The rate of success was smoothed by a moving average with window size 50.

obtained a substantial improvement. By including a small negative reward after each performed action, the robot is encouraged to choose shorter paths towards the final state. This negative reward leads to faster convergence and improved the success rate considerably from previously 4% to the level close to 35%.

The average collected reward over 100 runs in 1000 episodes is shown in Fig. 4.4. It is possible to see that the reward curve starts with values of -1 which mean that in the beginning the robot fails the task immediately and up to 600 episodes later is still failing most of the time. However, after 600 episodes the robot is able to finish the task more regularly and thus increasing its collected reward.

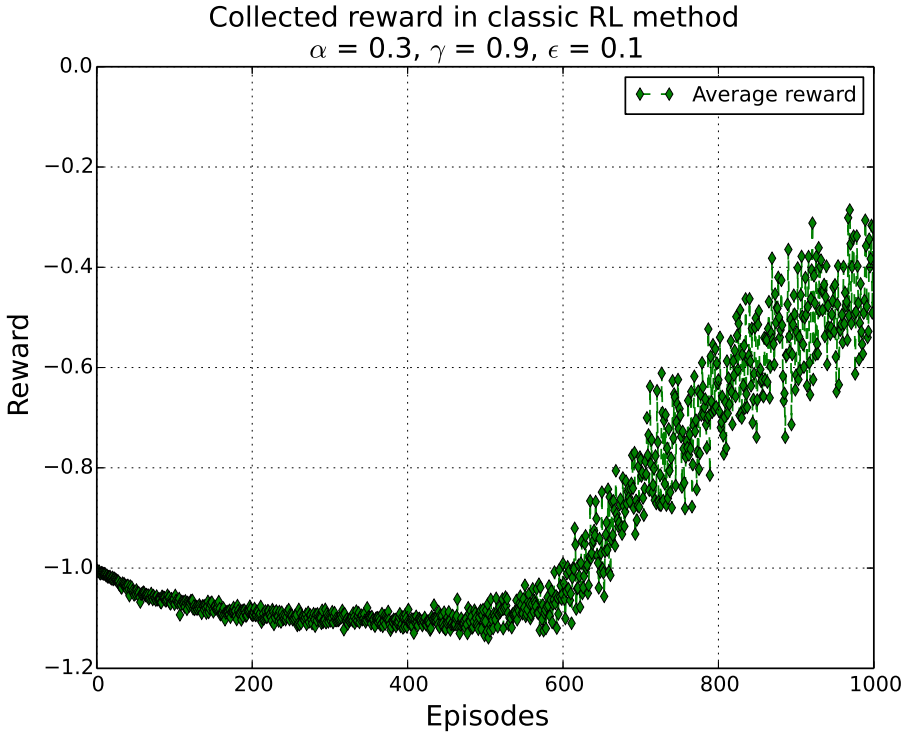


Figure 4.4: Average collected reward over 100 runs using classic RL in 1000 episodes. The collected reward starts at -1 which means the robot failed on performing the cleaning task most of the time immediately. From there onwards and until approximately 600 episodes the robot still mostly fails the task and then completes it more and more often.

4.4.2 Training an Agent Using RL with Contextual Affordances

As mentioned in the previous subsection, excessive episodes are required in order to reach a stable system. Even though this is computationally expensive it would be feasible in a simulated environment. Nevertheless, it would be unfeasible to perform these quantities of episodes in a real scenario. Therefore, we decided to explore the benefit of contextual affordances which are implemented to reduce the valid action space for the agent by avoiding failed-states. Fig. 4.3 shows the number of actions in each episode with this set-up. Using this approach, we manage to reduce the number of episodes considerably, i.e., we need less than 100 episodes to obtain a stable behavior and an average number of performed actions close to the minimum.

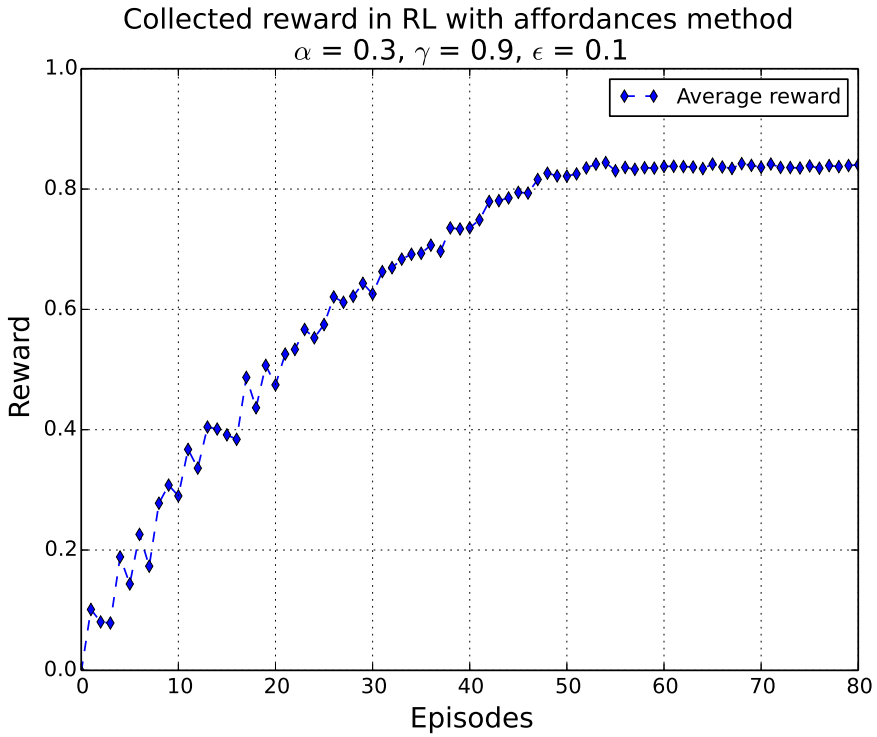


Figure 4.5: Average collected reward over 100 runs using RL with contextual affordances in 80 episodes. The collected reward starts already near to 0 since in the beginning the robot does not have knowledge on how to perform the task but the final average reward is much bigger than in the previous case since no failed-state is reached anymore.

Whereas in the classic RL method the probability of success is still low in the first episodes of training, in this set-up no episode ends in a failed-state because of the use of contextual affordances since an episode can only end when the agent reaches the final state (see dashed blue line in Fig. 4.3), which also produces considerably lower variation in comparison to the preceding method.

Fig. 4.5 shows the average collected reward over 100 runs in only 80 episodes to highlight the behavior in the first part of the training. It can be seen that the reward curve starts with values close to 0 (-0.0024 in the first episode) since in the beginning, the robot needs many intermediate actions until completing the task but around 60 episodes later the robot is able to finish the task performing a number of actions near to the minimum and increasing the average collected reward.

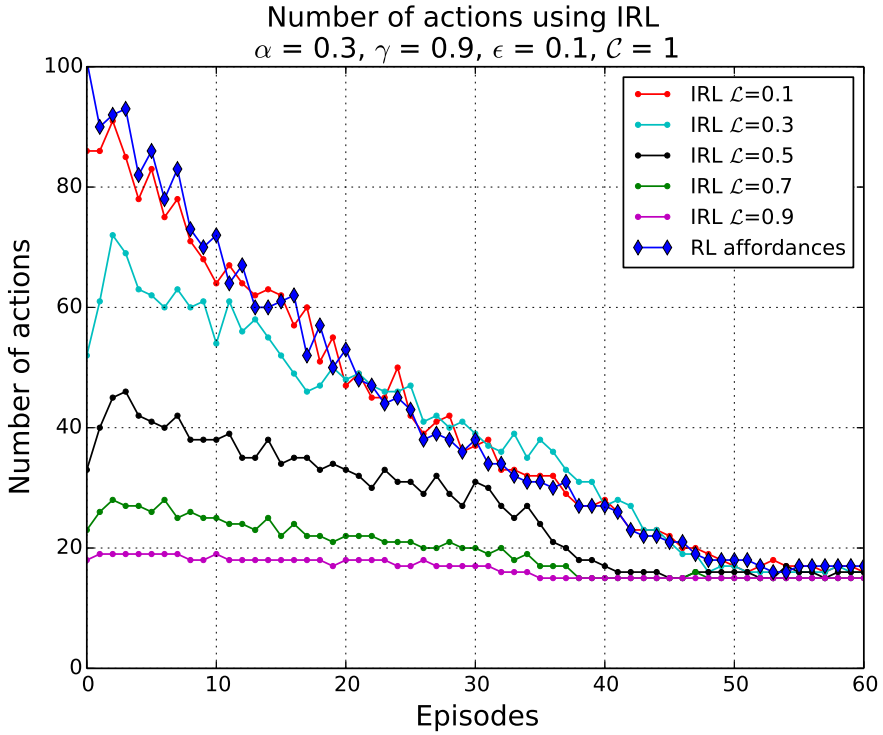


Figure 4.6: Average number of actions needed for reaching the final state for RL with contextual affordances approach (blue diamonds) and IRL approach with different probabilities of interaction $\mathcal{L}$ and a fixed probability of consistency $\mathcal{C} = 1$ over 100 runs. The agent takes advantage of probabilities of interaction as small as $\mathcal{L} = 0.3$ by reducing the total number of performed actions.

4.4.3 Training a Second Agent Using IRL with Contextual Affordances

Once a first agent has been trained, a second agent is trained with an IRL approach that allows manipulating selected actions as shown in Fig. 2.6. In this method, the external trainer that provides feedback is the trained agent which already has knowledge about the task to be performed. Furthermore, we base the interaction model on the *advise* method (Griffith et al., 2013) which uses two likelihoods, $\mathcal{C}$ to refer to the consistency of feedback which comes from an external human agent, and $\mathcal{L}$ to refer to the probability of receiving feedback, i.e., a likelihood that an external human agent delivers guidance at some point.

First, we tested diverse values for $\mathcal{L}$ and $\mathcal{C}$ to investigate the influence within the

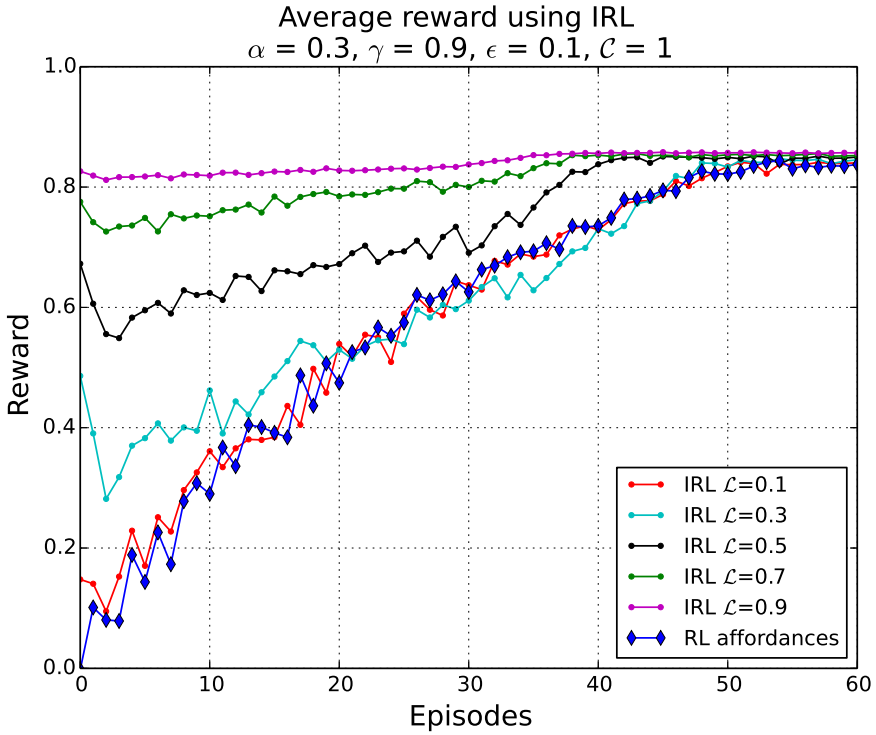


Figure 4.7: Average collected reward over 100 runs for RL with contextual affordances approach (blue line) and IRL approach with different probabilities of interaction $\mathcal{L}$ and a fixed probability of consistency $\mathcal{C} = 1$. After 50 episodes all approaches reach a reward over 0.8.

learning process in terms of performed actions and collected reward. Fig. 4.6 shows the average number of performed actions with $\mathcal{L} \in [0.1, 0.9]$ and $\mathcal{C} = 1$. As a reference, the number of performed actions with RL using contextual affordances is shown with blue diamonds which is equivalent to have $\mathcal{L} = 0$. We can see that even with a probability of feedback as small as $\mathcal{L} = 0.3$ the agent can improve its performance, especially in the first episodes. Moreover, Fig. 4.7 shows the average collected reward by the agent over episodes for different values of $\mathcal{L}$.

Afterward, we explored the learning behavior with different values for the consistency of feedback $\mathcal{C}$. The higher the consistency, the more accurate the advice which means that fewer mistakes are made during the learning process. We also use contextual affordances but in this case, the worst guidance is advised as shown in algorithm 4.4 and as a result, a bad advice is selected among the actions which still do not lead to a failed-state.

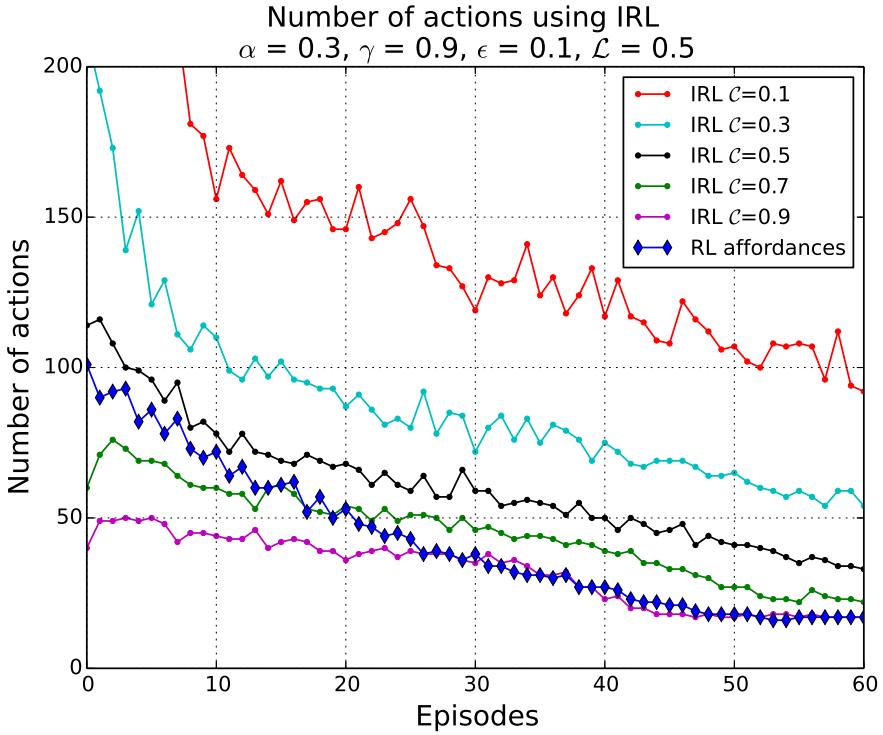


Figure 4.8: Average number of actions needed for reaching the final state for RL with contextual affordances approach (blue diamonds) and IRL approach with different probabilities of consistency $\mathcal{C}$ and a fixed probability of interactions $\mathcal{L} = 0.5$ over 100 runs. The agent takes advantage of probabilities of interaction larger than $\mathcal{C} = 0.5$ by reducing the total number of performed actions as small as RL with contextual affordances approach.

To investigate the consistency of feedback $\mathcal{C}$, we fixed the average probability of feedback to $\mathcal{L} = 0.5$ and then performed experiments with different consistency $\mathcal{C} \in [0.1, 0.9]$. Fig. 4.8 depicts the results and also shows the number of actions for RL with contextual affordances which are equivalent to have $\mathcal{L} = 0$ and $\mathcal{C} = 0$. It is clear that a more consistent trainer leads to better results or in this context to perform fewer actions. However, even with a consistency of $\mathcal{C} = 0.5$ the agent can improve its performance over time, especially in the beginning of the training. Additionally, Fig. 4.9 shows the average collected reward by the agent over episodes for different values of $\mathcal{C}$.

Finally, we ran an additional experiment where we decreased the probability of feedback $\mathcal{L}$ and therefore the contribution of advice over time to simulate the

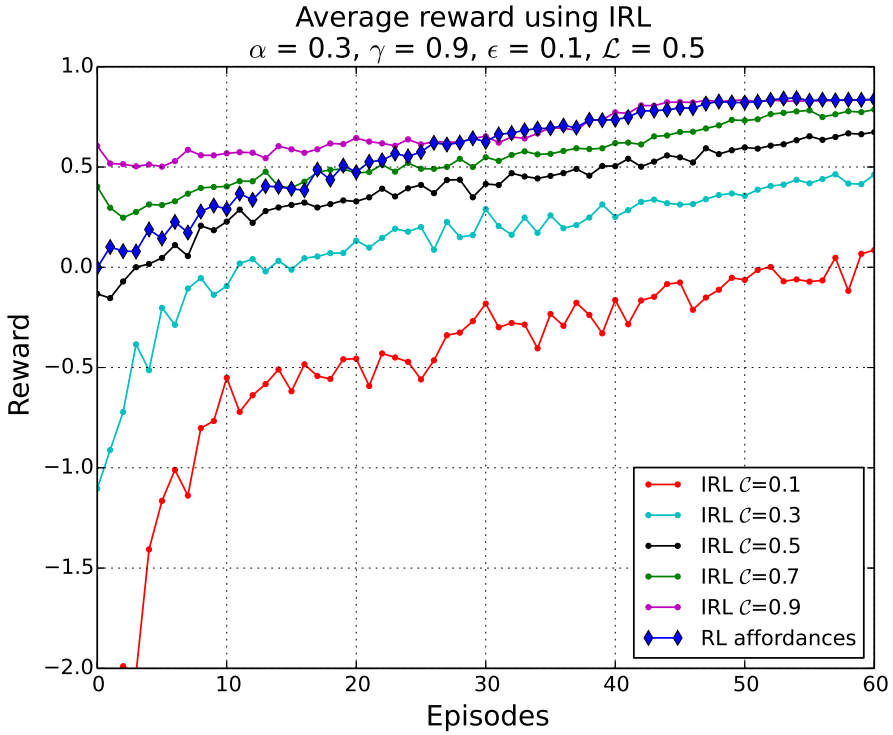


Figure 4.9: Average collected reward over 100 runs for RL with contextual affordances approach (blue line) and IRL approach with different probabilities of consistency $\mathcal{C}$ and a fixed probability of interaction $\mathcal{L} = 0.5$. The final reward in all cases is less than RL with contextual affordances approach, nevertheless with probabilities of consistency over 0.5 is observed a similar behavior than a perfect trainer.

fatigue of an external trainer to provide feedback during the whole learning process. We made a reduction of the feedback after every episode as:

$$\mathcal{L}_{t+1} = \eta \mathcal{L}_t \tag{4.6}$$

starting from different initial values of $\mathcal{L}$ and with $\eta = 0.95$ for all cases. Fig. 4.10 shows the average number of actions performed in every case. It is possible to observe that as interaction is decreasing, the number of performed actions increases after the first episodes where the agent explores non-optimal actions in the absence of guidance. Nevertheless, after 25 episodes even with a very low amount of interaction the agent is able to reduce the number of performed actions due to its own knowledge on how to perform the cleaning task.

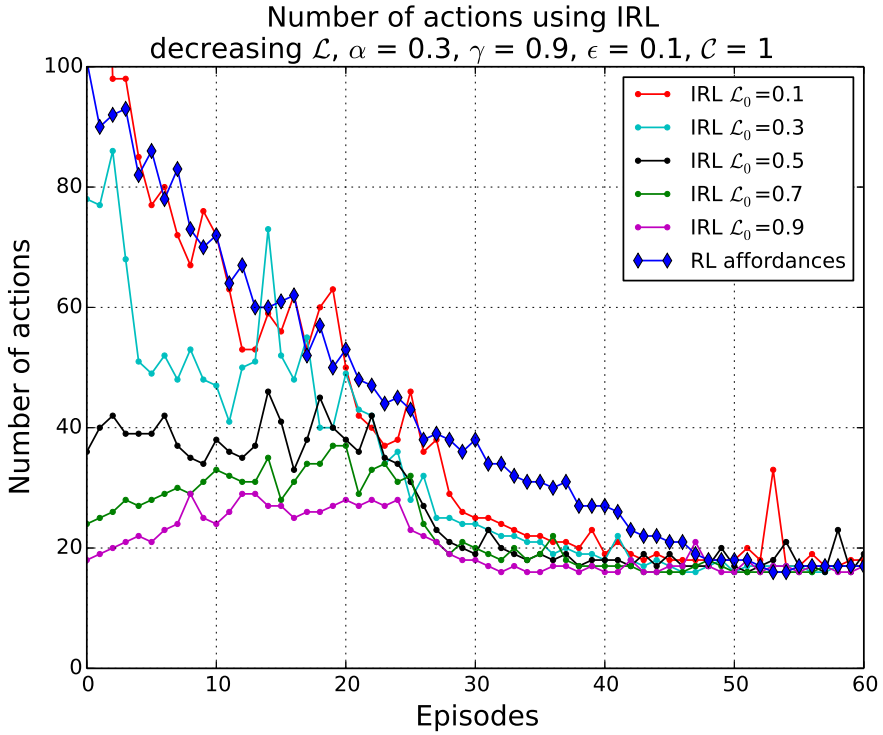


Figure 4.10: Average number of actions needed for reaching the final state for RL with affordances approach (blue diamonds) and IRL approach with different initial probabilities of interaction $\mathcal{L}_0$ and decreasing over time.

4.5 Discussion

In this chapter, three particular learning methods were realized to test the performance. The first method consisted of a robotic agent learning to execute the cleaning task in an autonomous fashion using classic RL. The agent was not able to learn the task before 400 episodes and still with a low success rate which increased slowly to 35% in episode number 1000. Furthermore, collected reward decreased in the first 600 episodes because of the low success rate and from there onwards it increased to values around -0.4 showing that the agent was able to learn slowly.

In the second method using RL with affordances, the robotic agent also learned the task in an autonomous fashion but this time utilizing contextual affordances to avoid failed-states. The agent mastered the task faster in comparison to the method used previously, reducing the number of actions needed to complete the task from 100 actions in the beginning to fewer than 20 actions within 100 episodes.

Furthermore, with this method collected reward was always positive, reaching values over 0.8 because the success rate, in this case, is always 100% as a result of the usage of contextual affordances.

The third method consisted of IRL with affordances, and in this method, the robotic agent which had previously learned to execute the task became the trainer of a second agent. In this scheme, the second robot was the learner-agent which was advised in certain periods of the training process by the trainer-agent which had acquired knowledge on how to perform the cleaning task.

Training robotic agents with interactive feedback and contextual affordances presented an advantage over classic RL in terms of the number of performed actions and collected reward. Even low levels of interaction showed progress in comparison to RL working without an external trainer. Moreover, the agent was able to learn the proposed cleaning task even when being misadvised or receiving inconsistent feedback in some time steps during the learning process.

The applicability of the proposed method in more realistic scenarios is an open question to be addressed, therefore, we will transfer the present set-up to a human-robot interaction scenario, where advice is given by human trainers who must not necessarily be experts on developmental robotics or machine learning. In this regard, human advice can be interpreted as parental scaffolding (Ugur et al., 2015) and therefore it is interesting to investigate it in terms of the number of given instructions and the frequency of these.

To transfer this scenario to real environments in a more plausible manner more advanced architectures have to be developed considering also different modalities (Farkaš et al., 2012), e.g., the audio modality using a microphone or the vision modality using a depth sensor, to make it more realistic and integrate it in a multi-modal system to control the robot interactively. This enables to get much closer to real environments in which any human trainer even without a background in robotics can teach a robot. This kind of set-up will be deeper developed and described in chapters 6 and 7 within Part III. Human-Agent Interactive Reinforcement Learning.

While this chapter has shown how an affordance-based model, so-called contextual affordance, may benefit the IRL framework as a tool to reduce the search space during the apprenticeship process, we have also investigated parameters of interaction as frequency and consistency of feedback. The following chapter will focus on the features of trainer-agents and bring additional details about agent-agent interaction to deeper discuss what makes a good teacher, considering the parameters of interaction and learner-agent's obedience. Subsequently, in the following chapters, we will focus on whether the theoretical, idealized parameters can be transferred to human teachers.

Chapter 5

Influence of Different Trainer Types on Learner-Agents

5.1 Introduction

Interactive reinforcement learning (IRL) has become an important apprenticeship approach to speed up convergence in classic reinforcement learning problems (Thomaz and Breazeal, 2007; Knox et al., 2013b; Taylor et al., 2014). In this regard, a variant of IRL is policy shaping which uses a parent-like trainer to propose the next action to be performed and by doing so reducing the search space by advice. On some occasions, as shown in the previous chapter, the trainer may be another artificial agent which in turn was trained using reinforcement learning methods to afterward becoming an advisor for other learner-agents.

In this chapter, we not only study this situation, i.e., utilizing artificial trainer-agents, but rather assessing teacher performance over learner-agents. Initially, we look into the internal representation and visited states of prospective advisor agents in order to explore which features may be important to act as a good trainer. Afterward, we compare the behavior of both the advisor and the learner in terms of the internal representation, visited states, and collected rewards. Finally, we evaluate the system interaction parameters along with the learner behavior in terms of learner-agent's obedience. We hypothesize that the inclusion of this additional parameter may be beneficial in presence of a low consistency of feedback

allowing to ignore bad advice. Experiments and results in this chapter have been designed and obtained in order to answer the second research question posed in this thesis: What makes a good teacher-agent when considering internal knowledge representation and interaction parameters?

By using artificial agents as teachers, some properties have been studied so far such as different effects of delivering advice in different episodes and with different strategies during the learning process (Torrey and Taylor, 2013; Taylor et al., 2014) and effects of different probabilities and consistency of feedback (Griffith et al., 2013; Cruz et al., 2014, 2016a) as shown in the previous chapter. Nonetheless, the implications of utilizing artificial teachers with different characteristics and different internal representations of the knowledge based on their previous experience have not been studied. Moreover, the effects when the learner ignores some of the advice have also not been studied in artificial agent-agent interaction, although some insights are given in Griffiths' work using human-human interaction with a computational interface (Griffiths et al., 2012).

We study effects of agent-agent interaction in terms of achieved learning when parent-like teachers differ in essence and when learner-agents vary in the way they incorporate the advice. Therefore, we analyze internal representations and characteristics of artificial agents to determine which agent may outperform others to become a better trainer-agent. We hypothesize that certain agents, acting as advisers, may lead to a larger reward and faster convergence of the reward signal and also to a more stable behavior in terms of the state visit frequency of the learner-agents. Moreover, we further analyze system interaction parameters in order to determine how influential they are in the apprenticeship process, where the consistency of feedback may be much more relevant than other parameters when dealing with different learner obedience parameters.

5.2 Interactive Reinforcement Learning with Artificial Trainers

As aforementioned, RL has been used to allow robotic agents to autonomously explore their environment in order to develop new skills (Wiering and Otterlo,

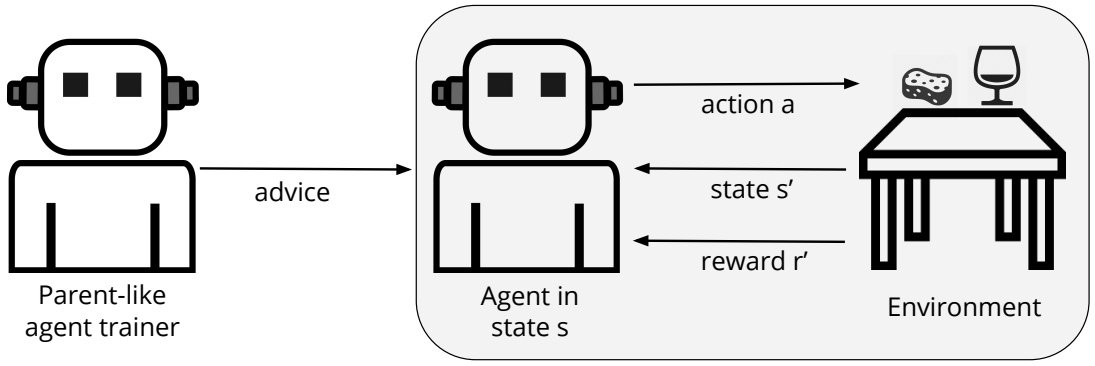


Figure 5.1: An interactive reinforcement learning approach with policy shaping. In selected states, the trainer advises the learner-agent changing the action to be performed in the environment.

2012; Mnih et al., 2015). The gray box in Fig. 5.1 shows the general description of the RL framework, with the environment represented by domestic objects which are related to our scenario which is described in the Sec. 3. Furthermore, Fig. 5.1 shows a general overview of the agent-agent scheme where the trainer provides advice in selected episodes to the learner-agent to bootstrap its learning process.

Even though in an IRL scenario reward and policy shaping are alternatives to train a learner-agent, in a domestic scenario, a robotic agent is expected to work with humans as external trainers. However, there exist asymmetries when humans quantify a reward including sometimes feedback about the past actions and also about actions they predict the robot will do (Thomaz and Breazeal, 2006, 2007). Hence, we decided to use the policy shaping method with artificial trainer-agents in this chapter, shown in Fig. 2.6. To this aim, in the cleaning-table scenario a learner-agent has previously been trained using classic RL and then this learner becomes the external trainer. Therefore, this agent has full knowledge of all possible actions and has to deliver it to a second agent which is thus trained with IRL. We use an artificial trainer to have better control over the feedback compared to experiments with a human trainer. Nevertheless, diverse information sources can be employed to obtain feedback from, for instance, a person (as shown in the next chapters), another robot, or any other artificial system.

Although interactive advice improves the learning performance of learner-agents, something that may significantly affect the agent's performance is the need of a

good trainer since consecutive mistakes may lead to a worse training time as shown is the previous chapter. In principle, one may think that the agent with the largest accumulated reward should be a good candidate to become the trainer. However, when we look into the internal knowledge representation this may not necessarily be the best option. In some occasions, agents with lower overall performance may be better trainers due to a possibly vast experience about less common states (i.e., states that not necessarily lead to the optimal performance) and therefore may give better advice in those states.

For a good trainer to emerge with knowledge of most of the situations or in all possible states we suggest an agent with small standard deviation σ_s from the mean frequency over all visited states which represents a better distribution of the experience during the training. Therefore, we select the trainer-agent T^* computing:

$$T^* = \underset{i \in Ag}{\operatorname{argmin}}\ \sigma_s^i \tag{5.1}$$

where Ag is the set of all the trained agents and σ_s^i the standard deviation of the visited states during the learning process for the agent i.

To test the implemented methods in this chapter, we use the previously defined RL scenario in Sec. 3. However, with the aim of comparing pure RL agents serving as trainers and test other variables, we do not include here contextual affordances and, therefore, we do not have to previously learn them, which results in a shorter training time, in general.

5.3 Experimental Set-up and Results

In the following section, we show the performed experiments and the obtained results. All experiments include the training of 100 agents through 3000 episodes. Q-values are randomly initialized using a uniform distribution between 0 and 1. Other parameter values selected after performing a grid search are learning rate $\alpha = 0.3$ and discount factor $\gamma = 0.9$. Besides this, we use ϵ-greedy action selection with $\epsilon = 0.1$. To assess the interaction between learner and trainer-agents we initially used a probability of feedback $\mathcal{L} = 0.25$; nevertheless, we afterward vary this parameter along with the consistency of feedback and learner behavior.

All the aforementioned parameters were empirically determined and related to our scenario.

5.3.1 Choosing an Advisor Agent

To acquire a sample of trainer-agents, autonomous RL was performed with 100 agents, each of them a prospective trainer for the IRL approach. In the presented scenario, there are agents with diverse behaviors which differ mostly in the path they choose until reaching a final state in the cleaning-table scenario, where there are 2 possible paths towards a final state. First, there are agents which most of the time choose the same path to complete the task, either path A or path B (see Fig. 3.2), which leads to a biased behavior due to the way the knowledge is acquired during the learning process. From this kind of behavior and taking into account our scenario, there exist agents that regularly take the shorter path (path A) and others that take the longer one (path B), we refer to them as the specialist-A and the specialist-B agents respectively. In both cases, agents successfully accomplish the task, although they accumulate different amounts of average reward. Obviously, specialist-A agents are the ones with better performance in terms of collected reward since fewer state transitions are needed to reach the final state. Secondly, there are agents with a more homogeneously distributed experience, meaning that they do not have a favorite sequence to follow and have equally explored both paths. We refer to such agents as polymath agents.

To illustrate this, Fig. 5.2 shows a frequency histogram of visited states for two potential trainer-agents over all training episodes. The histogram shows two distinct distributions, one for a specialist-A agent in gray and one for a polymath agent in blue. The specialist-A agent decided to clean the table following the shorter path most of the time and, therefore, there is an important concentration of visits among the states from 16 to 29 which are intermediate states to complete the task on this path. Furthermore, there is a clear subset of states which was never visited during the learning. In contrast, the polymath agent visited all the states and transits on both paths to a similar extent. In the case of the specialist-B agent, there is also a concentration of visits among a subset of states, similarly to the specialist-A agent. The specialist-B agent decided most of the time to clean following the longer path along the states from 30 to 48 and barely visiting states

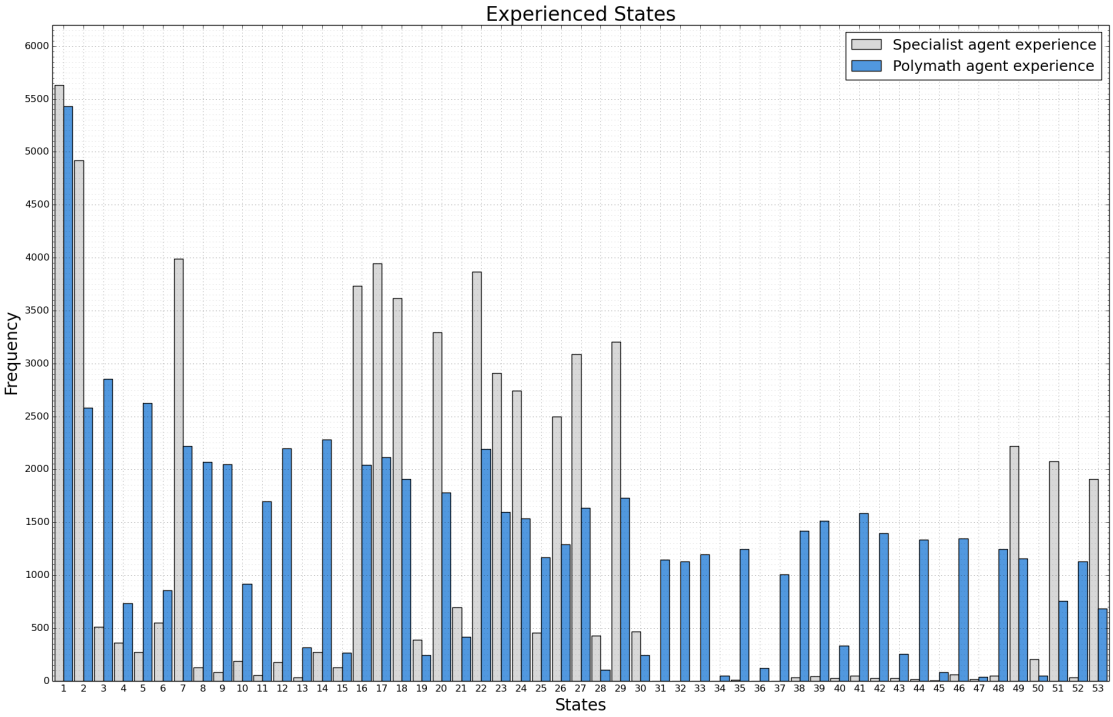


Figure 5.2: Frequencies of visits per states for two agents. It is possible to observe two different behaviors. The biased (specialist-A) agent gained experience mostly on the shorter path, whereas the homogeneously-distributed (polymath) agent gained experience through most states.

from 16 to 29. Therefore, we do include this agent in the results hereafter but we do not present it in some plots to make the relevant information more accessible.

To further analyze the agents' behavior we took three representative agents, one per class, that we will from now on use with the respective names: A specialist-A agent with biased behavior for the shorter path, a specialist-B agent with biased behavior for the longer path, and one polymath agent with unbiased behavior. The specialist-A agent visited each state with an average of $s_1 = 1121.21$ times, a standard deviation of $\sigma_s^1 = 1570.75$, an accumulated average reward of $r_1 = 0.11105$ per episode, and $R_1 = 333.15$ during the whole training. The specialist-B agent visited each state on average $s_2 = 1561,15$ times obtaining more diverse experience than the previous agent but certainly not homogeneously distributed, which can also be seem in the standard deviation of $\sigma_s^2 = 1628.70$. The specialist-B agent accumulated an average reward of $r_2 = -0.17839$ for each episode and a total of $R_2 = -535.18$. In the case of the polymath agent, each state was visited

Table 5.1: Visited states, standard deviation, reward accumulated per episode, and total collected reward for three agents from classes with different behavior. The agents show different properties as result of the autonomous learning process.

Agent	$\boldsymbol{s}$	$\boldsymbol{\sigma_s}$	$\boldsymbol{r}$	**R**	**Properties**
Specialist-A agent	1121.21	1570.75	0.11105	333.15	Largest accumulated reward
Specialist-B agent	1561.15	1628.70	-0.17839	-535.18	Largest amount of experience
Polymath agent	1307.51	947.96	-0.00427	-12.82	Smallest standard deviation

an average of $s_3 = 1307.51$ times with standard deviation of $\sigma_s^3 = 947.96$. The accumulated average reward was $r_3 = -0.00427$ per episode and the total reward was $R_3 = -12.82$ during the whole training. Table 5.1 shows a summary of the performance of the three aforementioned agents. The table shows that the specialist-A agent accumulated the largest reward, the specialist-B agent visited more states, and the polymath agent obtained the smallest standard deviation.

Nevertheless, accumulating plenty of reward does not necessarily lead to becoming a good trainer, in fact it only means that the agent is able to select the shorter path most of the time from the initial state, but the experience collected in other states not involved in that route is absent or barely present and therefore such an agent cannot give good advice in those states where it does not know how to act optimally.

Therefore, as shown in Eq. (5.1), we propose that a good trainer is, in essence, an agent which not only collects more reward but also with fairly distributed experience. From the three agents shown above, the polymath agent has a standard deviation of $\sigma_s = 947.96$ and thus might be a good advisor. In Fig. 5.2 the experience distribution of such an agent is shown in blue, which suggests that the agent has the knowledge to advise what action to perform in most of the states. In the case of the initial state, the frequency is much higher in comparison since this state is visited every time at the beginning of a learning episode. In fact, similar

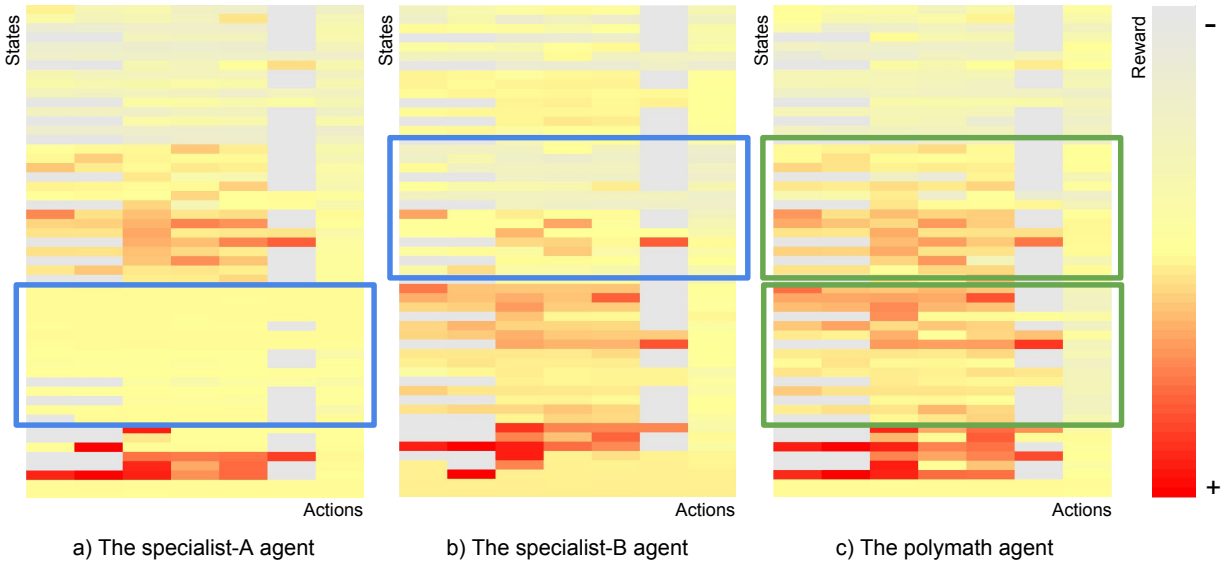


Figure 5.3: Internal knowledge representation for three possible parent-like advisors, namely the specialist-A, the specialist-B, and the polymath agent. The specialist-A agent shown in figure a), despite collecting more reward, does not have enough knowledge to advise a learner in every situation represented by the blue box. A similar situation is experienced in the specialist-B agent, shown in figure b). The polymath agent shown in figure c) has overall much more distributed knowledge which allows it to better advise a learner-agent.

frequencies are observed in this state for a biased distribution.

We also recorded the internal representation of the knowledge through the Q-values to confirm the lack of learning in a subset of states. Fig. 5.3 shows a heat map of the internal Q-values of three agents, the specialist-A, the specialist-B, and the polymath agent. Warmer regions represent larger reward and colder regions lower values. In fact, the coldest regions are associated with failed-states from where the agent should start a new episode, obtaining a negative reward of $r = -1$ according to Eq. (3.4). In Fig. 5.3 can be observed that the specialist-A agent may be an inferior advisor since there exists a whole region uniformly yellow, which shows no knowledge about what action to prefer. In the case of the specialist-B agent, there exists a region which shows much less knowledge on what action to prefer when comparing with the two other agents. In other words, the learned policies are partially incomplete as highlighted by the blue boxes in Fig. 5.3. On the contrary, the policy learned by the polymath agent is much more complete when observing

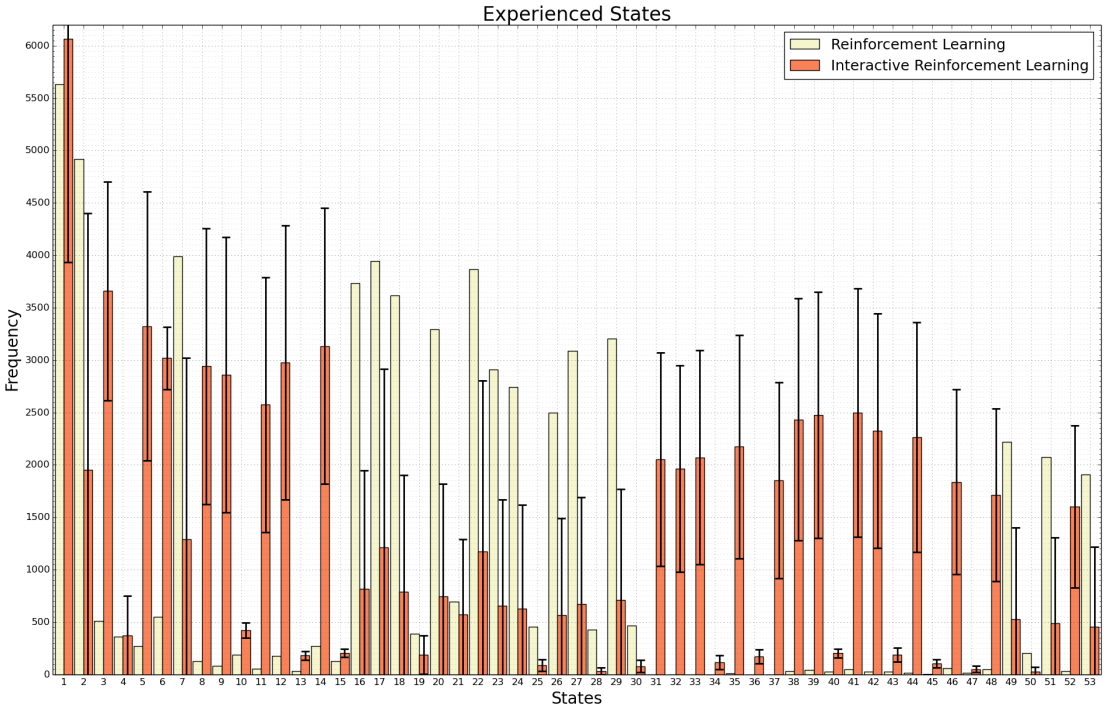


Figure 5.4: Visited states for the specialist-A RL trainer-agent and average state visits of IRL learner-agents. The averaged frequency for IRL agents moreover includes the standard deviation for visited states showing that in many cases the trainer-agent does not know how to advise and in consequence leads the learner-agent to dissimilar behavior.

the same regions as highlighted by the green boxes. It is important to note that the region on top is in all cases colder than the rest because it is the most distant one from the final states where a positive reward $r = 1$ is given, but in spite of that, the polymath agent is still able to select a suitable action according to the learned policy.

5.3.2 Comparing Advisor and Learner Behavior

Once we have chosen trainer-agents, we are able to compare how influential such a trainer is in the learning process of a learner. We use two agents shown in the previous subsection, the specialist-A and the polymath agent, the former with the largest accumulated reward and the latter with the smallest standard deviation.

Fig. 5.4 shows the frequency with which each state was visited for 100 learner-

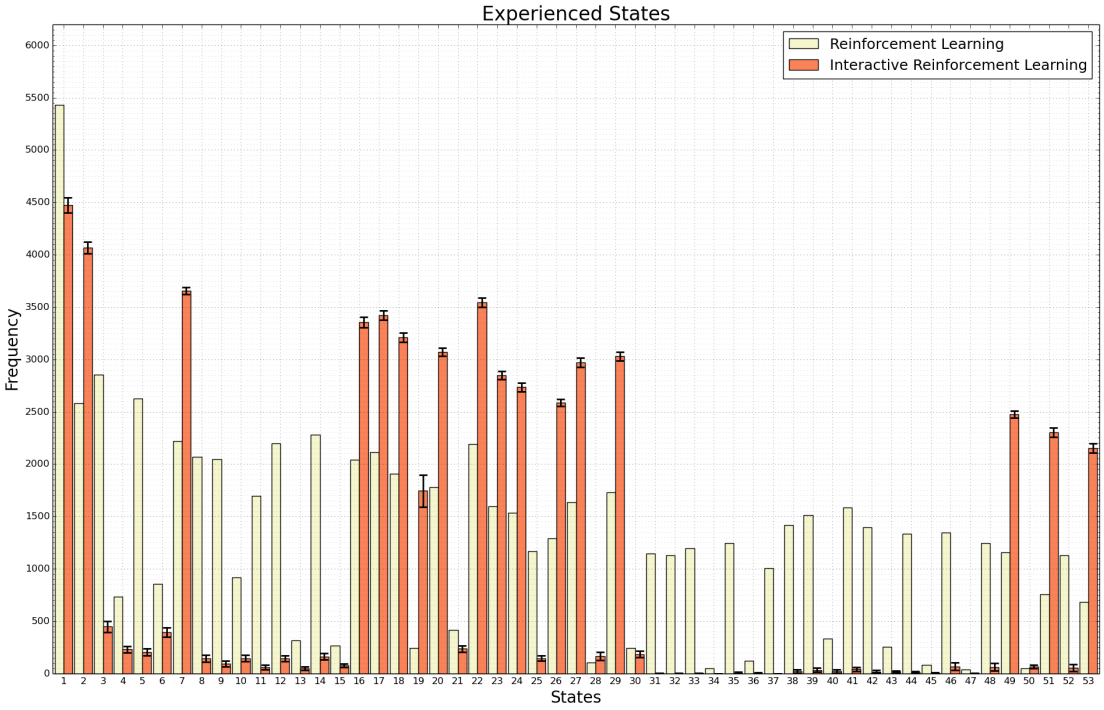


Figure 5.5: Visited states for the polymath RL trainer-agent and average state visits of IRL learner-agents. The averaged frequency for IRL agents includes the standard deviation which in this case is considerably lower as the learners are assisted by a trainer with more knowledge about the task-space which also leads learner-agents to have more stable behavior as they are consistently advised.

agents in average using the specialist-A agent with biased frequency distribution as a trainer. We can observe a large standard deviation for visited states in IRL agents in most of the cases, which suggests diversity in terms of frequency for those states among the learner-agents. Fig. 5.5 shows the average frequency of visits for each state for 100 learner-agents using the polymath agent as a trainer which has a more homogeneous frequency distribution. It can be observed that the standard deviation for visited states in IRL agents is much lower in comparison to the previous case. This shows a more stable behavior in terms of visiting frequency in learner-agents when using the polymath trainer-agent.

By using the specialist-A agent as a trainer in our IRL approach the average collected reward is slightly higher in comparison to autonomous RL. In general, the IRL approach collects the reward faster than RL but in a similar magnitude after 400 episodes. Fig. 5.6 depicts the average collected reward during the first 500

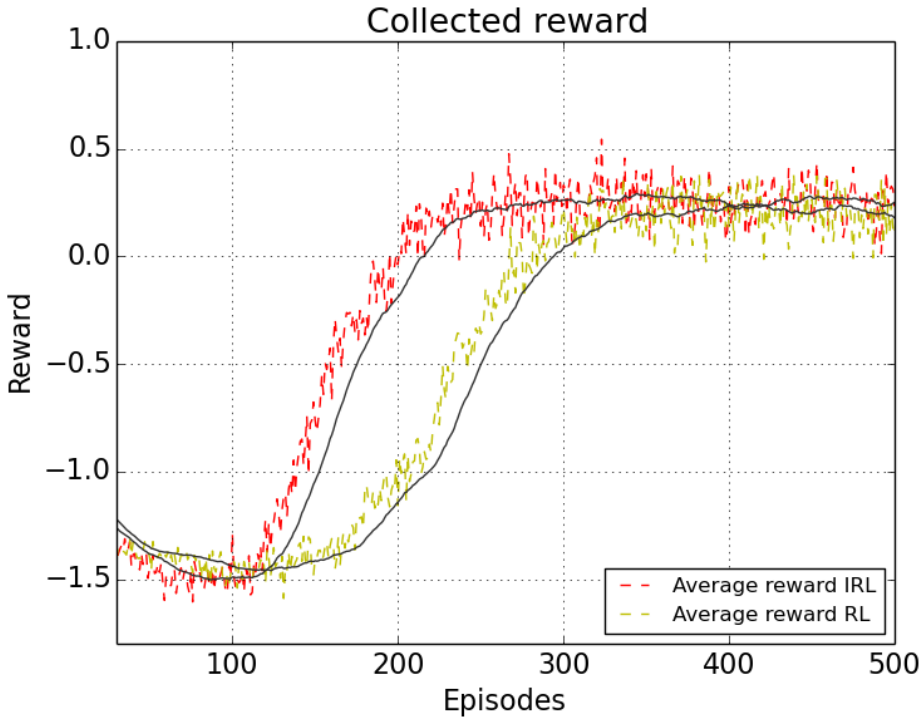


Figure 5.6: Average collected reward by 100 agents using RL and IRL approaches. In this case, a biased trainer (the specialist-A agent) is used to advise the learner-agents. The advice slightly improves the performance in terms of accumulated reward and convergence speed. The gray curves show the convoluted collected reward inside of a window of 30 values to smooth the results shown.

episodes using autonomous RL and IRL approaches with yellow and red respectively using the specialist-A agent as the trainer in the case of IRL. The gray curves show the convoluted collected reward inside of a window of 30 values to smooth the results shown.

On the other hand, by using the polymath agent as the trainer the IRL approach converges both faster and to a higher amount of reward when compared to the previous case. This is due to the polymath agent which knows the task-space better and is able to advise correctly in more situations than the specialist agent. In consequence, this allows the learner to complete the task faster and therefore accumulate more reward since the learner-agent receives less incorrect advice from the trainer-agent, i.e., with a higher consistency of feedback. Fig. 5.7 shows the average collected reward in 500 episodes for RL and IRL approaches. Once again,

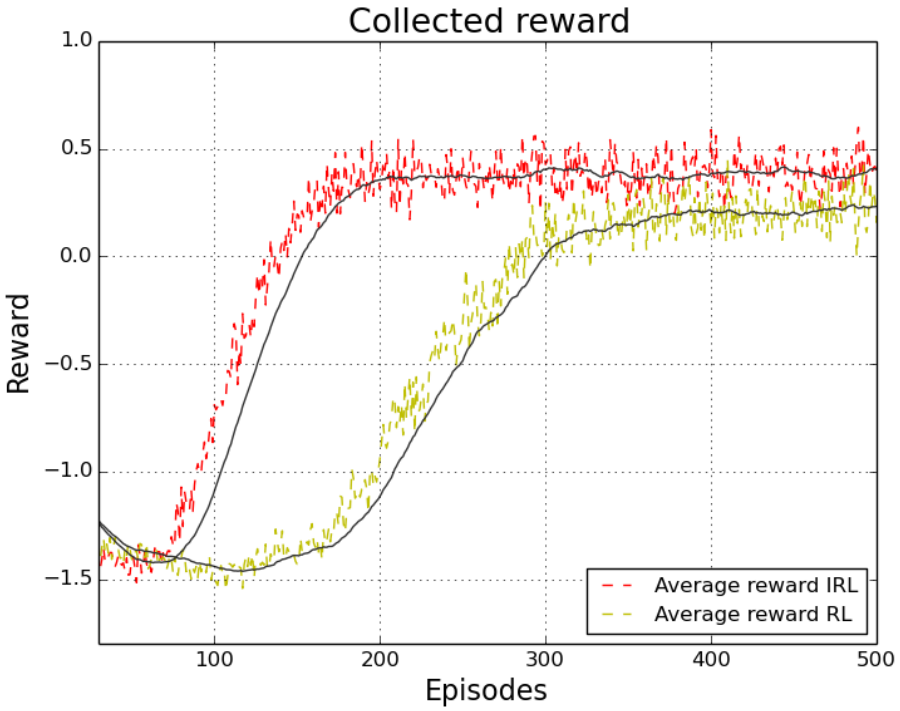


Figure 5.7: Average collected reward by 100 agents using RL and IRL approaches. When using an unbiased trainer-agent (the polymath agent) the accumulated reward is higher and the convergence speed faster in comparison to the previous case using a biased agent as an advisor. The gray curves show the convoluted collected reward inside of a window of 30 values to smooth the results shown.

the gray curves show the convoluted collected reward inside of a window of 30 values to smooth the results shown. In the following experiments, only smooth curves will be used to simplify the analysis of the results.

5.3.3 Evaluating Interaction Parameters

As shown in previous section, IRL is in general beneficial for a learner-agent in terms of accumulated reward and convergence speed. Nevertheless, the selection of the trainer can have significant implications on learner's performance. To further study the trainer-learner interaction, we evaluated the involved interaction parameters along with the effect whether the learner follows the received advice or not in order to mimic actual human-human behavior where the learner occasionally does not follow the advice (Griffiths et al., 2012). The inclusion of this parameter may

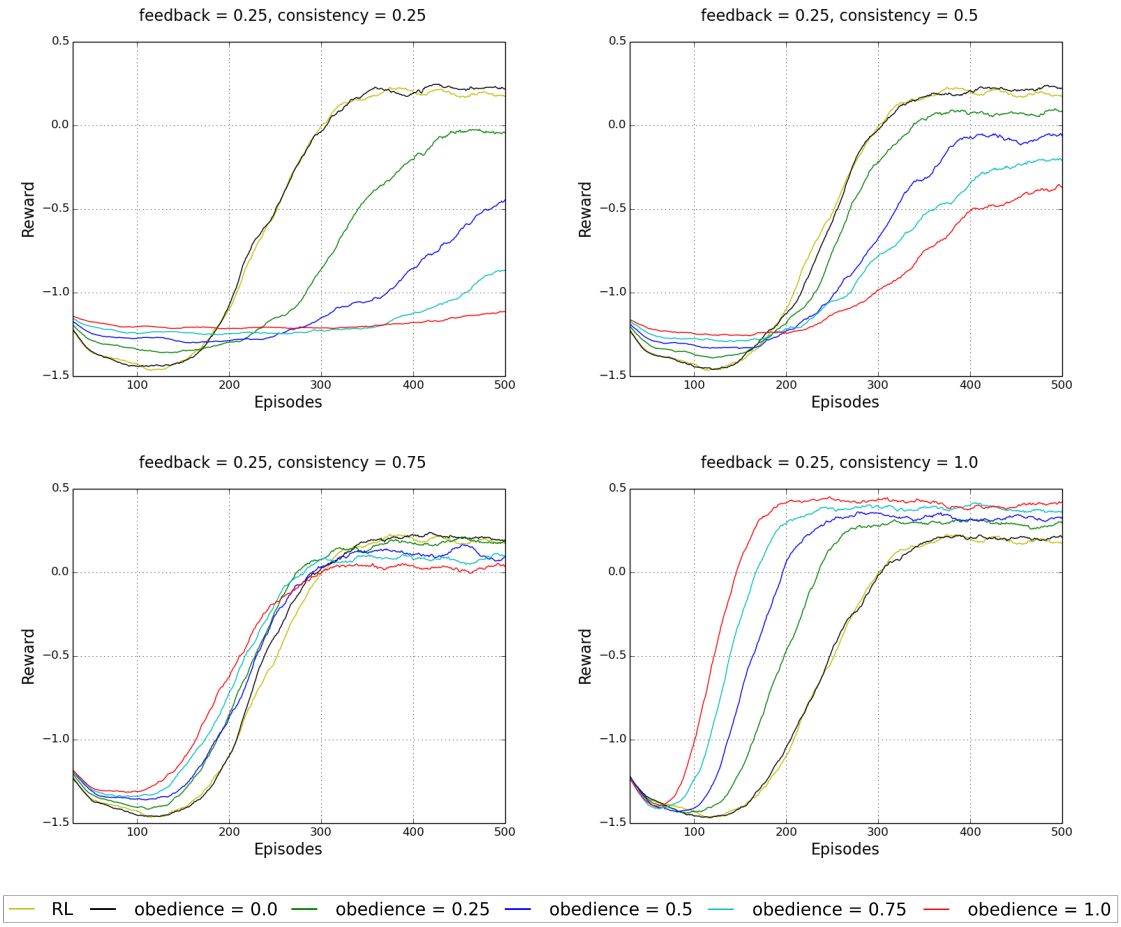


Figure 5.8: Collected reward for different values of learner obedience using fixed probability of feedback of 0.25 and four different values for consistency of feedback between 0.25 and 1.0.

be beneficial in presence of a low consistency of feedback allowing to ignore bad advice. We called this parameter *learner obedience* $\mathcal{O} \in [0, 1]$, 0 meaning that the learner-agent never follows the advice and thus corresponds to a pure autonomous RL learner.

Initially, we used a fixed probability of feedback $\mathcal{L} = 0.25$. The idea then was to test the system over a number of different values of consistency of feedback and learner obedience. Fig. 5.8 shows the collected reward during 500 episodes for the different values of consistency of feedback $\mathcal{C} \in \{0.25, 0.5, 0.75, 1.0\}$ and learner obedience $\mathcal{O} \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$. In all cases, the learner obedience $\mathcal{O} = 0$, shown in black, corresponds to autonomous RL which is shown in yellow. The collected

rewards indicate generally that the more consistent the feedback was, the better the performance. Even though that difference in the performance seems to be intuitive, it is important to note that even with comparatively high values of consistency like $\mathcal{C} = 0.75$ the learner does not achieve significantly better performance compared to autonomous RL while on the other hand, an idealistic perfect consistency ($\mathcal{C} = 1$) allows the learner-agent to achieve much higher collected reward than autonomous RL even when the learner obedience is as low as $\mathcal{O} = 0.25$. Therefore, in the current scenario, wrong advice has an important negative effect since it does not only lead to the execution of more intermediate steps but also, in many cases, leads to failed-states and thus to a high negative reward (-1) and the start of a new learning episode.

In Fig. 5.8, agents which follow the advice only 25% of the time ($\mathcal{O} = 0.25$), depicted in green, show much better performance when the consistency of feedback $\mathcal{C}$ is lower which is due to the agent being able to ignore the suggested wrong advice and select an action on its own. On the contrary, agents which follow the advice all the time ($\mathcal{O} = 1.0$), depicted in red color, show much better performance in presence of consistent feedback.

Thereupon, we modified the probability of feedback for the purpose of testing how influential different consistencies of feedback $\mathcal{C}$ and different learner obedience levels $\mathcal{O}$ are. Fig. 5.9 shows the accumulated reward during 500 episodes for probability of feedback $\mathcal{L} \in \{0.5, 0.75, 1.0\}$ (the outcome using probability of feedback of 0.25 is already shown in Fig. 5.8) and consistency of feedback $\mathcal{C} \in \{0.25, 0.5, 0.75, 1.0\}$ using learner obedience $\mathcal{O} \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$.

In Fig. 5.9 the columns show the performance over different probabilities of feedback, while the rows show the performance over different values of consistency. Observing each row, it can be seen that higher probabilities of feedback do not considerably improve the outcomes in terms of the collected reward, suggesting that often interactive feedback does not necessarily enhance the overall performance but it is rather the consistency of feedback that makes prominent differences. In fact, observing the outcomes down the columns, thus with the same probability of feedback, different values of consistency lead to significant improvements in the collected reward and consequently, consistency of feedback has much more impact on the final learning performance. For instance, when using the consistency of

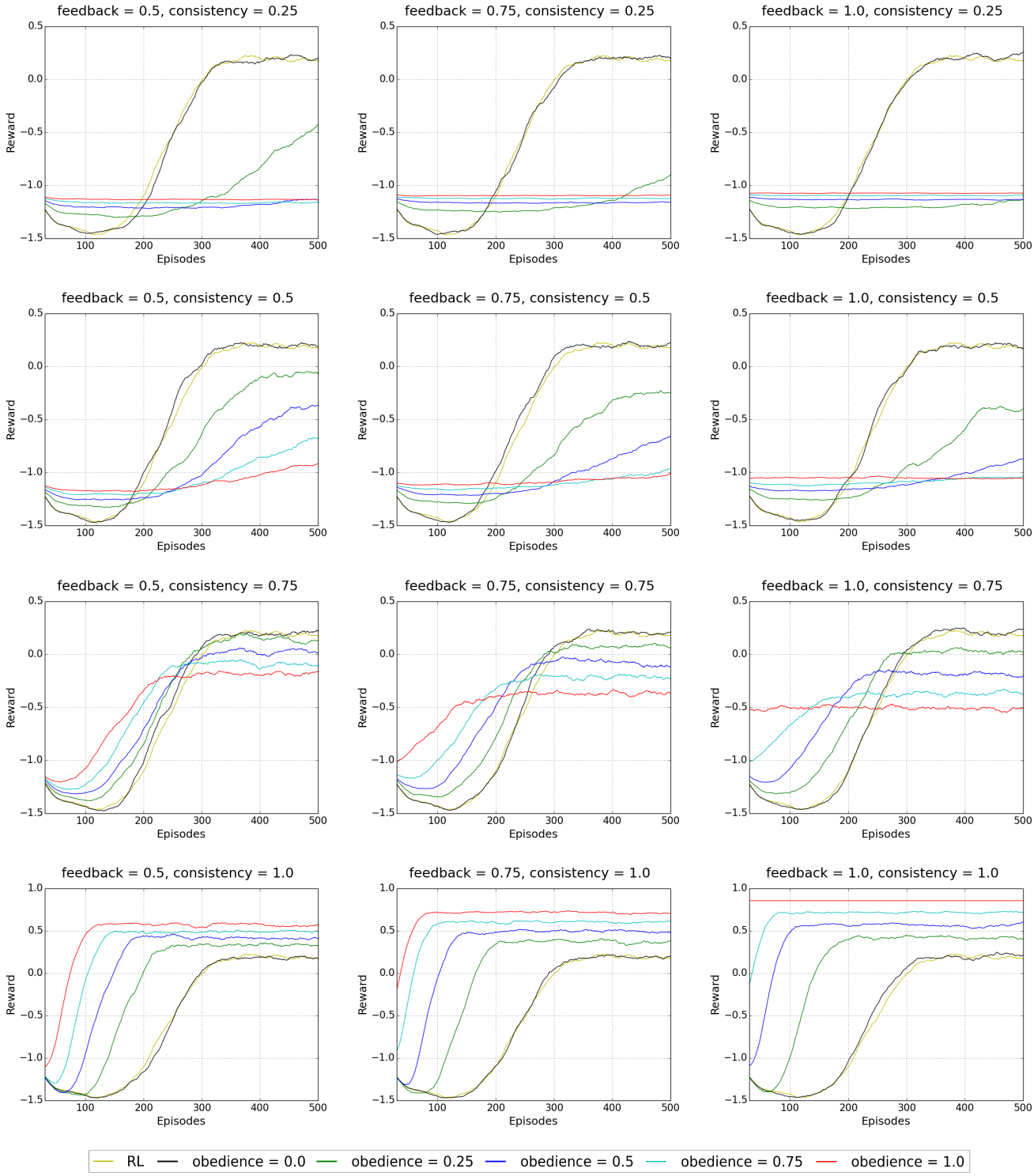


Figure 5.9: Collected reward for different learner obedience levels using several probabilities and consistencies of feedback. Higher probabilities of feedback do not necessarily lead to discernible improvements in the overall performance; however, important differences can be noted as higher consistencies of feedback are used.

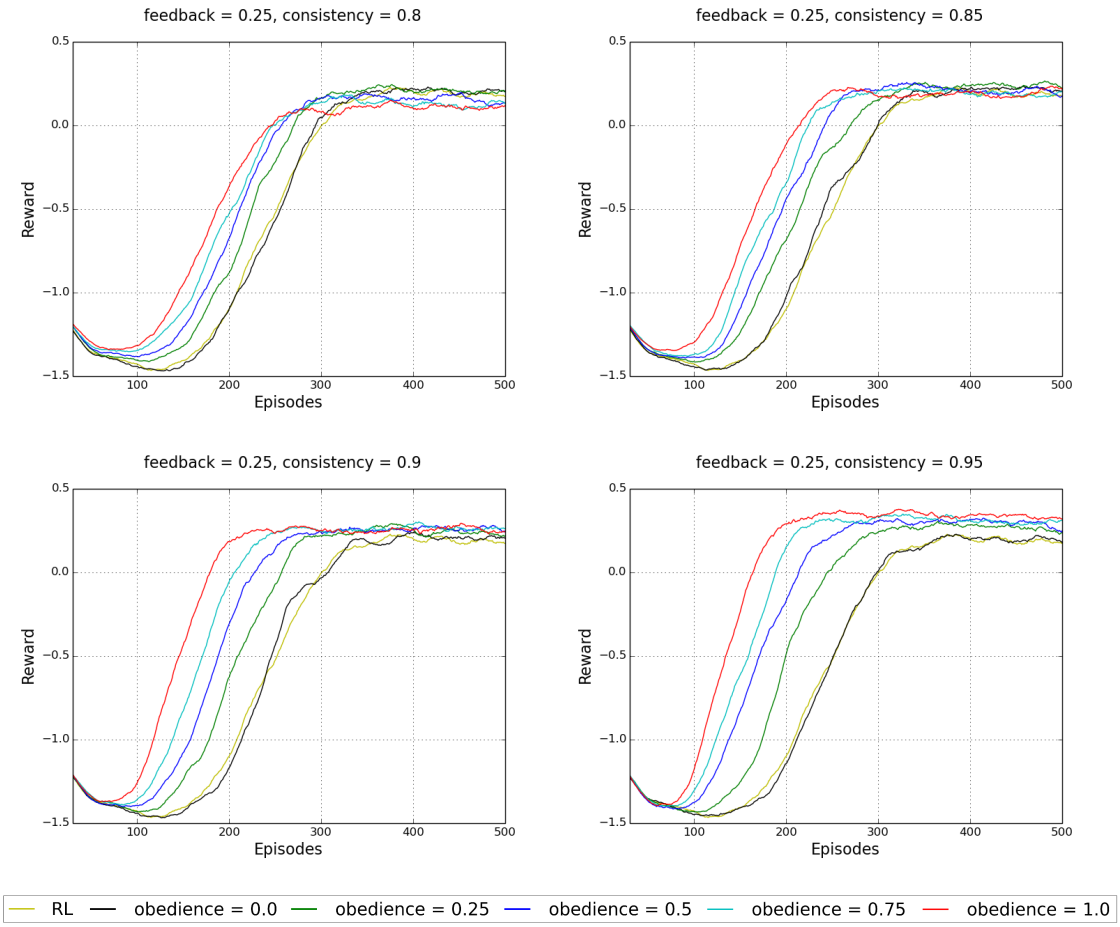


Figure 5.10: Collected reward for different values of learner obedience using fixed probability of feedback 0.25 and for four different cases for higher consistencies of feedback between 0.8 and 0.95.

feedback $\mathcal{C} = 1.0$ (fourth row in Fig. 5.9) in all cases the accumulated reward is higher than 0.5, but on the other hand, when using the consistency of feedback $\mathcal{C} = 0.75$ (third row in Fig. 5.9), the accumulated reward tends to slightly decrease as trainer advice increases, meaning that more interactive feedback does not help in the presence of poor consistency of feedback, or in other words of bad advice.

Ultimately, since the consistency of feedback shows considerable sensibility in the presence of small variations, we performed one additional experiment keeping the probability of feedback fixed to $\mathcal{L} = 0.25$ as in Fig. 5.8 since we use this value as a base as aforementioned. We tested the consistency of feedback with values $\mathcal{C} \in \{0.8, 0.85, 0.9, 0.95\}$ (consistency of 0.75 and 1.0 were already shown in Fig. 5.8)

to evaluate how these slight changes impact on the overall performance. Fig 5.10 shows the accumulated rewards for learner obedience $\mathcal{O} \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$. It can be seen that such small differences in the consistency of feedback can lead to dissimilar outcomes, ranging from behavior similar to autonomous RL when $\mathcal{C} = 0.8$ to behavior similar to a fully correctly advised learner-agent when $\mathcal{C} = 0.95$. Therefore, even a small proportion of bad advice can considerably impoverish the learning process, which shows how important it is to select trainers that can give useful advice in most states since specialised trainers, despite being more successful themselves from the initial state, have limited knowledge when it comes to states that lie outside their specialised policy.

5.4 Discussion

In this chapter, we presented a comparison of artificial agents that are used as parent-like teachers in an IRL cleaning scenario. We have defined three classes of trainer-agents related to our scenario, two of them being specialists in a particular path and another with no preference in any of them. Thus, the agents differ not only in their characteristics but also in the obtained performance during their own learning process and in turn as trainers. The differences in their main properties reflects in their behavior as i) the specialist-A agent with the largest accumulated reward, ii) the specialist-B agent with the largest amount of experience in terms of the number of explored states, and iii) the polymath agent with the smallest standard deviation.

The polymath agent is a better candidate to become an advisor since has experience in most of the states, and, therefore, it is able to properly advice in most of situations. Results show that using the polymath agent as an advisor, a learner-agent is able to both collect more reward and faster converge. Furthermore, the scenario has been tested on diverse parameter values for probability of feedback, consistency of feedback, and learner-agent's obedience showing differences in accumulated rewards especially regarding the consistency of feedback suggesting that slightly variations on the consistency affect considerably the learner-agent performance.

In this part of the thesis, we have introduced an agent-agent IRL approach. In

chapter 4 we have focused on IRL complemented by an affordance-based model, so-called contextual affordances. We have confirmed contextual affordances to be an efficient strategy to reduce the search space and, therefore, enabling a faster learning. Moreover, in chapter 4, we have studied interaction parameters such as the probability of feedback and consistency of feedback. We have observed that the consistency of feedback seems to be very important since small differences may affect the performance considerably. In chapter 5 we have analyzed what features of trainers are more relevant to become a good teacher. The polymath agent has shown the best results since it has more general knowledge about all the states, which is reflected in the smallest standard deviation. Furthermore, in this chapter, we have investigated in deeper the previous interaction parameters plus the learner-agent's obedience observing that the inclusion of this parameter improves the learner performance in presence of low consistencies of feedback.

In the following part of the thesis, we focus on implementing the cleaning-table scenario adding human parent-like trainers in the IRL loop using uni- and multi-sensory inputs. By using a human parent-like trainer, we expect not only to have a more realistic scenario but also to investigate how the sensory processing affects the learning process in terms of performed actions and accumulated rewards. In chapter 6, we introduce an architecture utilizing uni-modal input signals to guide the robot during the learning process. Afterward, in chapter 7, we will extend the approach to incorporate multi-modal inputs to enrich the feedback signal attempting to speed up the learner-agent performance.

Part III

Human-Agent Interactive Reinforcement Learning

Chapter 6

Speech Guidance Using a Domain-specific Language

6.1 Introduction

In the present chapter, we implement the cleaning-table scenario adding a human parent-like trainer in the IRL loop using uni-sensory inputs. By adding a human trainer, we move the scenario into a more realistic set-up, but additionally, we are able to investigate how the sensory processing affects the learning process. During this chapter, due to the good results obtained in chapter 4, we use again contextual affordances. Moreover, a robotic learner-agent obtains interactive feedback via a speech recognition system which is tested to work with five different microphones concerning their polar patterns and distance to the teacher to recognize sentences of different instructions. The designed experiments and obtained results are oriented to answer, in part, the third research question of this thesis: How beneficial is uni- and multi-modal advice during the apprenticeship process. This question is relevant in the IRL context, since it allows us to determine how uni-modal advice from humans may affect some parameters such as the consistency of feedback where small variations may impoverish the overall learning performance. In this chapter, we focus on uni-modal sensory inputs represented as audio input signals and processed by an automatic speech recognition (ASR) system. We use speech guidance during the apprenticeship in order to reduce the number of actions needed to complete the task by the learner-robot.

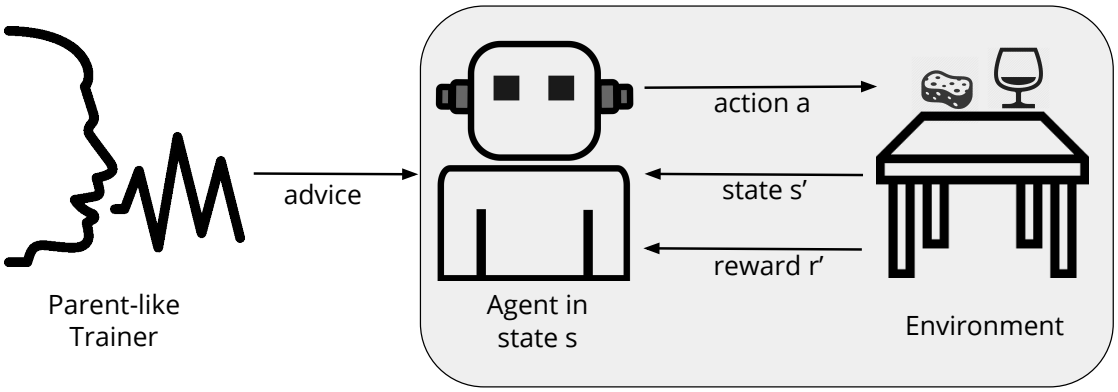


Figure 6.1: Interactive reinforcement learning with a human parent-like trainer to deliver spoken advice to the agent on how to perform the task faster with respect to the agent's autonomous exploration.

In the domestic scenario, the robot performing the cleaning-table task may be assisted receiving a degree of external guidance. In this regard, robots working in domestic scenarios may benefit from human expertise on how to perform a particular task (Thomaz and Breazeal, 2007; Suay and Chernova, 2011; Knox and Stone, 2012). Therefore, in our scenario, the robot receives spoken advice from a human trainer which is recognized by the ASR system. This way of giving instructions is natural for humans, but we need to control the probability of supplying feedback by the teacher, as humans can decide if they provide an instruction in a given situation. Hence, we use an artificial agent with full knowledge about the task to provide the spoken advice.

The implemented IRL approach allows to speed up the learning process by using a human parent-like advisor (as in Fig. 6.1) to support the learning by delivering useful spoken advice in selected episodes using policy shaping (Thomaz and Breazeal, 2007; Griffith et al., 2013). This approach reduces the search space and allows to learn the task faster in comparison to a fully autonomous agent (Suay and Chernova, 2011; Cruz et al., 2015). Hence, an apprentice agent can be taught by a parent-like trainer in a similar way as caregivers assist infants during the learning of new tasks.

Moreover, the implemented IRL approach uses contextual affordances to allow the robot to complete the cleaning task in every episode anticipating when chosen actions are possible to be performed. We hypothesize that contextual affordances

along with spoken advice may improve the convergence speed of reinforcement learning, avoiding wrong instructions that result from errors of the speech recognition system.

6.2 Automatic Speech Recognition

The given scenario originates from the Human-Robot Interaction (HRI) domain. For this reason, we do not only employ a humanoid robot as the learner, but there is also a humanoid teacher. Now, since the human way of instructing a robot is employing speech, the teacher also uses speech to instruct the learning robot by providing pre-recorded audio data that was spoken by a human. To understand the verbal commands, the apprentice processes audio data and recognizes the given guidance by applying an ASR system.

The ASR system we employ for our approach is the DOCKS system developed by Twiefel et al. (2014). The DOCKS system is based on *Google Voice Search* (Schalkwyk et al., 2010) which is a cloud-based ASR service processing audio data captured by a local microphone and generating hypotheses for the corresponding text representation. As *Google Voice Search* is generally applied in web searches, the involved language models are optimized for this task. The given HRI scenario differs from this field as robot instructions are verbalized which are not the first preference in web search-based ASR hypotheses. One possibility to overcome this issue is to integrate a local open-source ASR system which can be configured by providing a domain-specific language model for the given HRI scenario. However, the acoustic models employed by local open-source ASR systems provide a lower quality due to the lower amount of training data available during training.

To overcome the issues of either weak acoustic models or out-of-domain language models, DOCKS uses a post-processing technique to fit the ASR provided hypotheses by *Google Voice Search* to the given HRI domain. To be able to exploit the quality of the well-trained acoustic models employed by *Google Voice Search*, the ASR hypothesis is converted to a phonemic representation (Bisani and Ney, 2008). The converter is capable of creating a phoneme sequence for unknown words based on the provided training data and so overcomes the issue of unknown words contained in the ASR hypothesis provided by *Google Voice Search*.

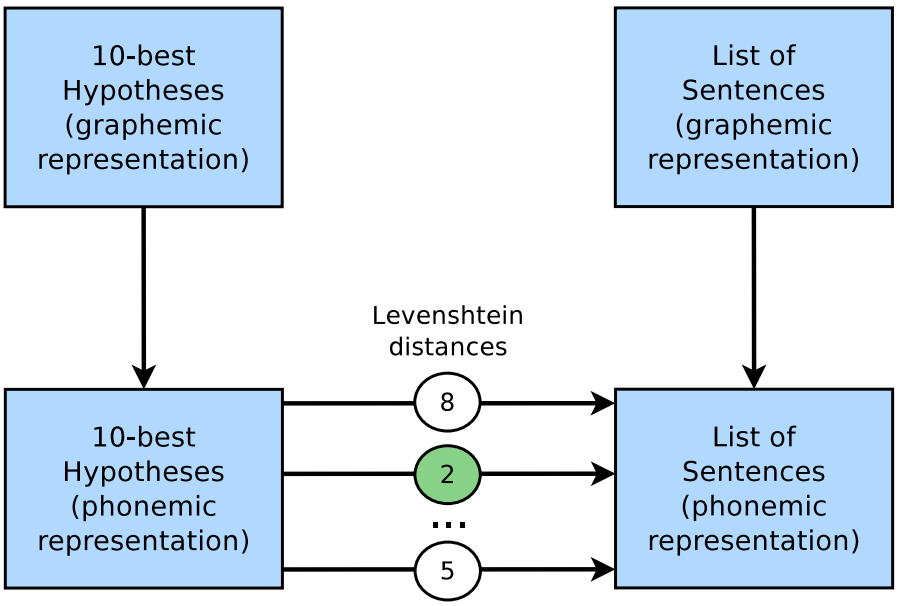


Figure 6.2: Functional principle of the ASR system. The left side shows the ASR hypotheses provided by Google and the right side contains the list of sentences for the given HRI scenario. In the middle, the Levenshtein distances are calculated.

For the given HRI scenario, a fixed set of robot commands is defined and represented by a list of sentences. To receive the best-matching hypothesis out of the list of sentences, the phonemic representation of the ASR hypothesis is compared to the phonemic representations of each sentence in the list. For this task, the Levenshtein distance (Levenshtein, 1966) is employed to calculate the difference between phoneme sequences. After calculating the Levenshtein distance between the ASR hypothesis and each sentence of the list, the sentence possessing the shortest distance is chosen as the best-matching result. To improve the technique, the Levenshtein distance is calculated for the ten best hypotheses provided by *Google Voice Search*. Fig. 6.2 summarizes the mentioned functional principle.

6.3 Experimental Set-up

This section presents an integrated system to teach a robot to perform the cleaning-table task using a parent-like trainer with occasional spoken instructional feedback. The system architecture consists of three modules which are shown in Fig. 6.3. At the top, there is the interface module where the external trainer provides a voice

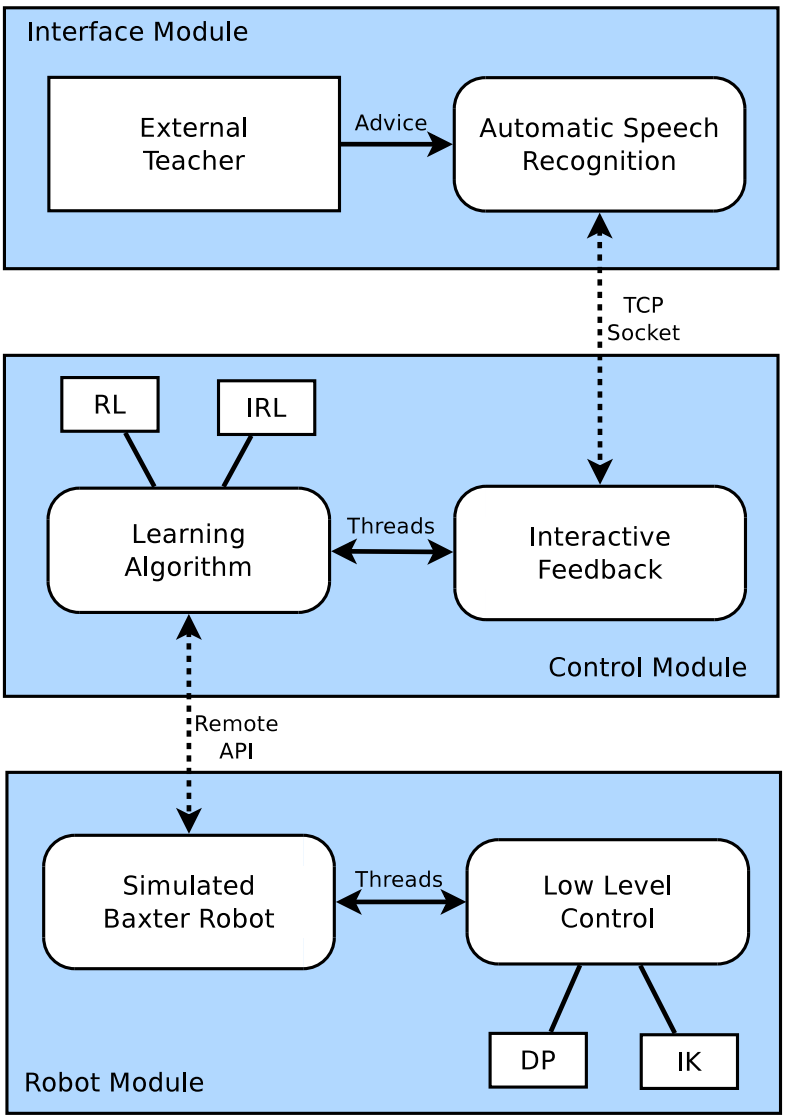


Figure 6.3: System architecture in three levels using speech guidance. At the top is the interface module which interacts with the external teacher, in the middle is the control module and at the bottom the robot module where the actions are performed.

stream which is processed by the ASR system and sent to the control module in the middle. The control module runs the learning algorithm that is able to perform autonomous RL and IRL generating choices for actions. These are passed to the robot module to be executed by a simulated Baxter robot combining two different approaches for low-level control, namely direct planning and inverse kinematics. We use a simulated Baxter robot because the low-level control model has been already implemented and runs quite stable in the V-REP simulator (Rohmer et al.,

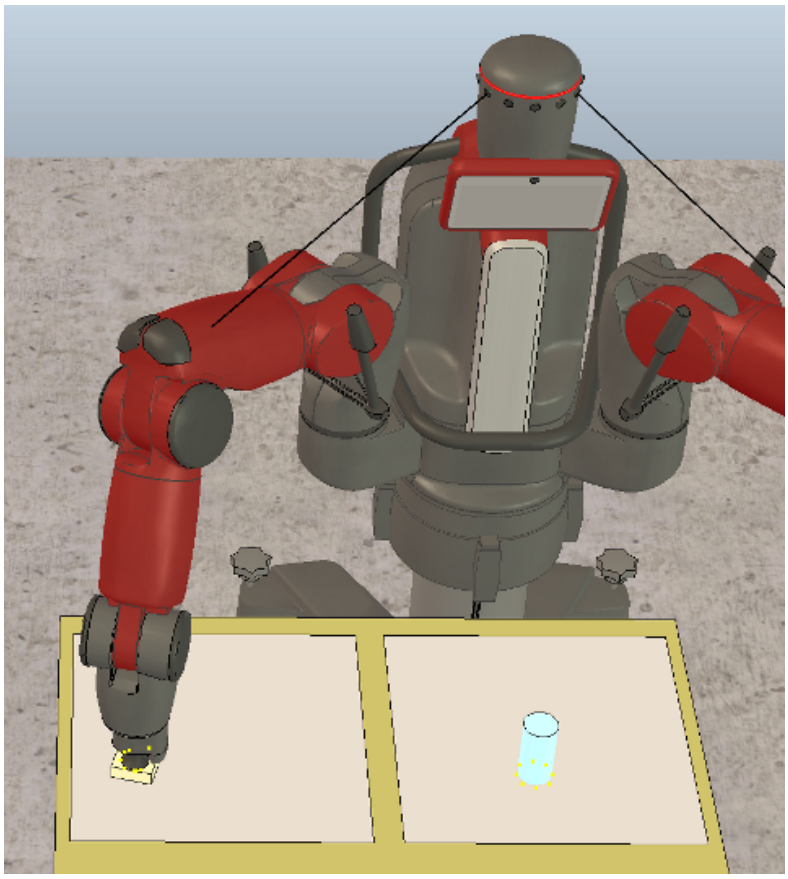


Figure 6.4: A simulated Baxter robot performs the actions in the environment which is created in the V-REP simulator.

2013). This allows us to set the focus into the analysis of the interaction parameters as well as into the uni-modal robot interface.

The cleaning-table task is carried out by the Baxter robot in a simulated environment using the V-REP simulator. All actions are performed using only one arm which has seven degrees of freedom (DoF). Fig. 6.4 shows the scenario while the Baxter robot is cleaning the table using the *sponge*. The Baxter robot has as end effector a vacuum cup, also called suction pad. We do not employ a gripper to grasp the object since the main focus of this work is to learn the right sequence quickly. Moreover, to reach the defined locations direct planning is used and then afterward inverse kinematics for low-level control is used to grab objects.

To make the scenario more complex, we created variations for every guidance instruction, expanding to 33 domain-specific instructions belonging to the 7 different classes of advice. For instance, advice *get the sponge* could also be stated as *pick*

up the sponge, *take the sponge*, *grasp the sponge*, or *lift the sponge*, but all of them represent the same instruction.

We use the IRL approach to training the robot utilizing the ϵ-greedy method for action selection with the following parameters determined by a grid search, $\alpha = 0.3$, $\gamma = 0.9$, and $\epsilon = 0.1$. Therefore, in most of the cases, the next action is selected as shown in Eq. (6.1):

$$a_t = \underset{a \in A_s}{\operatorname{argmax}}\, Q(s_t, a) \tag{6.1}$$

where s_t is the current state at time t, and A_s corresponds to a subset of all available actions. The subset of available actions is determined based on the contextual affordances using a similar ANN architecture as shown in Fig. 4.2. This way, it is possible to anticipate the effect of performing an action with an object in a particular state.

To train the contextual affordances model, data were obtained considering all possible states mentioned in Sec. 3.3.2 as well as the instructive classes taking into account the combination of actions and states. This led to 371 data for the training process.

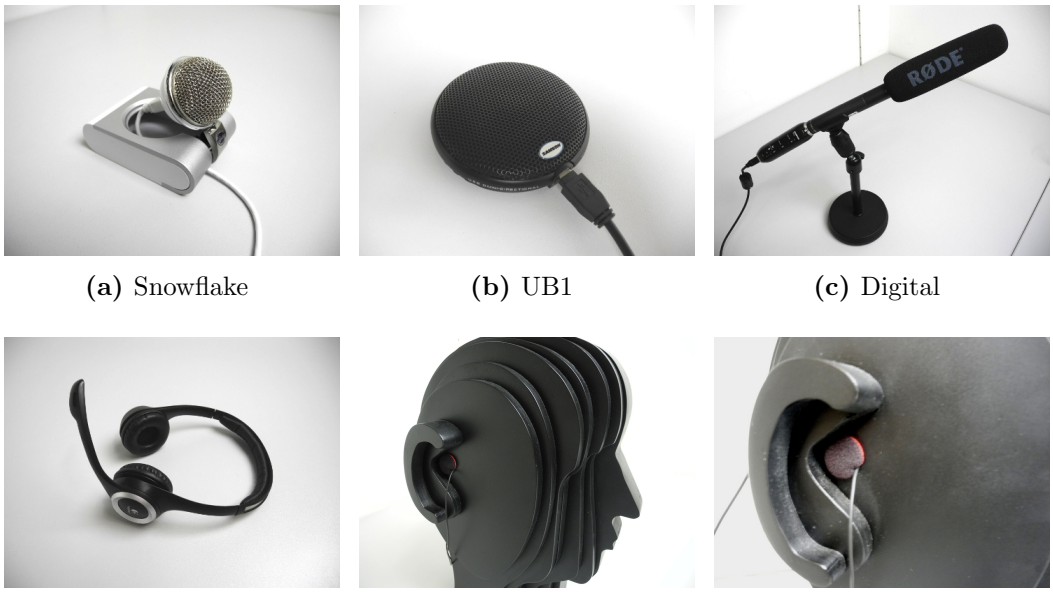
Our parent-like trainer possesses instructional recordings of the different advice classes. Therefore, the parent-like trainer is able to deliver selected interactive feedback to the robot using ASR at certain times during the learning process. Our IRL approach uses probability of feedback $\mathcal{L} = 0.2$ and consistency of feedback $\mathcal{C} = 0.9$ as interaction parameters.

Algorithm 6.1 shows the IRL approach using contextual affordances, interaction and speech recognition. The conditional statement starting in line 19 represents the fact that the external teacher delivers advice and changes the next action a_{t+1} by formulating a verbal instruction that is processed by the ASR system. Conditions in lines 8 and 18 represent the response of the neural network about the feasibility of performing the action in the current state which is called contextual affordance (method CAff(a_t, o, s_t) in the algorithm).

Algorithm 6.1. Interactive reinforcement learning approach using contextual affordances, interaction and speech recognition

Input: Previous definition of states and actions

Initialize $Q(s, a)$ arbitrarily
repeat
  if $rand(0, 1) <= \epsilon$ **then**
    Choose a_t randomly from A
  else
    Choose a_t according to $a = \underset{a=a_s}{\text{argmax}}\ Q(s, a)$
  end if
until CAff$(a_t, o, s_t) <> -1$
repeat
  Take action a_t
  Observe reward r_{t+1} and next state s_{t+1}
  repeat
    if $rand(0, 1) <= \epsilon$ **then**
      Choose a_{t+1} randomly from A
    else
      Choose a_{t+1} according to $a = \underset{a=a_s}{\text{argmax}}\ Q(s, a)$
    end if
  until CAff$(a_{t+1}, o, s_{t+1}) <> -1$
  if $rand(0, 1) <=$ feedbackProbability & $rand(0, 1) <=$ consistencyProbability **then**
    get *advice* from teacher voice using ASR
    if CAff$(a_{t+1}, o, s_{t+1}) <> -1$ **then**
      $a_{t+1} \leftarrow advice$
    end if
  end if
  $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$
  $s_t \leftarrow s_{t+1}$
  $a_t \leftarrow a_{t+1}$
until s is terminal



(a) Snowflake **(b)** UB1 **(c)** Digital

(d) Headset **(e)** Pro1 **(f)** Pro1(Details)

Figure 6.5: Microphones used in the experiments.

6.4 Experiments and Results

6.4.1 Automatic Speech Recognition Module

To carry out the cleaning-table task we consider different microphones to measure how the hardware affects quality in the ASR system and consequentially in the IRL approach. Therefore, we made simultaneous recordings using 5 different kinds of microphones and evaluated the answers of our ASR system. Afterwards, we made the scenario more difficult by positioning the microphones at a distance of 1m away from the speaker, which leads to the necessity of increasing the strength of the audio signal to compensate for the lower volume of the speech instructions and with this also increasing the level of environmental noise contained in the audio signal. As a hypothesis, we claim that more noisy audio data leads to worse ASR performance and so we can measure the robustness of the learning system by providing incorrect instructions. The microphones were *Snowflake, UB1, Digital, Headset,* and *Pro1* which are shown in Fig. 6.5. *Snowflake*'s polar pattern is cardioid, *UB1* is omnidirectional, *Digital* is supercardioid, *Headset* is unidirectional, and the *Pro1* is omnidirectional. Only 16kHz, mono channel audio data was utilized.

Table 6.1: Word and Sentence Error Rate (%) in ASR for all microphones used at normal and at 1m distance.

Microphone	Normal distance		1m distance	
	WER	**SER**	**WER 1m**	**SER 1m**
Snowflake	0	0	0.88	3.03
UB1	0	0	1.75	6.06
Digital	0.88	3.03	1.75	6.06
Headset	0.88	3.03	3.51	12.12
Pro1	11.40	27.27	14.91	30.30

The response of the ASR module for the domain-specific language model measured in Word Error Rate (WER) and Sentence Error Rate (SER) is shown in Fig. 6.6 and in table 6.1 as percentages for normal distance and 1m distance. In this context, normal distance means that the microphone is placed in its normal working position depending on its characteristics, i.e., on the table just in front of the user for *Snowflake, UB1, and Digital*, on the ears of a wood head placed in front of the user for *Pro1*, and on the user's head for *Headset*. The SER depends on the sentence accuracy S_{Acc} as shown in the following equation:

$$SER = 1 - S_{Acc} \tag{6.2}$$

We observed that the microphone with the best results working with and without noise is *Snowflake* and the microphone with the worst result in both cases is *Pro1*.

6.4.2 Learning Module

To test the learning module three different set-ups were implemented: first the robot working autonomously, second the robot working with advice taken from the *Pro1* microphone, and third the robot working with advice taken from the *Snowflake* microphone. The two latest set-ups were run with the best and the worst microphone performances in the domain-specific language model. In both cases, microphones at a distance of 1m away from the teacher were used. The

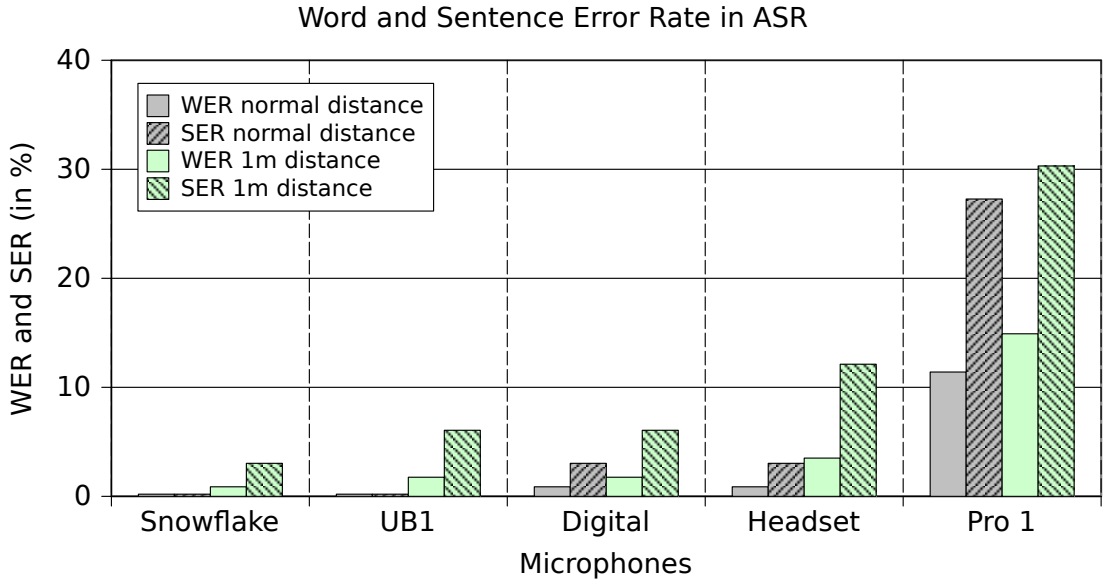


Figure 6.6: Response of the ASR system to the list of sentences using different microphones at normal and at 1m distance. WER and SER are shown as percentages.

motivation is to test how influential the quality of the microphone is in improving the speed of convergence in IRL. This is a relevant question in the IRL context, since, as shown in the previous chapter, small differences in the consistency of feedback impoverish the learning performance.

Each set-up was carried out 100 times and the results were averaged. Fig. 6.7 shows the average number of actions performed during 100 episodes. The y-axis is truncated at 200 actions to highlight the difference between the 3 set-ups. In each episode, the cleaning task was always finished because of contextual affordances which allowed to avoid performing actions that, according to the robot's current state, were not feasible.

In Fig. 6.7, we show that both IRL approaches perform better than RL working autonomously, nevertheless, none of them outperform the previous IRL approaches shown in chapter 4. These results are in connection with using an ASR system that implies a bigger error rate from recognition mistakes which can also be seen as a lower consistency of feedback. Moreover, there is no significant difference between IRL with *Pro1* and *Snowflake* microphones. To analyze this, we defined the actual interactive feedback rate ($\mathcal{I}$) which is the percentage of steps where a

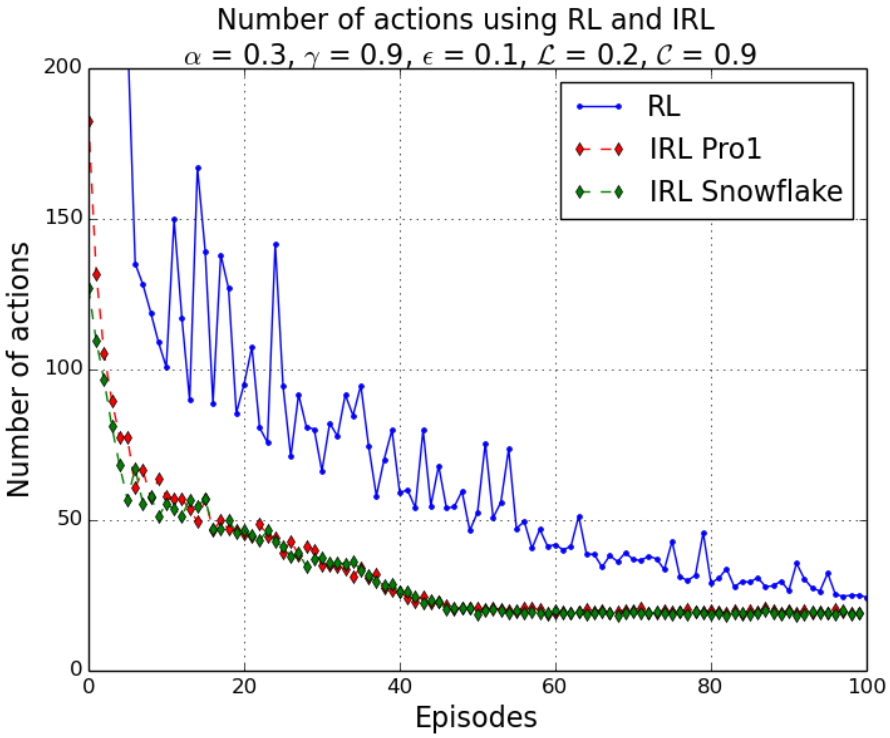


Figure 6.7: Average number of actions performed to finish the task using an RL agent (blue) and an IRL agent with two different microphones (red and green). Despite the differences in the hardware quality, the IRL approaches show improvement compared to the RL approach.

correct instruction was properly received by the agent:

$$\mathcal{I} = \mathcal{L} * \mathcal{C} * S_{Acc} \tag{6.3}$$

We computed $\mathcal{I}$ from the feedback probability ($\mathcal{L} = 0.2$), consistency probability ($\mathcal{C} = 0.9$), and the S_{Acc}. Given $SER \in [3.03\%, 30.30\%]$ the results are $\mathcal{I} = 12.55\%$ for *Pro1* and $\mathcal{I} = 17.45\%$ for *Snowflake*, these values are already enough for the agent to benefit from the interaction. This is consistent with a study where it is shown that large improvements of RL by IRL are already achieved at low interaction rates (Stahlhut et al., 2015). In fact, it is possible to observe in Fig. 6.7 that there is a small variation in the first ten episodes but in the following episodes variations get even smaller. This leads to a system which is able to perform the task properly and which is robust for a variety of audio hardware.

6.5 Discussion

In this chapter, we have shown ASR to be an effective method to work in IRL scenarios to improve the speed of convergence of RL agents. For scenarios where a human would verbally instruct a robot during IRL, our results indicate that interaction helps to increase the learning speed robustly even with an impoverished ASR system. However, the IRL approach using uni-modal spoken advice does not outperform agent-agent approaches reviewed in chapters 4 and 5 due to the error rate of the ASR system.

Although we have shown uni-modal interactive feedback to benefit IRL apprenticeship process, it is still an open question how multi-modal advice may favor a learner-agent in comparison to autonomous RL and uni-modal IRL. In the following chapter, we incorporate another sensory modality in order to compose a stronger advice signal. Our hypothesis is that multi-modal inputs may lead to higher consistency of feedback, i.e., more confidence level of advice and therefore to a faster learning process. Moreover, in multi-modal IRL, contextual affordances may contribute not only in terms of faster convergence but also in modulating the integration of signals from multiple sources by representing the expectation of a learner-agent. These issues will be addressed in the next chapter.

Chapter 7

Multi-modal Feedback Using Audiovisual Sensory Inputs

7.1 Introduction

Robots in domestic environments are receiving more attention, especially in scenarios where they should interact with parent-like trainers for dynamically acquiring and refining knowledge. In the previous chapter, we have proposed an uni-modal control interface that is limited to audio signals and thus do not take into account multiple sensor modalities. In this chapter, we propose the integration of audiovisual patterns to provide advice to the agent using multi-modal information. In this approach, advice can be given using either speech, gestures, or a combination of both. The performed experiments are designed in order to complement the answer of the third research question: How beneficial is uni- and multi-modal advice during the apprenticeship process? This question has been partially addressed in the previous chapter where an uni-modal IRL approach has been implemented by using an ASR system. In this chapter, we hypothesize that processing audiovisual input information as interactive feedback may lead the learner-agent to faster convergence by receiving a more accurate, consistent advice. Furthermore, we investigate the multi-modal integration model modulated by the use of contextual affordances.

In real domestic scenarios, caregivers interact with infants through diverse stimuli

(e.g., speech, gestures) that can be seen as multiple modalities. These stimuli are seen as guidance from the parent-like teacher to the learner-agent. Nevertheless, when multiple modalities are taken into account, this may lead to misinterpretations and problems with the integration of such multi-modal signals, especially when the information from multiple sources is in conflict or ambiguous (Bauer et al., 2015). Consequently, instructions may not be clear and may be misunderstood, thereby leading to a decreased performance in the apprentice agent when solving a task, as shown in the previous chapter 4. Although IRL approaches have been implemented in robotic scenarios (Suay and Chernova, 2011; Knox et al., 2012, 2013b), an open issue is that the communication interface between the teacher and the robot may not be straightforward for non-expert trainers in a domestic environment. Therefore, there is a motivation to develop easier interactive scenarios where parent-like teachers can provide instructions using their natural communication skills such as speech and gestures. In this setting though, the feedback provided by the user may be incongruent or noisy. The integration of multiple modalities should also consider this case in order to provide the learning algorithm with robust perceptual cues.

Our algorithm integrates multi-modal information to provide robust commands from sensory cues along with a confidence value indicating the trustworthiness of the feedback. The integration considers also the case in which the two modalities convey incongruent information. We utilize a neural network-based approach to integrate multi-modal information from uni-modal modules based on their confidence. Additionally, we modulate the influence of sensory-driven feedback in the IRL task using environmental knowledge in terms of contextual affordances. This is motivated by neurobehavioral studies on multi-modal processing in which human subjects exposed to audiovisual stimuli integrate multiple sources of information driven by a combination of sensory representations and prior expectations (Odegaard et al., 2015). In this regard, we use a neural network architecture to predict the effect of performed actions to avoid failed-states.

During the apprenticeship process, advice can be provided by a parent-like trainer using audiovisual inputs, respectively speech and gestures. Our proposed architecture is able to process information from multiple sources with the use of a neural associative memory that computes multi-modal advice as a function of the recognition and confidence of uni-modal modules. We present a set of experiments using

the 7 possible advice classes from audiovisual inputs. We want to show that multi-modal integration may lead to a better performance of IRL, with the robot being able to learn using a smaller number of training episodes compared to uni-modal scenarios.

With this aim, we conduct a set of experiments to explore the interplay of external feedback and task-oriented affordances in the cleaning-table scenario in which a robot can interact with objects with the goal of cleaning the surface of the table. We compared the learning performance in terms of speed of convergence and accumulated reward under 3 different conditions: different threshold of minimal confidence, uni- and multi-modal advice comparison, and integration of environmental knowledge by contextual affordances. For this purpose, we varied the percentage of available feedback and contextual affordances during the learning process.

7.2 Interactive Reinforcement Learning Interfaces

Although IRL has been implemented in robotic scenarios, a general problem is that the communication interface between the trainer and the robot has not been developed in a natural manner for domestic scenarios. For instance, Suay and Chernova (2011) addressed an IRL task where the parent-like trainer was able to deliver guidance using a graphic interface built from a camera image and adding buttons and bars for interaction. Another IRL approach was proposed by Knox et al. (2012), in which the device utilized to deliver feedback to the robot was a presentation control (a presenter), allowing to switch between positive and negative reward.

In the aforementioned approaches, the interfaces are useful in terms of accomplishing the interaction with an external trainer. Nevertheless, these interfaces are quite tedious and impractical for non-expert trainers who may not be familiar with the use of technological interfaces and hence have to spend longer time learning how to use the interaction device which, in the case of unsuccessful attempts, may lead to trainer's frustration. In home-like environments, external trainers should be able to use their natural communication skills (e.g., speech and gestures).

Therefore, it is much more desirable to have more natural interactive scenarios where external parent-like trainers can deliver their instructions similar to caregivers instructing infants. To this aim, in the previous chapter, we already presented an uni-modal IRL approach restricted to the use of an ASR system to guide an apprentice robot in the achievement of the cleaning-table task. To add more natural communication manners, in this chapter, we also incorporate visual patterns and integrate it with audio patterns as consistent guidance for the learner-agent during the apprenticeship process.

7.3 Multi-modal Integration in Robotics

People are constantly subject to different perceptual stimuli through different modalities such as vision, audition, and touch among others. Such modalities are used to perceive information and process it independently, in parallel, or integrating the received information to provide a coherent and robust perceptual experience. Similarly, humanoid robots work with many of these sensory modalities and the way of processing and integrating the information coming from various sources is currently an important research issue in autonomous robotics. In HRI scenarios, robots can take advantage of such multi-sensory information in order to improve their capabilities by improving the frequency and consistency of feedback when any sensory modality is limited, lacking, or unavailable.

For instance, early work by Andre et al. (1998) proposed a multi-modal integration of speech and gestures for human-computer interaction using a tactile glove to identify hand gestures and a microphone array for speech recognition. The system functionality was limited to manipulate geometric objects on topographical maps. In robotic scenarios, Wermter et al. (2003) designed a neurobiologically inspired robot for multi-modal integration and topological organization of actions with an associative memory. Their work integrated motor, vision, and language representations for learning by demonstration.

Lacheze et al. (2009) presented an approach for the recognition of static patterns fusing audio and video. In their work, auditory information was used to recognize objects that were partially occluded and therefore difficult to detect using only vision. Sanchez-Riera et al. (2012) presented a scenario with a robot com-

panion that performs audiovisual fusion for speaker detection using a multi-modal Gaussian mixture model. The approach detected multiple speakers in a domestic scenario with information from two microphones and two cameras mounted on a humanoid robot.

Kimura and Hasegawa (2015) used an incremental neural network to integrate real-time information in order to estimate attributes for unknown objects. The method used an RGB-D camera, a stereo microphone, and pressure and weight sensors to process different modalities. Ozasa et al. (2012) proposed the integration of image and speech recognition confidence values to improve the recognition accuracy of unknown objects using a logistic regression. In their approach, the confidence integration does not consider the case in which predicted labels are in contradiction. Moreover, in order to obtain improved recognition, it is also necessary to estimate proper logistic regression coefficients.

In all the aforementioned approaches, the confidence level was either not used or used only when the predicted labels are identical which is not always the case in HRI scenarios. Therefore, in domestic scenarios and dynamic environments, assistive robot companions still need to understand and interpret instructions faster and more efficiently, yielding the integration of available multi-sensory information with different confidence levels in a consistent mode considering equal and different predicted labels.

7.4 Experimental Set-up

In our architecture, a parent-like trainer interacts with an apprentice robot using speech and gestures as guidance for the cleaning-table scenario. In this chapter, we are particularly focused on processing audiovisual inputs and their integration. Fig. 7.1 shows the overall extended architecture of our system, where we now use a microphone and a depth sensor to capture the advice from the parent-like trainer that is subsequently integrated and sent to the IRL algorithm as one single piece of consistent advice. The integrated advised action is then sent to a NICO (Neural Inspired COmpanion) robot to be performed using the *pypot* library (Lapeyre et al., 2014), allowing to control the robot actuators either in real or simulated environments.

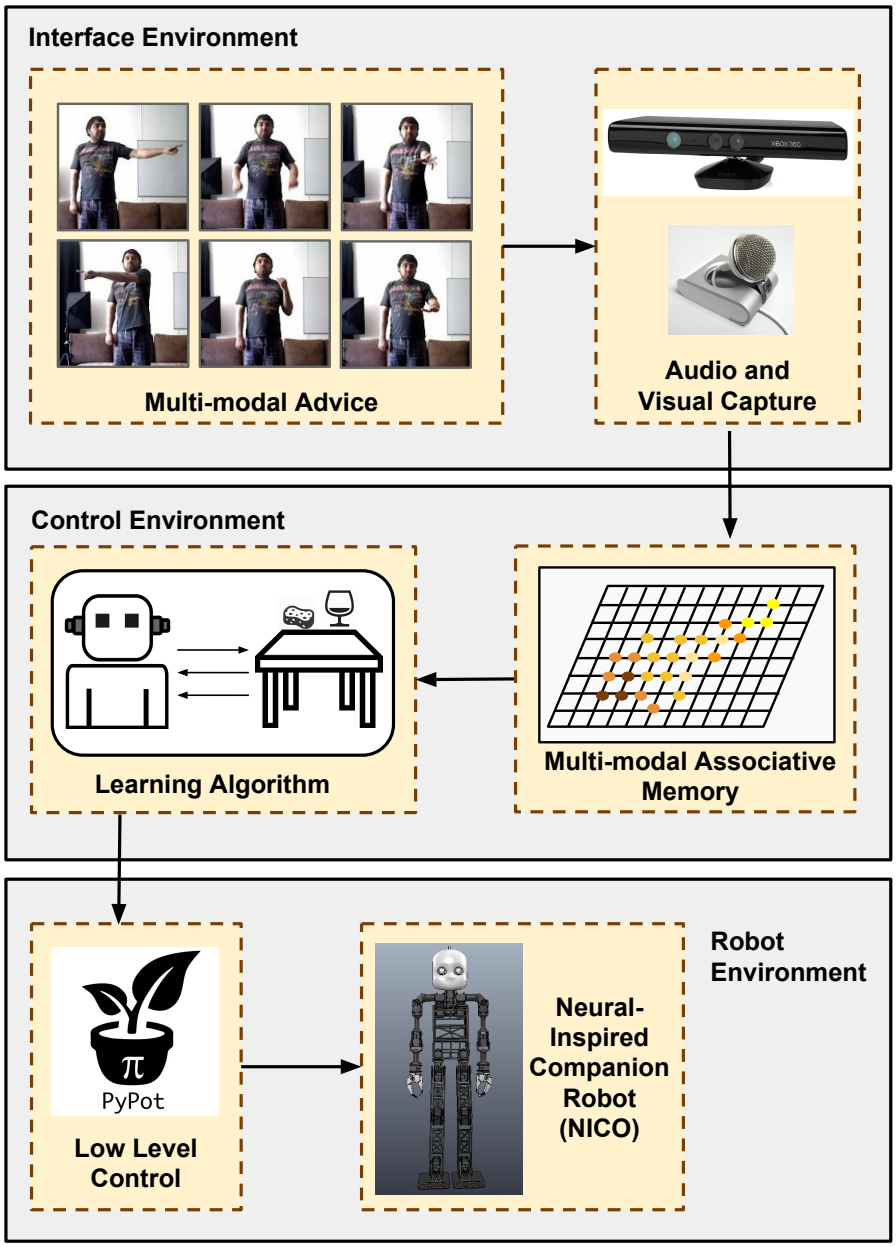


Figure 7.1: Overall view of the system architecture in three levels using multi-modal advice. In the interface environment, we use the robot with a microphone and a depth sensor to capture advice from the parent-like trainer. In the control environment, we integrate the advice and send it to the IRL algorithm associated with a confidence value to decide when a valid advice is considered according to a defined threshold. The integrated advised action is then sent to the robot environment where a NICO robot performs the action using the *pypot* library which allows to control the robot actuators either in the real or simulated environment.

In this approach, RL is performed using the SARSA algorithm with learning rate $\alpha = 0.3$, discount factor $\gamma = 0.9$, and ϵ-greedy action selection with $\epsilon = 0.1$. As before, the parameters are selected using the grid search method. In the IRL approach, we use probability of advice $\mathcal{L} = 0.3$. The following subsections describe how each modality module is implemented and how they are integrated in order to obtain a unified advice to provide a more effective guidance for the robot learning task.

7.4.1 Automatic Speech Recognition

To understand the verbal commands, the apprentice robot processes audio data and recognizes the given advice by applying the same ASR system used in the previous chapter proposed by Twiefel et al. (2014) (see Sec. 6.4.1). Box A in Fig. 7.2 shows in context the functional principle of the ASR system employed in our architecture.

Given the set H of the 10-best sentence hypotheses and the set S of the in-domain sentences, the predicted auditory class label is computed as:

$$\lambda^A \;=\; \text{argmin}\; \mathcal{L}(h_i, s_j) \tag{7.1}$$

where $\mathcal{L}$ is the Levenshtein distance in our ASR system. The confidence value is computed as:

$$\gamma^A = \max(0, 1 - \mathcal{L}(h_i, s_j)/|s_j|) \tag{7.2}$$

with $h_i \in H$ and $s_j \in S$, both represented as phonemes.

7.4.2 Gesture Recognition

For gesture recognition, we used an extended version of the HandSOM framework, developed by Parisi et al. (2014), that extracts hand-independent gesture features from depth map sequences. The learning model consists of a set of two hierarchically arranged Growing When Required (GWR) self-organizing networks (Marsland et al., 2002) that learn the spatiotemporal structure of the input sequences

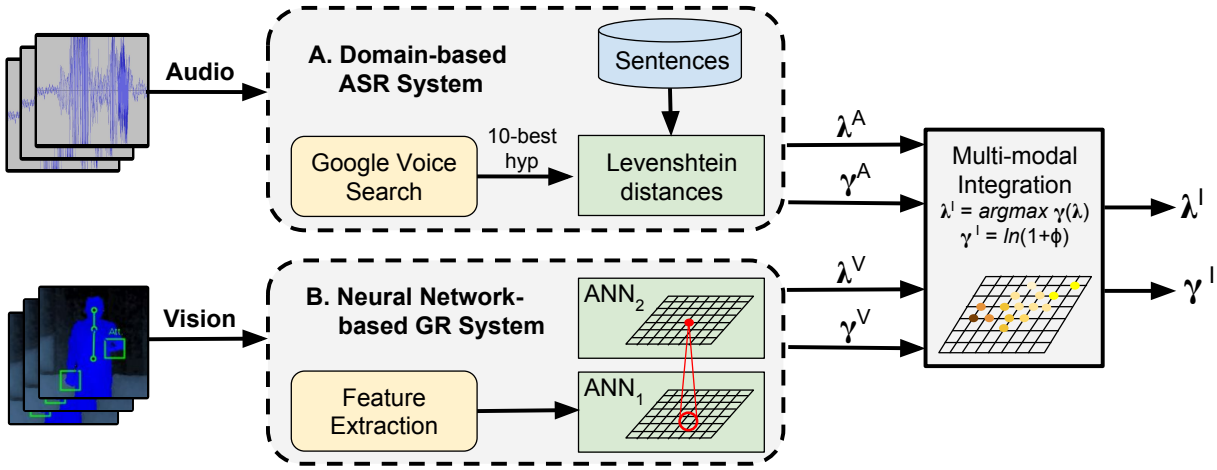


Figure 7.2: Overall view of the system processing scheme. The domain-based ASR system (on top) processes the audio input modality to obtain an audio advice label λ^A and an audio confidence value γ^A and the neural network-based gesture recognition system (at bottom) processes the visual input modality to obtain a visual advice label λ^V and a visual confidence value γ^V. Afterward, they become the input of the multi-modal integrative system to obtain the integrated advice label λ^I and the integrated confidence value γ^I (Cruz et al., 2016b).

in terms of gesture features (box B in Fig. 7.2). The GWR training algorithm for attaching labels to neural activation trajectories and the training parameters were discussed in (Parisi et al., 2015) and (Cruz et al., 2016b) respectively.

Along with a predicted label, we also estimate a confidence value that expresses the degree of belief that the prediction is correct based on sensory-driven observations. Training videos were recorded with an ASUS Xtion depth sensor from which we estimated the 3D skeleton model.

A label prediction is carried out every 3 frames to attenuate noise in a sliding window scheme. We consider the last 5 observations and compute the statistical mode that returns the most frequent value given the set of predictions Λ^V, from which we compute the gesture class as:

$$\lambda^V = Mo(\Lambda^V) \tag{7.3}$$

Let N be the number of occurrences of λ^V in Λ^V so that the confidence value can

be defined as:

$$\gamma^V = N/|\Lambda^V| \tag{7.4}$$

thus yielding a confidence value in the range between 1 and 0.2.

7.4.3 Multi-modal Integration of Audiovisual Patterns

A general overview of the processing scheme including the speech and gesture approaches is depicted in Fig. 7.2, where λ and γ are the label and the confidence value respectively. First, the audio and visual sensory inputs are individually processed. Then, the outputs, i.e., predicted labels and confidence values, become inputs for the multi-modal integration system. To ingrate the two aforementioned sensory modalities, we propose a mathematical transformation.

Our mathematical function receives as inputs the predicted advice classes and confidence pairs from the uni-sensory inputs respectively denoted as (λ^A, γ^A) for audio and (λ^V, γ^V) for vision. As outputs, the proposed model calculates an integrated advice and confidence value denoted by (λ^I, γ^I)

We compute the integrated label λ^I using the highest confidence value:

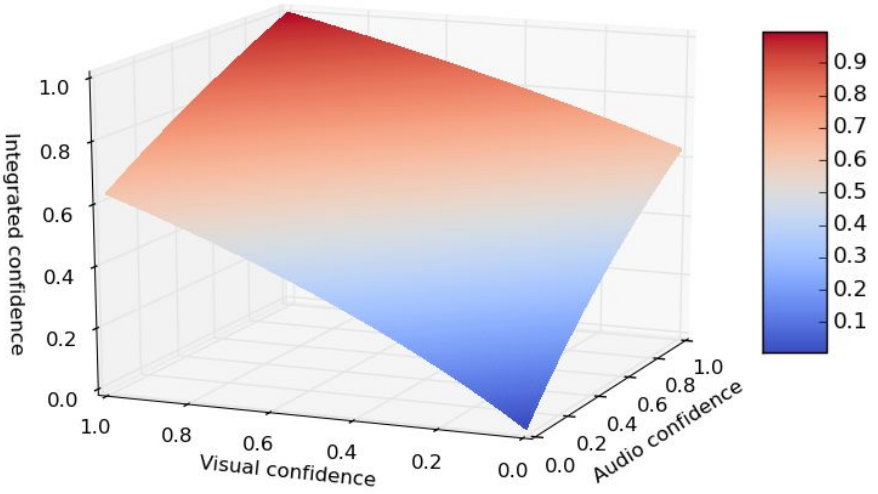
$$\lambda^I = \begin{cases} \lambda^A & \text{if } \gamma^A > \gamma^V \\ \lambda^V & \text{otherwise} \end{cases} \tag{7.5}$$

Therefore, when λ^A and λ^V are equal, any of them is assigned to λ^I regardless the confidence level. In case that λ^A and λ^V are different, then the label with a greater confidence is assigned to λ^I.

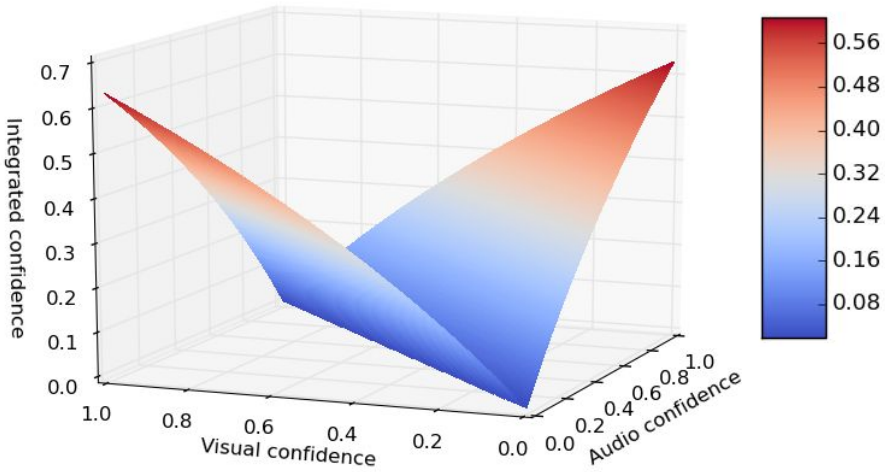
The integrated confidence value is determined by:

$$\gamma^I = \ln(1 + \phi) \tag{7.6}$$

with ϕ being a dynamic parameter depending on each congruent or incongruent pair of predicted labels and their confidence values. We refer to this time-varying parameter as the *likeliness parameter* which we compute as follows:



(a) Integrated confidence with equal uni-modal predicted labels



(b) Integrated confidence with different uni-modal predicted labels

Figure 7.3: Confidence values used in the neural network-based associative architecture. While in (a) the corresponding output labels for audio and visual modalities are the same, in (b) they are different.

$$\phi = \begin{cases} \gamma^A + \gamma^V & \text{if } \lambda^A = \lambda^V \\ |\gamma^A - \gamma^V| & \text{if } \lambda^A \neq \lambda^V \end{cases} \tag{7.7}$$

The likeliness parameter strengthens the integrated confidence value γ^I when the

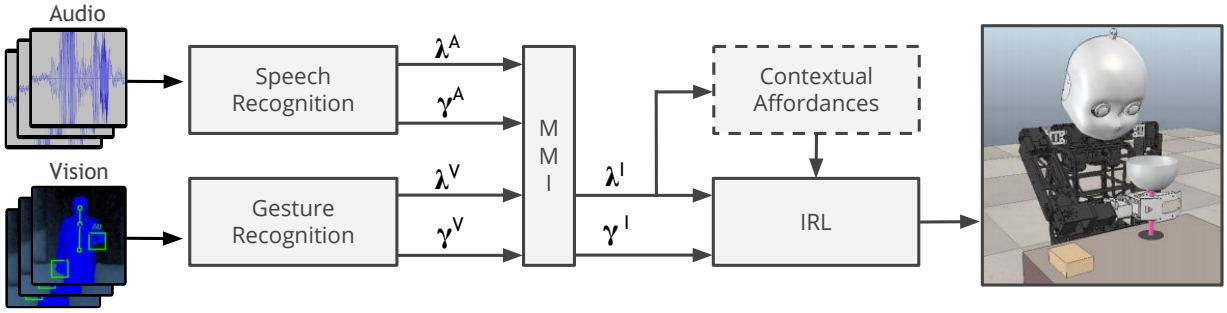


Figure 7.4: A diagram of the processing scheme for the IRL task including multi-modal integration (MMI) contextual affordances. Integrated feedback labels are used as input to compute contextual affordances (e.g., the effects of action λ^I given the current state). Contextual affordances modulate the influence of external feedback on the IRL algorithm, i.e., actions that lead to a failed-state are bypassed by the IRL algorithm.

predicted labels for audio and vision are congruent, whereas γ^I diminishes if the predicted uni-modal labels are incongruent.

The integration function yields an integrated confidence value $\gamma^I \in [\ln(1), \ln(3)] = [0, 1.0986]$. Therefore, after applying the transformation function, the confidence value is rescaled using a unity-based normalization to rescale the range of confidence between 0 and 1 as follows:

$$\gamma^I = \frac{\gamma^I - min(\Gamma)}{max(\Gamma) - min(\Gamma)} \tag{7.8}$$

where Γ is the set of all possible confidence values γ^I. Fig. 7.3 shows the behavior of the integrated confidence from audiovisual confidence values when the predicted labels are (a) congruent and (b) incongruent.

Additionally, we use contextual affordances to modulate the multi-modal integration. An updated diagram of the processing scheme including contextual affordances is illustrated in Fig. 7.4. The integrated feedback label λ^I is used as input to compute the contextual affordances to determine the effect of an action λ^I given the current state. Thus, the IRL algorithm may consider or not the feedback, e.g., disregarding the feedback that leads to the agent to a failed-state from which the task cannot be successfully carried out. In our IRL approach, we use different levels of availability for contextual affordances to modulate the learning process.

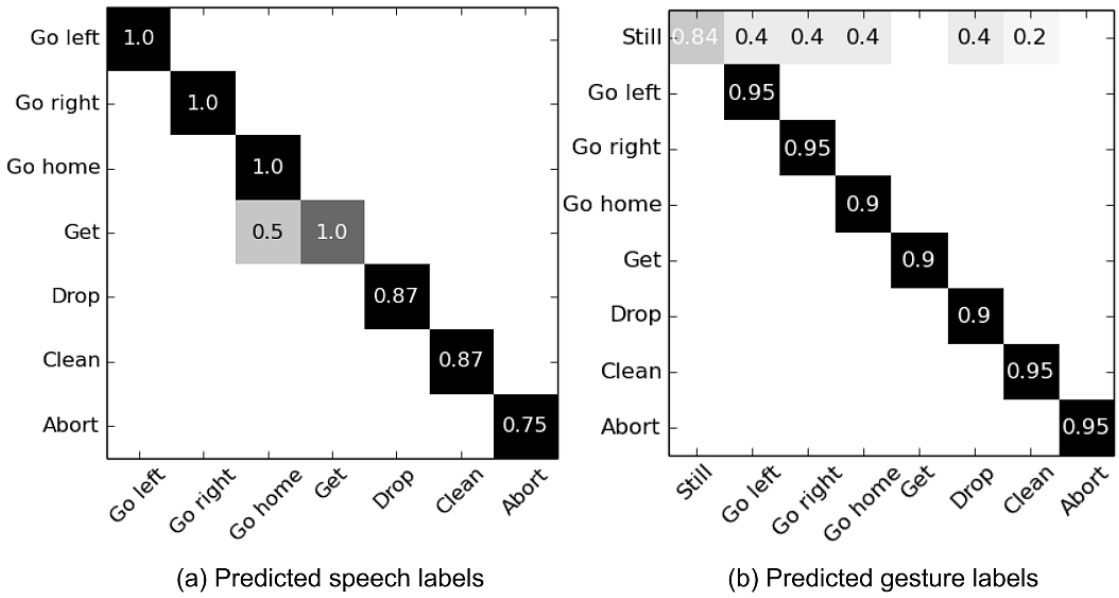


Figure 7.5: Confusion matrices with the average confidence values for predicted (a) speech and (b) gesture labels. The input speech advice of *get* was predicted in one occasion as *go home* with a confidence of 0.5. Nevertheless, all other predicted labels were correctly classified with high confidence values over 0.75. The gesture *still* was in some occasions misclassified with low confidence of 0.4 and 0.2. This was due to the transition from one gesture to the next and the use of the last three consecutive frames for the prediction. Regardless, all the gestures were correctly classified with high confidence values over 0.84.

7.5 Experiments and Results

7.5.1 Uni-modal Predictions

The robotic domestic scenario described in Sec. 3 has been implemented in order to test our proposed method. For this, we made recordings of audio and visual sequences from a parent-like teacher. Each advice class was recorded four times. Recordings enabled us to better control the conducted experiments in order to repeat the process under different learning conditions. After the training was completed, our goal was to predict the feedback labels from novel audio and video sequences (λ^A, λ^V) along with their confidence values (γ^A, γ^V). After processing each modality independently, the predictions were integrated using the multi-modal integration model to compute λ^I and γ^I.

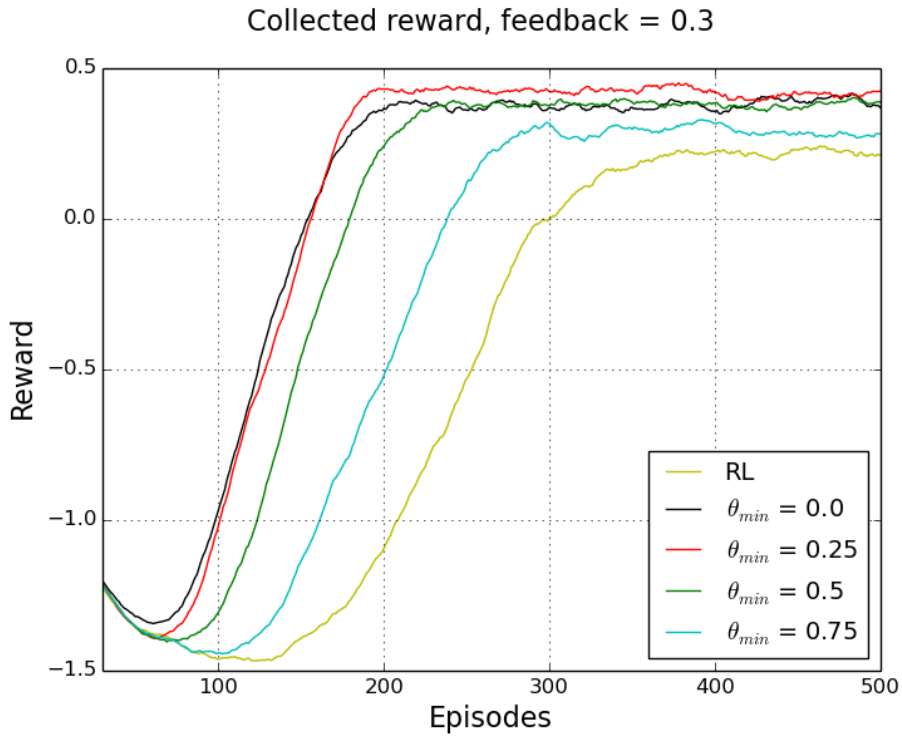


Figure 7.6: Collected rewards using autonomous RL and IRL with multi-modal feedback. We use different minimum confidence thresholds to consider an advice as valid. The best performance was observed for $\theta_{min} = 0.25$ (red line) (Cruz et al., 2016b).

Fig. 7.5a shows the confusion matrix with the average confidence values for the predicted speech labels whereas the confusion matrix with the average confidence values for the predicted gesture labels is shown in Fig. 7.5b. In the latter, we added the label *still* since the depth sensor is always processing visual information and this label allows to represent the fact that no gesture belonging to the advice classes is being recognized.

7.5.2 Multi-modal Interactive Reinforcement Learning

The RL approach is performed using SARSA with a discount factor $\gamma = 0.9$, learning rate $\alpha = 0.3$, and ϵ-greedy action selection with $\epsilon = 0.1$. IRL is carried out using a probability of feedback $\mathcal{L} = 0.3$, meaning that 30% of the time we use the recorded advice to assist the robot in the task execution. We considered $\gamma^I > \theta_{min}$ with θ_{min} as the minimum confidence value to be considered as a valid

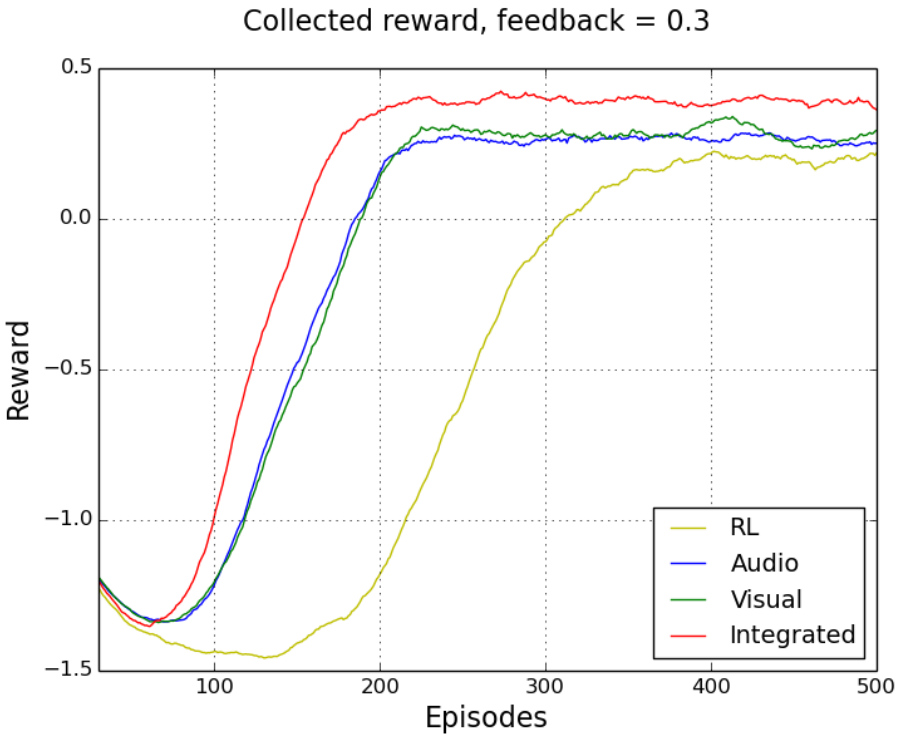
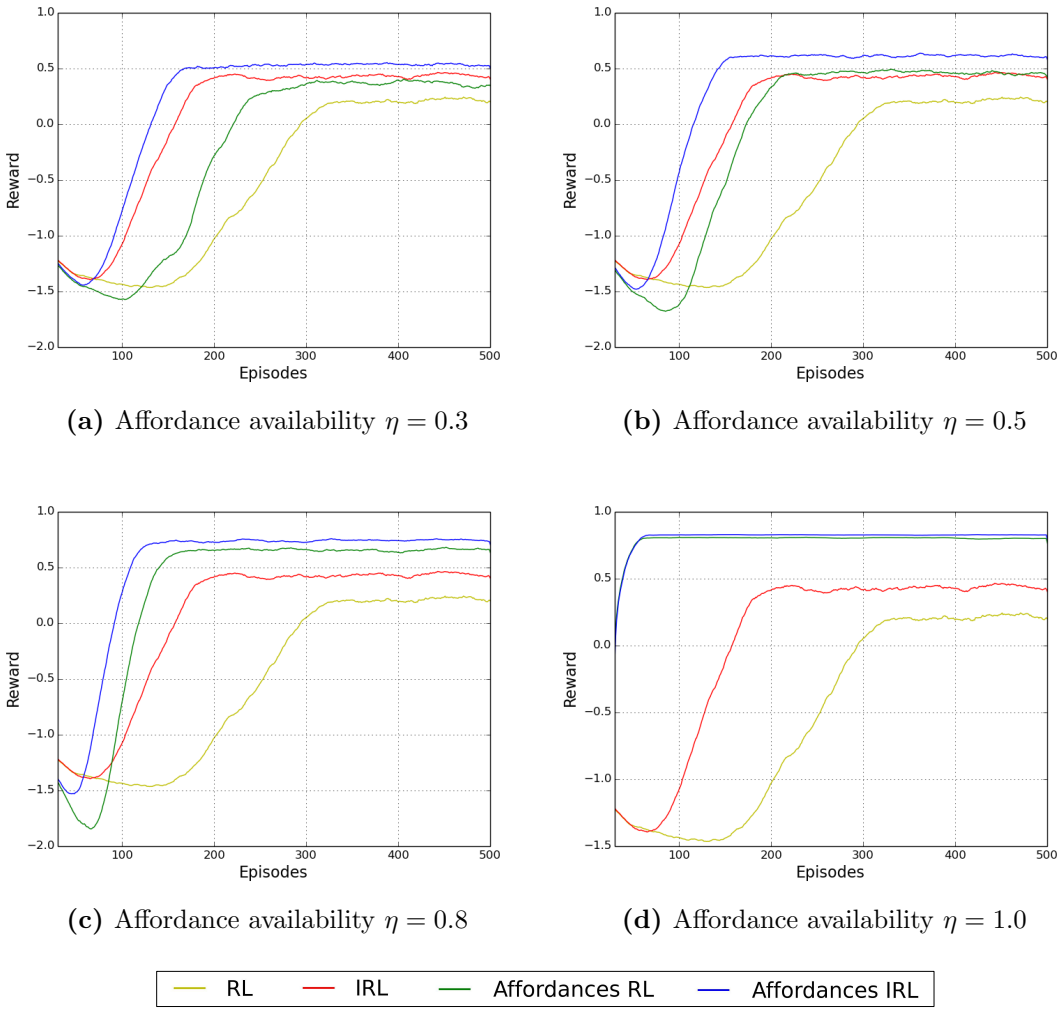


Figure 7.7: Collected rewards using autonomous RL, IRL with uni-modal feedback, and IRL with multi-modal feedback. A minimum confidence threshold of $\theta_{min} = 0.25$ is used in IRL approach. The multi-modal IRL approach converges faster and to a greater reward than the uni-modal IRL approaches (Cruz et al., 2016b).

advice. Then, we use different θ_{min} in order to verify whether smaller confidences are still beneficial. The thresholds used are $\theta_{min} \in \{0.0, 0.25, 0.5, 0.75\}$ observing that in general IRL works better with $\theta_{min} = 0.25$ in comparison to the other thresholds. The average convoluted rewards are shown in Fig. 7.6 for 100 agents and 500 episodes.

In order to evaluate differences in the uni-modal and multi-modal approaches, we use a threshold of $\theta_{min} = 0.25$. The results for 100 agents and 500 episodes are shown in Fig. 7.7 where is possible to observe that both uni-modal approaches lead to similar learning behavior, i.e.: similar convergence speed and accumulated reward. When using multi-modal integrated advice, the approach converges faster and collects greater reward in comparison with audio and visual advice only.



(a) Affordance availability $\eta = 0.3$

(b) Affordance availability $\eta = 0.5$

(c) Affordance availability $\eta = 0.8$

(d) Affordance availability $\eta = 1.0$

Figure 7.8: The plots show the collected reward for different values of affordance availability using autonomous RL and IRL. The results are compared to respect to RL and IRL without using contextual affordances. In all cases, the affordance-driven approaches yield a better performance in terms of the collected reward and convergence speed.

7.5.3 Contextual Affordance Integration

Both previously performed experiments did not use contextual affordances after the multi-modal integration. Therefore, we introduced the proposed affordance-driven model to avoid failed-states. This way, we do not only reduce the action space, but also the likelihood of a failed-state during an episode is diminished so that the agent is less likely to receive a punishment of -1. Consequently, using

the affordance-driven IRL approach increases the average accumulated reward and yields faster convergence (shown in Fig. 7.8). For a better comparison of our results, all plots in Fig. 7.8 show the autonomous RL and IRL approaches without the use of affordances using integrated feedback with minimal confidence threshold $\theta_{min} = 0.25$.

We evaluate our scenario using different percentages of available contextual affordances. We define a parameter η to be the likelihood of having a contextual affordance available. We set values for $\eta \in \{0.3, 0.5, 0.8, 1.0\}$ with $\eta = 1$ meaning that the affordance is fully available. We can observe in Fig. 7.8a that even a reduced availability of affordances ($\eta = 0.3$) allows to improve the learning process. Furthermore, the affordance-driven RL approach working autonomously (in green) accomplishes similar performance as IRL without affordances in terms of accumulated reward. In the case of affordance-driven IRL, it also reaches a better performance than IRL with no affordances. Fig. 7.8b shows the results for an affordance availability of $\eta = 0.5$. In this case, even the affordance-driven autonomous RL approach (in green) obtains bigger accumulated reward in comparison to the IRL approach where affordances are not used. Whereas in Fig. 7.8c with $\eta = 0.8$, both approaches with affordances outperform the traditional RL and IRL approaches in terms of accumulated reward and convergence speed.

Finally, we use an agent with full affordance availability, i.e., $\eta = 1.0$. Fig. 7.8d shows that with affordances being fully available, the agent quickly converges to its maximal possible reward in both RL and IRL approaches, with a slight difference in the maximal reward between both approaches when using affordances.

7.6 Discussion

In this chapter, we studied the interplay of multi-modal feedback with task-related knowledge in terms of contextual affordances in an IRL scenario with the aim of obtaining a more consistent advice to enhance the learning process in comparison to uni-modal IRL. The obtained results show that integrated audiovisual representations yield more robust feedback for an IRL task with respect to uni-modal approaches. In particular, audiovisual integration provides the means to solve conflicts, i.e., situations in which predicted feedback labels from the auditory and the

visual modules are incongruent.

In our approach, the integration is carried out taking into account the predicted labels and the confidence values from uni-modal cues. In the case of incongruent audiovisual predictions, the modality yielding the higher confidence value will be preferred. Gesture labels are predicted by the neural network processing of hand motion features, whereas vocal commands are predicted using in-domain automatic speech recognition. Consequently, these two approaches provide robust feedback predictions with confidence values computed as a function of a fully sensory-driven process, i.e., a high confidence value indicates that it is very likely that the feedback perceived by the agent matches the one actually given by the trainer. This procedure, however, does not give any information on whether the piece of feedback is correct or not in terms of the next actions required to accomplish the task.

Neurobehavioral studies in multi-modal processing have shown that human subjects exposed to audiovisual stimuli integrate multiple sources of information biased by a combination of sensory representations and prior expectations (Odegaard et al., 2015). In other words, signals from multiple sources are combined in the brain taking into account a combination of the reliability of low-level sensor representations and the expectations of an agent in a specific situation (e.g., in terms of task-oriented knowledge). Therefore, we integrated this aspect to our model in order for the study in the combination of sensory-driven multi-modal feedback and environmental knowledge in the context of our IRL task.

In the extended architecture, we integrated task-related knowledge in terms of contextual affordances which represent an effective method to anticipate the effect of actions performed by an agent interacting with objects based on its current state. We trained a neural network to predict the effect of performed actions with different objects in order to avoid states from which it is not possible for the agent to complete the cleaning-table task. Thus, contextual affordances modulate the influence of multi-modal feedback in the IRL algorithm, i.e., if an action provided by the trainer leads to a failed-state, it may be disregarded irrespective of a high (sensory-driven) confidence value.

Part IV

Closing

Chapter 8

Conclusions

8.1 Summary of the Thesis

From an experimental point of view, the present thesis was divided into two parts: Agent-Agent Interactive Reinforcement Learning and Human-Agent Interactive Reinforcement Learning. When analyzing agent-agent IRL, we first presented a method for training agents using interactive feedback and contextual affordances. Three different experimental set-ups were carried out: (i) training an agent with autonomous RL, used as the base to compare the results, (ii) training an agent with RL and contextual affordances to avoid failed-states, and (iii) training a second agent with IRL and contextual affordances. Moreover, in the IRL approach, we also tested the interplay of the probability of feedback and consistency of feedback. From this experimental set-up, we have shown that even small amounts of interaction are beneficial to the learner-agent performance in terms of performed actions and accumulated reward. The case of the consistency of feedback deserves special attention since small decreases in the probability of consistency diminish the performance considerably. Moreover, we have shown contextual affordances to be an effective method to avoid failed-states improving thus the overall IRL performance.

In agent-agent IRL, we implemented additional experiments to study what makes an agent a good teacher. We performed three additional experiments in order to: (i) study the differences of trainer-agents in terms of their internal knowledge

representation, (ii) investigate trainer and learner behavior to compare how the experience is distributed and how this affects the collected reward, and (iii) evaluate interaction parameters along with the learner-agent's obedience. In this regard, we have shown that the agent with the best performance may not be the best teacher due to its high specialization. On the contrary, using a polymath trainer-agent, with a more distributed experience, allows to advise the learner-agent properly in more situations. Additionally, an important finding from this set-up, since the learner-agent is much more sensitive to small variations in the consistency of feedback in comparison to the probability of feedback, is that the changes on the learner-agent's obedience may benefit the learning performance in the presence of low consistency in feedback.

Afterward, we studied two approaches in human-agent IRL, namely uni- and multi-modal feedback, with the aim of having a more realistic scenario but also to investigate how the reliability of the sensory processing affects the learning process. When using uni-modal feedback we used speech guidance with an ASR system. We used two different experimental set-ups: (i) ASR using different auditory sensors with presence of noise to evaluate how influential the hardware differences in the learning approach are with respect to consistent feedback, and (ii) IRL using speech guidance with contextual affordances using two different voice sensors which differ in the recognition error rate. We were able to verify that although speech guidance improved the performance of RL agents, in terms of the performed actions in comparison to autonomous RL and RL with affordances, an impoverished input sensor or the presence of a noisy communication channel may affect the consistency of feedback leading to worse IRL performance.

Finally, we extended our method to include multi-modal feedback and environmental knowledge in terms of contextual affordances. We incorporated multi-modal advice with the aim of producing a higher consistency of feedback, i.e., a higher confidence level of advice and therefore a faster learning process. Moreover, contextual affordances were added not only for faster convergence but also to modulate the integration of signals from multiple sources by representing the expectation of the learner-agent. We performed three different experimental set-ups: (i) multi-modal integration of speech and gestures using an associative architecture, (ii) comparing IRL using uni- and multi-modal advice with a minimal threshold to be considered as valid advice, and (iii) IRL using multi-modal advice modulated by

contextual affordances. We have shown multi-modal feedback to be an effective method to provide a stronger advice signal, allowing the IRL approach to accumulated more reward in comparison to uni-modal approaches. Furthermore, we have seen that using multi-modal advice incorporating environmental knowledge in terms of contextual affordances benefits the learner-agent by avoiding failed-states and accumulating more reward.

8.2 Discussion

The research presented in this thesis aimed at addressing the following main question: *May RL be sped up by using parent-like advice and affordance-driven environmental models?* This research question was addressed from three different perspectives by a subset of three more focused research questions. Although our experiments are related to the particular properties of the proposed scenario, our approach may be scaled up to more complex conditions in order to generalize our findings. In this regard, the transition function is defined in a general manner and therefore is feasible to extend it to consider more locations. The inclusion of additional objects or actions would consequently lead to more states. However, since the representation is discrete, the use of additional objects or actions would lead only to a limited number of new states. A different situation is the case of RL problems with continuous state-action representation, where the interactive feedback has to be modeled taking this continuous representation into consideration. This situation was out of the scope of this thesis.

In comparison to other previously presented discrete IRL approaches, our approach not only uses interactive feedback but we also integrate it with contextual affordances and we propose a more natural advice method using trainer's skills by means of the multi-modal integration of audiovisual signals. Although some results have already been discussed partially in the course of the thesis after each experimental chapter, we now turn to discussing the main results and findings that we obtained from the different performed experiments in the light of the research questions.

8.2.1 Interactive Feedback and Affordance-based Model

In this research, the first posed research question was: How can an affordance-based model of the environment support the IRL framework? In this regard, through the three different learning methods explained in chapter 4, we may verify significant differences in terms of the particular learning performance of each of them. In the case of classic RL, we obtained a substantial improvement by including a small negative reward after each performed action to encourage the robot to choose shorter paths towards the final state. This negative reward led to faster convergence and improved the success rate considerably in RL.

Nevertheless, despite the improvements in the classic RL paradigm, this approach still led to a lower performance than RL utilizing contextual affordances. Certainly, this performance was because the robotic agent on this occasion did not reach failed-states because the neural network architecture (using the current state, the action, and the object as input) anticipated the next state or the caused effect before the task execution, avoiding failed-states when necessary. This effectively decreased the search space for the learning. Better performance was furthermore observed in the collected reward. The maximal reward value reached by the robotic agent with classic RL was still less than the minimal reward value when RL with contextual affordances was used in all tested cases.

Results of the IRL approach showed that interaction provides advantages over RL with affordances in most of the tested levels of feedback. Even a small amount of interactive feedback helped the robot to finish the cleaning-table task faster. This is illustrated by a smaller number of performed actions as well as a bigger amount of collected reward. When the consistency of feedback was considered, it was observed that even small reductions of this parameter can make the learning process much slower. Therefore, we believe this parameter deserves special attention since small decreasing of it may be detrimental to the learning process substantially. Nevertheless, the learner-agent was still able to learn in the long run since an important part of the time the robot performed actions with reinforcement learning by autonomously exploring the environment. This also suggests that the consistency of feedback has a different impact on learning according to the probability of feedback used.

8.2.2 What Makes a Good Teacher?

The second research question posed in this research was: What constitutes a good teacher-agent when considering internal knowledge representation and interaction parameters? In chapter 5, it was shown that IRL generally helps to improve the performance of an RL agent using parent-like advice. Nonetheless, it is important to take into account that higher levels of interaction do not necessarily have a direct impact on the total accumulated reward. More importantly, the consistency of feedback seems to be more relevant when dealing with different learner obedience parameters (or a noisy or unreliable communication channel) since small variations can lead to considerably different amounts of collected reward. Therefore, the learner-agent's obedience is an effective way to ignore inconsistent feedback and an adaptive value of this may benefit the learner even further on.

Moreover, we have shown that there is divergence in the internal representation of the knowledge of the agents through state-action Q-values since there are states in which it is not possible to distinguish what action leads to greater reward. Agents with a smaller standard deviation among the visited states are preferred candidates to be parent-like teachers since they have a much better distribution of knowledge among the states. This allows them to adequately advise learner-agents on what action to perform in specific states. Agents with biased knowledge distributions collect more reward themselves, but nevertheless, have a subset of states where they cannot properly advise learners. This leads to a worse performance in the apprenticeship process in terms of maximal collected reward, convergence speed, and behavior stability represented as the standard deviation for each visited state.

Using the polymath agent as an advisor leads to both greater reward and faster convergence of the reward signal and also to more stable behavior in terms of the state visit frequency of the learner-agents, which can also be seen in the standard deviation of visited states when compared to the case of a specialist agent as a trainer.

8.2.3 Uni- and Multi-modal Advice

The final research question asked was: How beneficial is uni- and multi-modal advice during the apprenticeship process? In this regard, we presented a uni- and

a multi-modal IRL approach, in chapter 6 and chapter 7 respectively, to investigate how the reliability of the sensory processing affects the learning process in terms of consistent feedback and, consequently, performed actions and accumulated reward.

The presented multi-modal IRL approach uses dynamic audiovisual input as feedback in terms of vocal commands and hand gestures for guidance. Our approach integrates uni-modal cues to provide multi-modal feedback. The multi-modal integration module estimates a joint label and confidence value based on uni-modal predictions. The integration is of particular importance in the case in which the two modalities convey incongruent information, i.e., feedback classes predicted by the modules of speech and gesture recognition do not match. Therefore, our integration function takes into account the confidence level of the predictions to provide the IRL algorithm with more consistent feedback.

Although both sensory modalities show good advice prediction and confidence levels, the integrated advice leads to a better performance in our domestic scenario in terms of the accumulated reward and required learning episodes. In this regard, we have shown that our integration function allows to produce stronger unified advice with higher confidence levels and, consequently, to enhance the performance of a learning robot using multiple sources of information for a more natural parent-like learning procedure.

We have evaluated the learning performance under 3 different conditions: variations over the threshold of minimal confidence, comparison of uni- and multi-modal advice, and integration of environmental knowledge by contextual affordances with different values of availability. We have observed that multi-modal feedback allows to provide more confident advice to the learner-agent which leads to greater accumulated reward. Moreover, multi-modal integration modulated by contextual affordances enables the learner-agent to disregard the advice when this leads to failed-states improving further the IRL performance.

Multi-modal IRL allows the agent to interact in a more natural way with parent-like trainers for dynamically acquiring and refining task-related knowledge with respect to traditional IRL approaches. Together, our results demonstrate the contribution of multi-modal sensory processing integrated with environmental knowledge to significantly enhance the interaction between users and agents in robotic learning tasks.

8.3 Future Work

The obtained results motivate the extension of our approach in several directions. It would be an interesting improvement to investigate variations on the action selection method such as semi-uniform strategies like the adaptive ϵ-greedy strategy based on value differences (VDBE) (Tokic, 2010) where epsilon is reduced on the basis of the learning progress. On the one hand, high fluctuations in the estimates of value differences lead to a higher epsilon and further exploration and, on the other hand, low fluctuations lead to a lower epsilon and more exploitation. The method can also be combined with softmax-weighted action selection (Tokic and Palm, 2011). We hypothesize that a stronger classic RL approach which is the base for the other methods can lead to the necessity of less external advice allowing to reduce the number of iterations or needed episodes for training which is fundamentally important considering real scenarios where running through a large number of episodes would be impractical.

Furthermore, either the same or decreasing probability of feedback has been applied during the whole training process, i.e., we have not tested what the best time steps are to deliver interactive feedback. Evidence indicates that there are diverse factors which affect the ultimate performance of an apprentice agent using IRL methods such as the time period when the feedback is received (Torrey and Taylor, 2013; Taylor et al., 2014) as well as the magnitude of the problem where the method is applied (Stahlhut et al., 2015). Therefore, adjustments to the frequency of feedback in the implementation of the method GETADVICE (see Sec. 4.3.2) should also be investigated.

Further important future work is to investigate how the obtained results can scale in continuous scenarios. It can be expected that RL agents have similar behavior since they are designed to find the optimal solution, maximizing the collected reward. Moreover, adaptive learner behavior can be explored, thus allowing to decide which advice to follow depending on the collected knowledge about the current state that the learner-agent has at a specific time. Then, the learner-agent would act with diverse values for the learner obedience parameter, adapting it in real time. Greater learner obedience can be expected at the beginning of the learning process, but over time the learner-agent should take its own experience more into account and therefore follow its own policy instead of the parent-like

advice, leading to smaller obedience values. In the same way, if new space is explored and consequently less reward is received, then parent-like advice could be used once again, leading to a dynamic learning process, and taking advice into account when necessary while avoiding bad advice if possible.

So far, the proposed integration function considers two cues for predicting multi-modal feedback and computing its confidence. On the one hand, we could think of naturally extending our function to consider input from additional sensory sources, e.g., RGB information as an additional visual cue. It has been shown that combining depth and RGB information leads to a better recognition accuracy with respect to using a single cue (Koppula et al., 2013; Ni et al., 2013). In the case of our robotic task, conflicting input in terms of incongruent predictions from the auditory and visual modules may be solved by considering multiple visual cues and assuming that visual information is more reliable with respect to speech recognition. On the other hand, we could think of extending our function for additional modalities, e.g., haptics. In such a setting, parent-like advice may be delivered to the agent by providing haptic feedback to its actuators, e.g., moving its arm to grasp an object.

In its current version, our integration function considers each modality as equally contributing to multi-modal perception. Therefore, although our architecture scales up to a larger number of modalities, it does not account for crossmodal learning aspects, e.g., in an embodied robot perception scenario where motor contingencies may influence audiovisual representations (Morse et al., 2015). Thus, the extension of our approach in such a direction would require additional mechanisms for the crossmodal learning of spatiotemporal contingencies built on the basis of modality-specific properties (Noda et al., 2014; Giese and Rizzolatti, 2015) and the interplay with affordance-driven reinforcement learning.

Currently, our IRL scenario runs in an offline manner, i.e., with no dynamic human advice. Therefore, future work directions should also consider experiments accounting for online interactions. Furthermore, experiments should also consider a wider number of parent-like trainers with different teaching characteristics.

In this thesis, we have used robotic agents or simulated robots to perform the actions in the proposed task, therefore, another direction of extension is to move the proposed application onto more realistic robot platforms. For this, it is necessary

to count on robots which are able to grasp and place objects correctly in order to execute the subactions needed to complete the cleaning-table task.

8.4 Conclusion

This thesis contributes to the field of interactive reinforcement learning by exploring approaches aiming to speed up reinforcement learning methods, more specifically interactive feedback using both agent-agent and human-agent interaction. This was complemented by the use of contextual affordances as an approach to model the actions in the environment.

In conclusion, our experiments demonstrate that autonomous, classic reinforcement learning can be sped up by using parent-like trainers who are able to use their natural skills to deliver advice to the robot in terms of speech and gestures. Additionally, an affordance-driven environmental model, as contextual affordance, is beneficial to the learning process in terms of decreasing the training time by reducing the search space and using environmental knowledge to modulate sensory-driven parent-like advice.

Appendix A

Contextual Affordances with an Associative Neural Architecture

A.1 Introduction

Affordances are an effective method to anticipate the effect of actions performed by an agent interacting with objects. In this appendix, we present an additional implementation of contextual affordances in the framework of the robotic cleaning-table task.

We implement an associative neural architecture containing a layer with a quadratic complex neuron (Georgiou, 2006; Georgiou and Voigt, 2013) to learn the contextual affordances for predicting the effect of performed actions with different objects to avoid failed-states. The associative architecture shapes a virtual grid in a complex plane to map inputs into the output space.

Experimental results on a simulated robot environment show that our associative memory is able to learn in few iterations to predict future states with high accuracy using a humanoid robot that must clean the table interacting with different objects.

Moreover, in this experiment, we adjust the domestic scenario in terms of the allowed actions and data representation for the neural model. During the execution of the task, the robot will transit different states by performing actions and using objects until a desired final state is achieved.

A.2 Experimental Set-up

The robotic scenario used in this appendix differs from the one used during the thesis in terms of the available actions and their representation since, originally, we did not work with vision modality and, therefore, the scenario was modeled differently. As before, the task consists of a robot standing in front of a table to clean it. For this task, we used the same defined *objects* and *locations*. The scenario includes then a *sponge* and a *goblet*, and three zones, the *left* and *right* table sides and the *home* position. Differently, we allow the robot to perform actions as: *get* <*object*>, *drop* <*object*>, *go to* <*location*>, and *clean*. Therefore, now the actions are also linked to either objects or locations.

To implement the contextual affordance learning, we utilized an associative neural architecture as described in (Cruz et al., 2016c) using a complex-valued quadratic neuron to map inputs into a two-dimensional grid output. In this neural architecture, the output Y is computed according to Eq. (A.1) and the gradient descent learning rule as in Eq. (A.2) as follows:

$$y = \sum_{j=1}^{n} \sum_{k=1}^{n} \bar{x}_j x_k a_{jk} \tag{A.1}$$

$$\triangle A = \alpha \varepsilon \bar{X} X^T \tag{A.2}$$

where α is a small real-valued learning rate.

For a given input vector X, the desired output Y to be used in the learning algorithm is defined as the nearest intersection point of the grid lines of the complex plane. In practice, a function Ψ is defined that rounds to the nearest integer for grid lines spaced at a fixed distance δ in both directions:

$$\Psi(Y) = \frac{round(\delta Re(Y))}{\delta} + i\frac{round(\delta Im(Y))}{\delta} \tag{A.3}$$

All the variables are encoded as presented in Table A.1, which shows the data representation for side conditions, locations, actions, and objects. In side conditions, letters d and c represent the fact of being *dirty* or *clean* respectively.

Table A.1: Representation of training data used for neural classification.

Data Representation																
Side conditions					**Locations**				**Actions**					**Objects**		
dd	1	0	0	0	home	1	0	0	get	1	0	0	0	sponge	1	0
dc	0	1	0	0	left	0	1	0	drop	0	1	0	0	goblet	0	1
cd	0	0	1	0	right	0	0	1	go to	0	0	1	0	free	0	0
cc	0	0	0	1	none	0	0	0	clean	0	0	0	1			

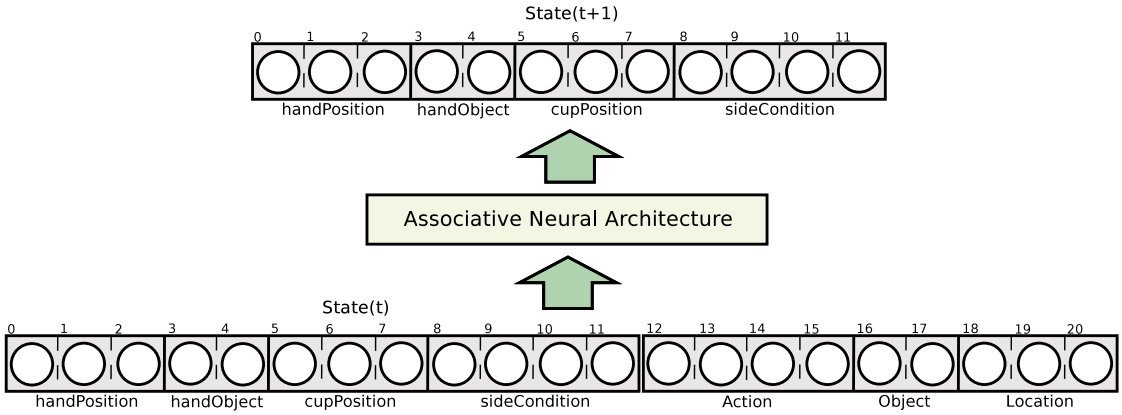


Figure A.1: Associative neural architecture for next state prediction. In our scenario, the state reached by the robot represents the affordance effect.

A.3 Experimental Results

Our approach uses contextual affordances to predict the effect of an action after it has been performed by the robot. We use the representation shown in Table A.1 for the training data. As input, we use vectors with 21 variables containing information about the current state, the action, the object and/or the location, whereas each state is contained in the first 12 components of the vector considering the four variables that define a state (see Fig. A.1). Our architecture comprises an associative neural layer that maps the current state of the system into the expected effect that corresponds to the effect from contextual affordances encoded as 12 variables representing the next state. When a performed action leads to a failed-state, all components of the output vector are equal to zero.

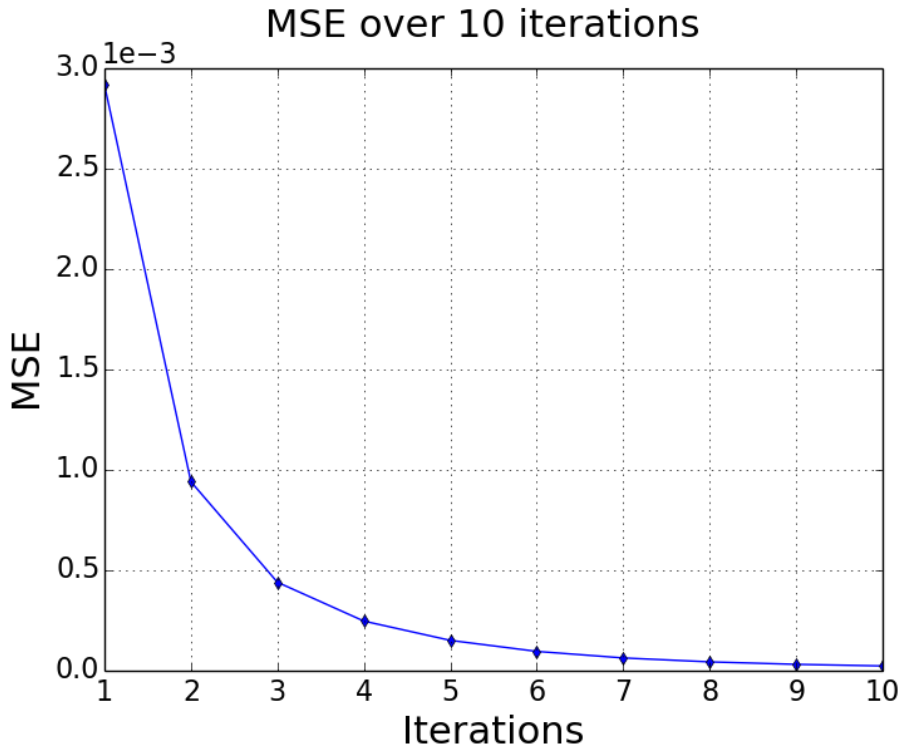


Figure A.2: Mean squared error over 10 training iterations.

During the training, we associate the desired output state label $l(\Psi(Y))$ for classification purposes. After the training phase, when a new sample is presented to the neuron, we compute y' and return the state label that minimizes $\|\Psi(y') - \Psi(Y)\|$. For our implementation, we set $\delta = 0.001$ and use the decaying learning rate:

$$\alpha_t = \alpha_0 * e^{\frac{-t(t+3)}{k}} \tag{A.4}$$

where t is the iteration number, $\alpha_0 = 0.01$ and $k = 5000$.

Experiments show that our architecture with an associative layer is able to classify all the instances correctly after training. The mean square error decreased from 2.92e-3 to 2.37e-5 after 10 iterations as shown in Fig. A.2. The final distribution of the output after 10 iterations is shown in Fig. A.3, where the x-axis and y-axis are the real and imaginary parts respectively of the complex plane.

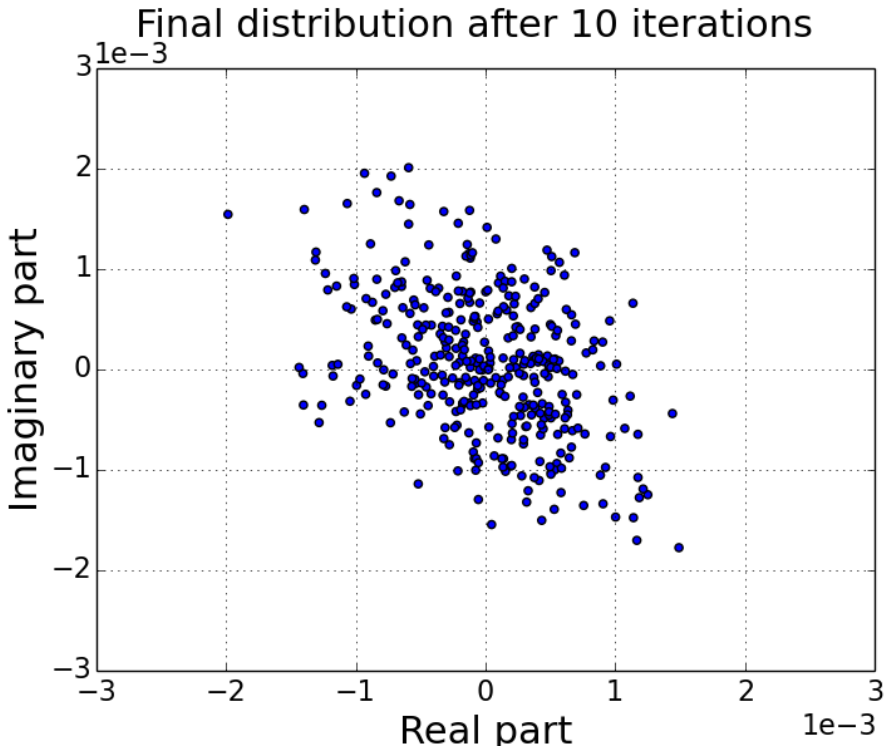


Figure A.3: Final distribution of the output projected into the complex domain.

A.4 Discussion

Our proposed architecture is able to successfully predict the effect of performing an action using an object by using contextual affordances. We use additional state information to distinguish different situations in the robotic cleaning scenario and avoid failed-states to effectively finish the task. The associative complex architecture is, therefore, an interesting alternative to implementing contextual affordance since it allows to map the input vectors into valid states with few training iterations, which represents an advantage for online learning applications where the response time plays a crucial role.

Appendix B

State Transitions of the Cleaning-table Scenario

In order to visualize the whole search space of the cleaning-table scenario, a more detailed diagram for the state transitions is shown in Fig. B.1. The figure does not pursue to show all the involved transitions between states in detail, but rather to show how the search space is structured. Transition details can be seen in the state transition table as shown in Table 3.3.

Fig. B.1 shows the 53 states plus a garbage collector state which represents the failed-states. All the involved transition are displayed in the figure, i.e., for every node, there are 7 output edges representing the 7 defined actions in the cleaning-table scenario.

A simplified version of the transitions is shown in Fig. B.2 where the edges for the action of *abort* (to cancel the task execution and return to the initial state) and the edges for actions which lead to a failed-state are omitted. In Fig. B.2, we can see the path A to the left and the path B to the right. As stated in chapter 3, the path A comprises 23 states and the path B 31 states.

The state transitions were represented using the graph description language DOT and the plot layouts were created with the graph visualization software Graphviz for rendering. For more details, the source files of the vectorial figures can be found at https://git.informatik.uni-hamburg.de/cruz/IRL.

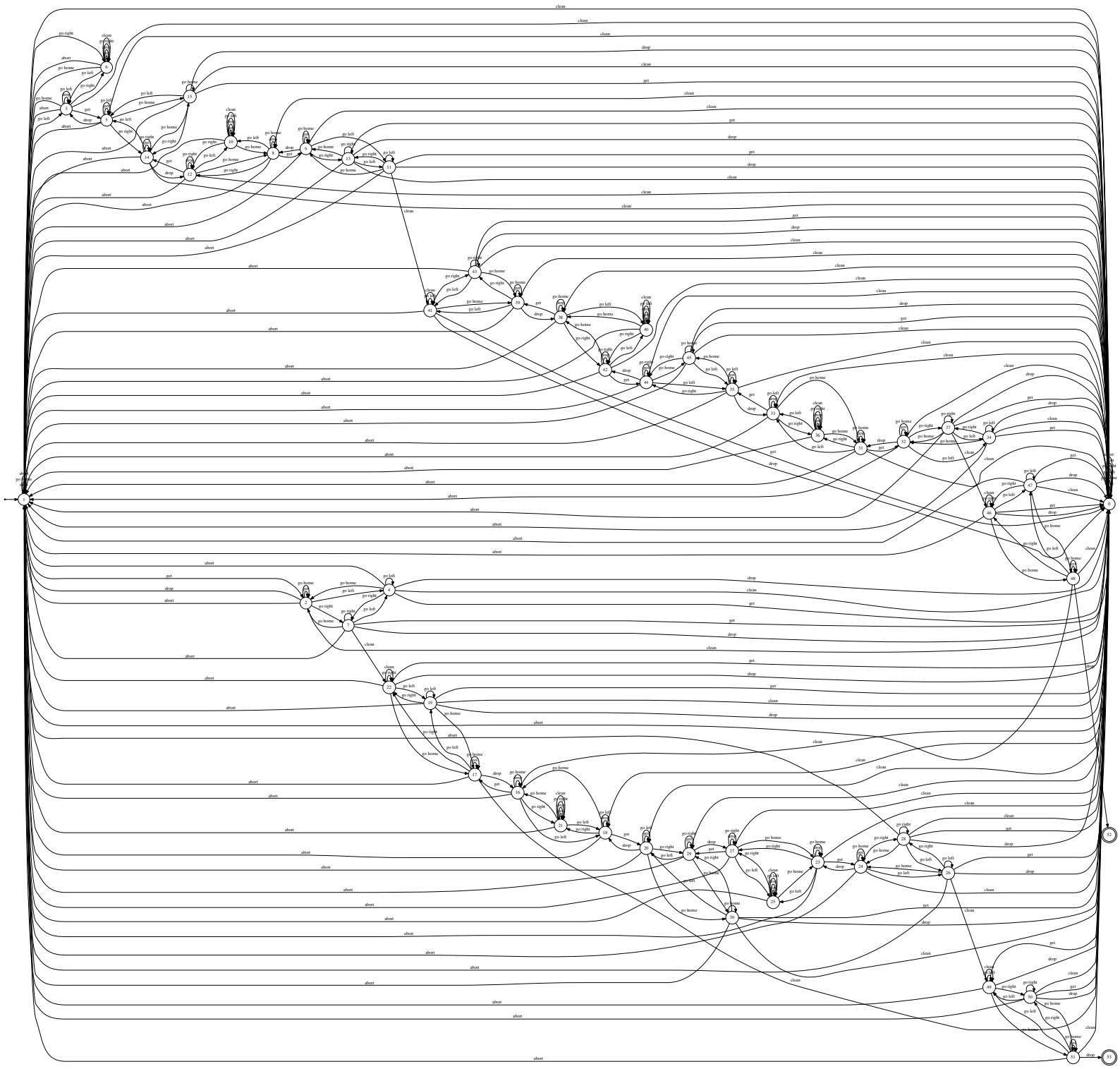


Figure B.1: Full transition diagram of the cleaning-table scenario.

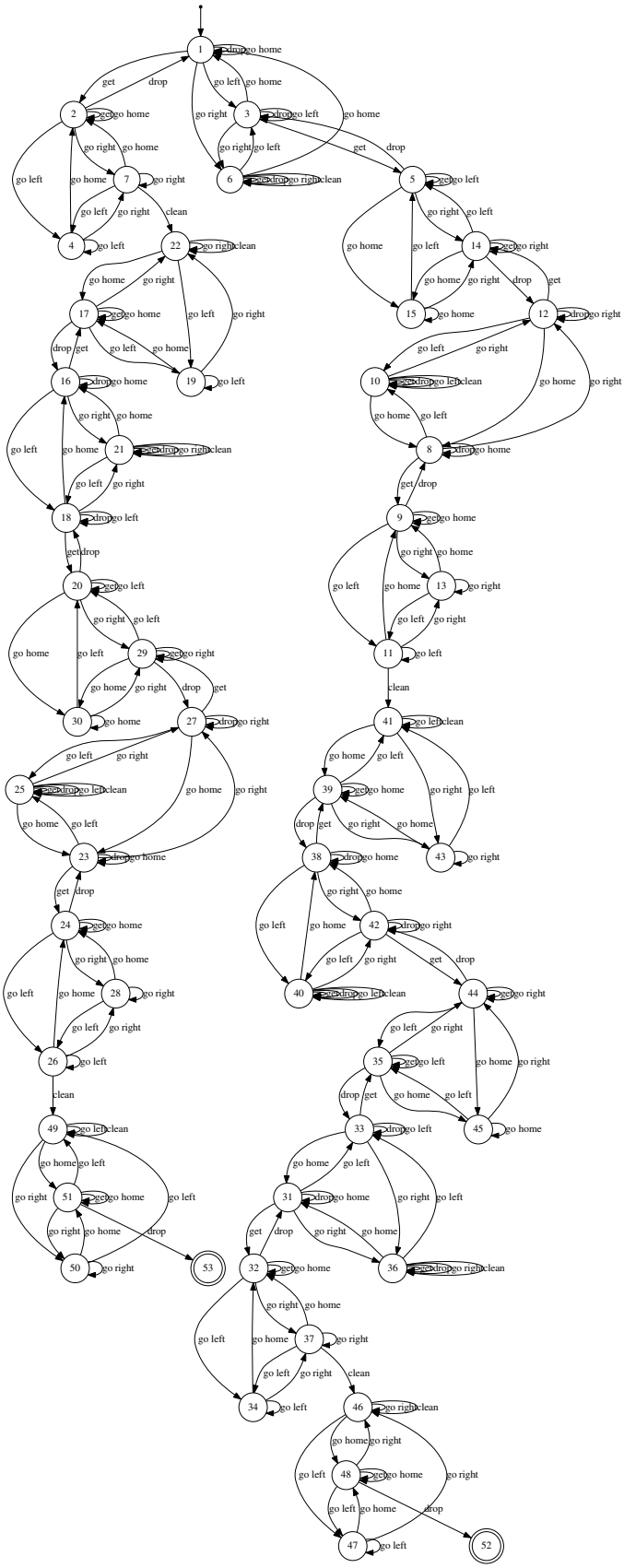


Figure B.2: Simplified transition diagram of the cleaning-table scenario.

Appendix C

Published Contributions Originating from this Thesis

C.1 Journals

- Francisco Cruz, Sven Magg, Cornelius Weber, and Stefan Wermter. Training Agents with Interactive Reinforcement Learning and Contextual Affordances. *Journal IEEE Transactions on Cognitive and Developmental Systems (TCDS)*, Vol. 8, Nr. 4, pp. 271-284, December 2016.
- Francisco Cruz, Sven Magg, Yukie Nagai, and Stefan Wermter. Improving interactive reinforcement learning: What makes a good teacher? *Submitted to Connection Science*, 2017.

C.2 Conferences

- Francisco Cruz, German I. Parisi, Johannes Twiefel, and Stefan Wermter. Multi-modal Integration of Dynamic Audiovisual Patterns for an Interactive Reinforcement Learning Scenario. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp.759-766, Daejeon, Korea, 2016.

- Francisco Cruz, German I. Parisi, and Stefan Wermter. Learning Contextual Affordances with an Associative Neural Architecture. *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pp. 665-670, Bruges, Belgium, 2016.
- Francisco Cruz, Johannes Twiefel, Sven Magg, Cornelius Weber, and Stefan Wermter. Interactive Reinforcement Learning through Speech Guidance in a Domestic Scenario. *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, pp. 13411348, Killarney, Ireland, 2015.
- Francisco Cruz, Sven Magg, Cornelius Weber, and Stefan Wermter. Improving Reinforcement Learning with Interactive Feedback and Affordances. *Proceedings of the Fourth Joint IEEE International Conference on Developmental Learning and Epigenetic Robotics (ICDL-EpiRob)*, pp. 125-130, Genoa, Italy, 2014.
- Francisco Cruz, Peter Wüppen, Alvin Fazrie, Sven Magg, and Stefan Wermter. Agent-advising Approaches in an Interactive Reinforcement Learning Scenario. *Accepted to the Joint IEEE International Conference on Developmental Learning and Epigenetic Robotics (ICDL-EpiRob)*, Lisbon, Portugal, 2017.

C.3 Workshops

- Francisco Cruz, German I. Parisi, and Stefan Wermter. Multi-modal Integration of Speech and Gestures for Interactive Robot Scenarios. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on Machine Learning Methods for High-Level Cognitive Capabilities in Robotics*, Daejeon, Korea, 2016.
- Nikhil Churamani, Francisco Cruz, Sascha Griffiths, and Pablo Barros. iCub: Learning Emotion Expressions using Human Reward. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on Bio-inspired Social Robot Learning in Home Scenarios*, Daejeon, Korea, 2016.

- Francisco Cruz, Johannes Twiefel, and Stefan Wermter. Performing a Cleaning Task in a Simulated Human-Robot Interaction Environment. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop An Open-source Recipe for Teaching/Learning Robotics with a Simulator*, Hamburg, Germany, 2015.
- Francisco Cruz, German I. Parisi, and Stefan Wermter. Contextual Affordances for Action-Effect Prediction in a Robotic-Cleaning Task. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop Learning Object Affordances: A Fundamental Step to Allow Prediction, Planning and Tool Use?*, Hamburg, Germany, 2015.

Appendix D

List of Acronyms

AI – Artificial Intelligence.

ANN – Artificial Neural Network.

ASR – Automatic Speech Recognition.

CAff – Contextual Affordance.

DMLP – Deep Multi-layer Perceptron.

DoF – Degree of Freedom.

G2P – Grapheme to Phoneme.

GVS – Google Voice Search.

GWR – Growing When Required.

HRI – Human-robot Interaction.

IRL – Interactive Reinforcement Learning.

MDP – Markov Decision Process.

MLP – Multi-layer Perceptron.

MMI – Multi-modal Integration.

NICO – Neural Inspired COmpanion.

NN – Neural Network.

RL – Reinforcement Learning.

SARSA – State, Action, Reward, State, Action.

SER – Sentence Error Rate.

VDBE – Value-Difference Based Exploration.

WER – Word error rate.

Appendix E

Acknowledgements

I would like to thank my supervisor Prof. Stefan Wermter for all his support during my doctoral studies. Undoubtedly, without his appropriate advice this work would not have been possible whatsoever. I also want to thank Dr. Cornelius Weber and Dr. Sven Magg for helpful discussions and inspiring feedback. Special thanks to Katja Köster and Erik Strahl for their immeasurable administrative and technical support respectively.

Additionally, I would like to thank all my colleagues from the Knowledge Technology group at the University of Hamburg, especially to the ones I had the opportunity to collaborate with as German I. Parisi, Johannes Twiefel, and Pablo Barros.

The most special acknowledgment is for the members of my family and their unconditional support through my whole life. To my parents and my sister, who believed in me since I was child encouraging me to be a better person. To my loved wife, who decided to give it all from the very beginning when this project was just an idea, and for all her support during our stay in Germany. Now, we will come back to Chile with a new family member, our son Nahuel.

Finally, I gratefully acknowledge the partial financial support by Universidad Central de Chile and Comisión Nacional de Investigación Científica y Tecnológica CONICYT, and the program BecasChile, beca 5043.

Bibliography

Amir, O., Kamar, E., Kolobov, A., and Grosz, B. (2016). Interactive teaching strategies for agent training. In *Proceedings of International Joint Conference on Artificial Intelligence IJCAI*, pages 804–811.

Ammar, H. B., Taylor, M. E., Tuyls, K., and Weiss, G. (2012). Reinforcement learning transfer using a sparse coded inter-task mapping. In *Proceedings of European Workshop Multi-Agent Systems*, pages 1–16. Heidelberg, Germany: Springer.

Andre, M., Popescu, V. G., Shaikh, A., Medl, A., Marsic, I., Kulikowski, C., and Flanagan, J. (1998). Integration of speech and gesture for multimodal human-computer interaction. In *Proceedings of International Conference on Cooperative Multimodal Communication*, pages 28–30.

Atıl, I., Dağ, N., Kalkan, S., and Şahin, E. (2010). Affordances and emergence of concepts. In *Proceedings of 10th International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, pages 149–156.

Awaad, I., Kraetzschmar, G., and Hertzberg, J. (2015). The role of functional affordances in socializing robots. *International Journal of Social Robotics*, 7:421–438.

Bandera, J. P., Rodríguez, J. A., Molina-Tanco, L., and Bandera, A. (2012). A survey of vision-based architectures for robot learning by imitation. *International Journal of Humanoid Robotics*, 9:1–40.

Bauer, J., Dávila-Chacón, J., and Wermter, S. (2015). Modeling development of natural multi-sensory integration using neural self-organisation and probabilistic population codes. *Connection Science*, 27:358–376.

Bisani, M. and Ney, H. (2008). Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50:434–451.

Bishop, C. M. (2011). *Pattern Recognition and Machine Learning*. 2nd ed. New York, NY, USA: Springer.

Breazeal, C. and Velásquez, J. (1998). Toward teaching a robot 'infant' using emotive communication acts. In *Proceedings of Simulated Adaptive Behavior Workshop on Socially Situated Intelligence*, pages 25–40.

Busoniu, L., Babuska, R., Schutter, B. D., and Ernst, D. (2010). *Reinforcement Learning and Dynamic Programming Using Function Approximators*. CRC press.

Cangelosi, A. and Schlesinger, M. (2015). *Developmental Robotics: From Babies to Robots*. Cambridge, MA, USA: MIT Press.

Cederborg, T., Grover, I., Isbell, C., and Thomaz, A. (2015). Policy shaping with human teachers. In *Proceedings of International Joint Conference on Artificial Intelligence IJCAI*, pages 3366–3372.

Chemero, A. (2011). *Radical Embodied Cognitive Science*. Cambridge, MA, USA: MIT Press.

Chemero, A. and Turvey, M. T. (2007). Gibsonian affordances for roboticists. *Adaptive Behavior*, 15:473–480.

Cruz, F., Magg, S., Weber, C., and Wermter, S. (2014). Improving reinforcement learning with interactive feedback and affordances. In *Proceedings of 4th Joint IEEE International Conference on Developmental Learning and Epigenetic Robotics ICDL-EpiRob*, pages 165–170.

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

Cruz, F., Parisi, G. I., Twiefel, J., and Wermter, S. (2016b). Multi-modal integration of dynamic audiovisual patterns for an interactive reinforcement learning scenario. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, pages 759–766.

Cruz, F., Parisi, G. I., and Wermter, S. (2016c). Learning contextual affordances with an associative neural architecture. In *Proceedings of the 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 665–670.

Cruz, F., Twiefel, J., Magg, S., Weber, C., and Wermter, S. (2015). Interactive reinforcement learning through speech guidance in a domestic scenario. In *Proceedings of International Joint Conference on Neural Networks IJCNN*, pages 1341–1348.

Şahin, E., Çakmak, M., Doğar, M. R., Uğur, E., and Üçoluk, G. (2007). To afford or not to afford: A new formalization of affordances toward affordance-based robot control. *Adaptive Behavior*, 15:447–472.

Cybenko, G. (1989). Approximation by superpositions of a sigmoid function. *Mathematics of Control, Signals, and Systems*, 2:303–314.

Deak, G. O., Krasno, A. M., Triesch, J., Lewis, J., and Sepeta, L. (2014). Watch the hands: Infants can learn to follow gaze by seeing adults manipulate objects. *Developmental Science*, 17:270–281.

Farkaš, I., Malík, T., and Rebrová, K. (2012). Grounding the meanings in sensorimotor behavior using reinforcement learning. *Frontiers in Neurorobotics*, 6:1–13.

Funahashi, K. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192.

Georgiou, G. (2006). Exact interpolation and learning in quadratic neural networks. In *Proceedings of International Joint Conference on Neural Networks IJCNN*, pages 230–234.

Georgiou, G. and Voigt, K. (2013). Self-organizing maps with a single neuron. In *Proceedings of International Joint Conference on Neural Networks IJCNN*, pages 1–6.

Gershman, S. J. and Niv, Y. (2015). Novelty and inductive generalization in human reinforcement learning. *Topics in Cognitive Science*, 7:391–415.

Gibson, J. J. (1966). *The Senses Considered as Perceptual Systems*. Boston: Houghton Mifflin.

Gibson, J. J. (1979). *The Ecological Approach to the Visual Perception of Pictures*. Boston, MA, USA: Houghton Mifflin.

Giese, M. A. and Rizzolatti, G. (2015). Neural and computational mechanisms of action processing: Interaction between visual and motor representations. *Neuron*, 88(1):167–180.

Griffith, S., Subramanian, K., Scholz, J., Isbell, C., and Thomaz, A. (2013). Policy shaping: Integrating human feedback with reinforcement learning. In *Proceedings of Advances in Neural Information Processing Systems*, pages 2625–2633.

Griffiths, S., Nolfi, S., Morlino, G., Schillingmann, L., Kuehnel, S., Rohlfing, K., and Wrede, B. (2012). Bottom-up learning of feedback in a categorization task. In *Proceedings of IEEE International Conference on Development and Learning and Epigenetic Robotics ICDL-EpiRob*, pages 1–6.

Grizou, J., Lopes, M., and Oudeyer, P.-Y. (2013). Robot learning simultaneously a task and how to interpret human instructions. In *Proceedings of 3rd Joint IEEE International Conference on Developmental Learning and Epigenetic Robotics ICDL-EpiRob*, pages 1–8.

Hagan, M. T. and Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5:989–993.

Hämmerer, D. and Eppinger, B. (2012). Dopaminergic and prefrontal contributions to reward-based learning and outcome monitoring during child development and aging. *Developmental Psychology*, 48:862–874.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multi-layer feedforward networks are universal approximators. *Neural Networks*, 2:359–366.

Horton, T. E., Chakraborty, A., and Amant, R. S. (2012). Affordances for robots: A brief survey. *AVANT: Journal of Philosophical-Interdisciplinary Vanguard*, 3:70–84.

Jamone, L., Ugur, E., Cangelosi, A., Fadiga, L., Bernardino, A., Piater, J., and Santos-Victor, J. (2017). Affordances in psychology, neuroscience and robotics: A survey. *IEEE Transactions on Cognitive and Developmental Systems*, 9:1–22.

Kammer, M., Schack, T., Tscherepanow, M., and Nagai, Y. (2011). From affordances to situated affordances in robotics – why context is important. In *Frontiers in Computational Neuroscience (Conference Abstract: Joint IEEE International Conference on Developmental Learning and Epigenetic Robotics ICDL-EpiRob)*.

Kimura, D. and Hasegawa, O. (2015). Estimating multimodal attributes for unknown objects. In *Proceedings of International Joint Conference on Neural Networks IJCNN*, pages 1–8.

Knox, W., Glass, B., Love, B., Maddox, W., and Stone, P. (2012). How humans teach agents. *International Journal of Social Robotics*, 4:409–421.

Knox, W. B. and Stone, P. (2009). Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of Fifth International Conference on Knowledge Capture*, pages 9–16. Redondo Beach, CA, USA.

Knox, W. B. and Stone, P. (2012). Reinforcement learning from human reward: Discounting in episodic tasks. In *Proceedings of IEEE International Symposium on Robot and Human Interactive Communication RO-MAN*, pages 878–885.

Knox, W. B., Stone, P., and Breazeal, C. (2013a). Teaching agents with human feedback: A demonstration of the tamer framework. In *Proceedings of International Conference on Intelligent User Interfaces Companion*, pages 65–66.

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

Kober, J., Bagnell, J. A., and Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32:1–37.

Kober, J. and Peters, J. (2012). Reinforcement learning in robotics: A survey. *Reinforcement Learning*, 12:579–610.

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

Koppula, H. S., Gupta, R., and Saxena, A. (2013). Learning human activities and object affordances from RGB-D videos. *The International Journal of Robotics Research*, 32:951–970.

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

Kornblum, S., Hasbroucq, T., and Osman, A. (1990). Dimensional overlap: Cognitive basis for stimulus-response compatibility – a model and taxonomy. *Psychological review*, 97:253–270.

Lacheze, L., Guo, Y., Benosman, R., Gas, B., , and Couverture, C. (2009). Audio/video fusion for objects recognition. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, pages 652–657.

Lapeyre, M., Rouanet, P., Grizou, J., N'Guyen, S., Falher, A. L., Depraetre, F., and Oudeyer, P.-Y. (2014). Poppy: Open source 3D printed robot for experiments in developmental robotics. In *Proceedings of Joint IEEE International Conferences on Development and Learning and Epigenetic Robotics ICDL-EpiRob*, pages 173–174.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707.

Lin, L. J. (1991). Programming robots using reinforcement learning and teaching. In *Proceedings of Association for the Advancement of Artificial Intelligence Conference AAAI*, pages 781–786.

Lopes, M., Melo, F. S., and Montesano, L. (2007). Affordance-based imitation learning in robots. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, pages 1015–1021.

Marsland, S. (2015). *Machine Learning: An Algorithmic Perspective*. CRC press.

Marsland, S., Shapiro, J., and Nehmzow, U. (2002). A self-organising network that grows when required. *Neural Networks*, 15:1041–1058.

Min, H., Yi, C., Luo, R., Zhu, J., and Bi, S. (2016). Affordance research in developmental robotics: A survey. *IEEE Transactions on Cognitive and Developmental Systems*, 8:237–255.

Mitchell, T. M. (1997). *Machine learning*. Burr Ridge, IL: McGraw Hill.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518:529–533.

Moldovan, B., Moreno, P., van Otterlo, M., Santos-Victor, J., and Raedt, L. D. (2012). Learning relational affordance models for robots in multi-object manipulation tasks. In *Proceedings of IEEE International Conference on Robotics and Automation ICRA*, pages 4373–4378. St. Paul, MN, USA.

Montesano, L., Lopes, M., Bernardino, A., and Santos-Victor, J. (2008). Learning object affordances: From sensory-motor coordination to imitation. *IEEE Transactions on Robotics*, 24:15–26.

Morse, A. F., Benitez, V. L., Belpaeme, T., Cangelosi, A., and Smith, L. B. (2015). Posture affects how robots and infants map words to objects. *PLoS ONE*, 10(3):1–17.

Nguyen, D. and Widrow, B. (1990). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In *Proceedings of IEEE International Joint Conference on Neural Networks IJCNN*, pages 21–26.

Ni, B., Pei, Y., Moulin, P., and Yan, S. (2013). Multilevel depth and image fusion for human activity detection. *IEEE Transactions on Cybernetics*, 43(5):1383–1394.

Niv, Y. (2009). Reinforcement learning in the brain. *Journal of Mathematical Psychology*, 53:139–154.

Noda, K., Arie, H., Suga, Y., and Ogata, T. (2014). Multimodal integration learning of robot behavior using deep neural networks. *Robotics and Autonomous Systems*, 62(6):721–736.

Nola, R. and Sankey, H. (2014). *Theories of Scientific Method: An Introduction*. Routledge.

Odegaard, B., Wozny, D. R., and Shams, L. (2015). Biases in visual, auditory, and audiovisual perception of space. *PLoS Computational Biology*, 11:1–23.

Ozasa, Y., Ariki, Y., Nakano, M., and Martinetz, N. I. (2012). Disambiguation in unknown object detection by integrating image and speech recognition confidences. In *Proceedings of Asian Conference on Computer Vision*, pages 85–96.

Parisi, G., Jirak, D., and Wermter, S. (2014). Handsom - neural clustering of hand motion for gesture recognition in real time. In *Proceedings of IEEE International Symposium on Robot and Human Interactive Communication RO-MAN*, pages 981–986.

Parisi, G. I., Weber, C., and Wermter, S. (2015). Self-organizing neural integration of pose-motion features for human action recognition. *Frontiers in Neurorobotics*, 9:1–14.

Pavlov, I. (1927). *Conditioned Reflexes: An Investigation into the Physiological Activity of the Cerebral Cortex*. New York: Dover.

Payzan-LeNestour, E., Dunne, S., Bossaerts, P., and O'Doherty, J. P. (2013). The neural representation of unexpected uncertainty during value-based decision making. *Neuron*, 79:191–201.

Peters, J., Kober, J., Mülling, K., Krämer, O., and Neumann, G. (2013). Towards robot skill learning: From simple skills to table tennis. In *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 627–631. Heidelberg, Germany: Springer.

Pilarski, P. and Sutton, R. (2012). Between instruction and reward: human-prompted switching. In *Proceedings of the AAAI Fall Symposium: Robots Learning Interactively from Human Teachers*, pages 46–52.

Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley.

Rescorla, R. A. and Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and non reinforcement. *Classical Conditioning II: Current Research and Theory*, 2:64–99.

Rieser, V. and Lemon, O. (2011). *Reinforcement Learning for Adaptive Dialogue Systems*. Heidelberg, Germany: Springer.

Rivest, F., Bengio, Y., and Kalaska, J. (2004). Brain inspired reinforcement learning. In *Proceedings of Advances in Neural Information Processing Systems NIPS*, pages 1129–1136.

Rohmer, E., Singh, S. P. N., and Freese, M. (2013). V-REP: A versatile and scalable robot simulation framework. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, pages 1321–1326.

Rozo, L., Jiménez, P., and Torras, C. (2013). A robot learning from demonstration framework to perform force-based manipulation tasks. *Intelligent Service Robotics*, 6:33–51.

Rummery, G. A. and Niranjan, M. (1994). On-line q-learning using connectionist systems. *Technical Report CUED/F-INFENG/TR166*.

Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Egnlewood Cliffs.

Sanchez-Riera, J., Alameda-Pineda, X., Wienke, J., Deleforge, A., Arias, S., Cech, J., Wrede, S., and Horaud, R. (2012). Online multimodal speaker detection for humanoid robots. In *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, pages 126–133.

Schalkwyk, J., Beeferman, D., Beaufays, F., Byrne, B., Chelba, C., Cohen, M., Kamvar, M., and Strope, B. (2010). Your word is my command: Google search by voice: A case study. In *Advances in Speech Recognition. Mobile Environments, Call Centers and Clinics*, pages 61–90.

Sigaud, O. and Droniou, A. (2016). Towards deep developmental learning. *IEEE Transactions on Cognitive and Developmental Systems*, 8:99–114.

Stahlhut, C., Navarro-Guerrero, N., Weber, C., and Wermter, S. (2015). Interaction is more beneficial in complex reinforcement learning problems than in

simple ones. In *Proceedings of Interdisziplinärer Workshop Kognitive Systeme (KogSys)*, pages 142–150.

Suay, H. B. and Chernova, S. (2011). Effect of human guidance and state space size on interactive reinforcement learning. In *Proceedings of IEEE International Symposium on Robot and Human Interactive Communication RO-MAN*, pages 1–6.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: Bradford Book.

Szepesvári, C. (2010). *Algorithms for Reinforcement Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool.

Tadele, T. S., de Vries, T., and Stramigioli, S. (2014). The safety of domestic robotics: A survey of various safety-related publications. *IEEE Robotics & Automation Magazine*, 21:134–142.

Taylor, M. E., Carboni, N., Fachantidis, A., Vlahavas, I., and Torrey, L. (2014). Reinforcement learning agents providing advice in complex video games. *Connection Science*, 26:45–63.

Thomaz, A. L. and Breazeal, C. (2006). Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *Proceedings of 21st National Conference on Artificial Intelligence*, pages 1000–1005. Boston, MA, USA.

Thomaz, A. L. and Breazeal, C. (2007). Asymmetric interpretations of positive and negative human feedback for a social learning agent. In *Proceedings of IEEE International Symposium on Robot and Human Interactive Communication RO-MAN*, pages 720–725.

Thomaz, A. L., Hoffman, G., and Breazeal, C. (2005). Real-time interactive reinforcement learning for robots. In *Proceedings of Association for the Advancement of Artificial Intelligence Conference AAAI, Workshop on Human Comprehensible Machine Learning*, pages 9–13.

Thorndike, E. L. (1911). *Animal Intelligence: Experimental Studies*. The Macmillan Company, New York, NY, USA.

Tokic, M. (2010). Adaptive ϵ-greedy exploration in reinforcement learning based on value differences. In *Proceedings of Annual Conference on Artificial Intelligence*, pages 203–210. Heidelberg, Germany: Springer.

Tokic, M. and Palm, G. (2011). Value-difference based exploration: Adaptive control between ϵ-greedy and softmax. In *Proceedings of 34th Annual German Conference on Advances in Artificial Intelligence*, pages 335–346. Heidelberg, Germany: Springer.

Torrey, L. and Taylor, M. (2013). Teaching on a budget: Agents advising agents in reinforcement learning. In *Proceedings of International Conference on Autonomous Agents and Multi-agent Systems AAMAS*, pages 1053–1060.

Twiefel, J., Baumann, T., Heinrich, S., and Wermter, S. (2014). Improving domain-independent cloud-based speech recognition with domain-dependent phonetic post-processing. In *Proceedings of Association for the Advancement of Artificial Intelligence Conference AAAI*, pages 1529–1535.

Ugur, E., Nagai, Y., Celikkanat, H., and Oztop, E. (2015). Parental scaffolding as a bootstrapping mechanism for learning grasp affordances and imitation skills. *Robotica*, 33:1163–1180.

Wang, C., Hindriks, K. V., and Babuska, R. (2013). Robot learning and use of affordances in goal-directed tasks. In *Proceedings of International Conference on Intelligent Robots and Systems IROS*, pages 2288–2294.

Warren, H. C. (1916). Mental association from Plato to Hume. *Psychological Review*, 23:208–230.

Watkins, C. J. (1989). *Learning from Delayed Rewards*. Doctoral dissertation, University of Cambridge.

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

Weber, C., Elshaw, M., Wermter, S., Triesch, J., and Willmot, C. (2008). Reinforcement learning embedded in brains and robots. In *Reinforcement Learning: Theory and Applications*, pages 119–142. I-Tech Education and Publishing.

Wermter, S., Elshaw, M., Weber, C., Panchev, C., and Erwin, H. (2003). Towards integrating learning by demonstration and learning by instruction in a

multimodal robot. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems IROS, Workshop on Robot Learning by Demonstration*, pages 72–79.

Wiering, M. and Otterlo, M. V. (2012). *Reinforcement Learning, State-of-the-Art*. Springer Heidelberg.

Wimmer, G. E., Daw, N. D., and Shohamy, D. (2012). Generalization of value in reinforcement learning by humans. *European Journal of Neuroscience*, 35:1092–1104.

Wise, R. A., Spindler, J., and Gerberg, G. J. (1978). Neuroleptic-induced "anhedonia" in rats: pimozide blocks reward quality of food. *Science, New Series*, 201:262–264.

Declaration of Oath
Eidesstattliche Versicherung

I hereby declare, on oath, that I have written the present dissertation by my own and have not used other than the acknowledged resources and aids.

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Hamburg, June 2nd, 2017 — Francisco Cruz

City, date — Signature

Ort, Datum — *Unterschrift*